

9.3

*IBM MQ Geliştirme Uygulamaları
Başvurusu*

IBM

Not

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, [“Özel notlar” sayfa 2163](#) bölümündeki bilgileri okuyun.

Bu basım, yeni basımlarda tersi belirtilmedikçe, IBM® MQ sürüm 9 yayın düzeyi 3 ve sonraki tüm yayınlar ve deđişiklikler için geçerlidir.

IBM'e bilgi gönderdiğinizde, IBM ' e bu bilgileri size hiçbir sorumluluk yüklemeyen uygun gördüğü yöntemlerle kullanması ya da dağıtması için münhasır olmayan bir hak verirsiniz.

© Copyright International Business Machines Corporation 2007, 2024.

İçindekiler

Uygulama başvurusu geliştirilmesi.....	7
MQI uygulamaları başvurusu.....	7
Kod örnekleri.....	8
Değişmezler.....	61
MQI 'da kullanılan veri tipleri.....	234
İşlev çağrıları.....	618
Nesnelerin öznitelikleri.....	789
Dönüş kodları.....	863
MQI seçeneklerinin geçerliliğini denetleme kuralları.....	864
Kuyruğa alınan yayınlama/abone olma komut iletileri.....	867
Makine kodlamaları.....	888
Rapor seçenekleri ve ileti işaretleri.....	891
Veri dönüştürme çıkışı.....	895
MQRFH2 öğeleri olarak belirtilen özellikler.....	917
Kod sayfası dönüştürme.....	926
64 bitlik platformlarda kodlama standartları.....	976
IBM i Uygulama Programlama Başvurusu (ILE/RPG).....	980
IBM i üzerindeki veri tipi açıklamaları.....	981
IBM i üzerinde işlev çağrıları.....	1230
IBM i üzerindeki nesnelerin öznitelikleri.....	1344
Uygulamalar.....	1390
IBM i (ILE RPG) için dönüş kodları.....	1402
IBM i (ILE RPG) için MQI seçeneklerini doğrulama kuralları.....	1403
IBM i üzerinde makine kodlamaları.....	1406
IBM i üzerinde rapor seçenekleri ve ileti işaretleri.....	1409
IBM i üzerinde veri dönüştürme.....	1412
IBM i üzerinde dönüştürme işlemi.....	1412
IBM i üzerinde işleme kuralları.....	1414
IBM i üzerinde rapor iletilerinin dönüştürülmesi.....	1417
IBM i üzerinde MQDXP (Data-conversion exit parameter).....	1418
IBM i üzerinde MQXCNCV (karakterleri dönüştür).....	1423
IBM i üzerinde MQCONVX (Veri dönüştürme çıkışı).....	1428
Kullanıcı çıkışları, API çıkışları ve kurulabilir hizmetler başvurusu.....	1431
MQIEP yapısı.....	1431
Veri dönüştürme çıkış başvurusu.....	1435
MQ_PUBLISH_EXIT-Çıkış yayınlama.....	1439
Kanal çıkışı çağrıları ve veri yapıları.....	1446
Küme iş yükü çıkış çağrısı ve veri yapıları.....	1509
API çıkış başvurusu.....	1533
Kurulabilir hizmetler arabirimi başvuru bilgileri.....	1593
IBM i üzerinde kurulabilir hizmetler arabirimi başvuru bilgileri.....	1655
IBM MQ .NET sınıfları ve arabirimleri.....	1694
MQAsyncStatus.NET sınıfı.....	1694
MQAuthenticationInformationRecord.NET sınıfı.....	1696
MQDestination.NET sınıfı.....	1697
MQEnvironment.NET sınıfı.....	1699
MQException.NET sınıfı.....	1701
MQGetMessageOptions.NET sınıfı.....	1702
MQManagedObject.NET sınıfı.....	1705
MQMessage.NET sınıfı.....	1707
MQProcess.NET sınıfı.....	1719
MQPropertyDescriptor.NET sınıfı.....	1722

MQPutMessageOptions.NET sınıfı.....	1723
MQQueue.NET sınıfı.....	1726
MQQueueManager.NET sınıfı.....	1733
MQSubscription.NET sınıfı.....	1746
MQTopic.NET sınıfı.....	1747
IMQObjectTrigger.NET arabirimi.....	1752
MQC.NET arabirimi.....	1753
.NET uygulamaları için karakter kümesi tanıtıcıları.....	1753
IBM MQ C++ sınıfları.....	1756
C++ ve MQI çapraz başvurusu.....	1757
ImqAuthenticationKayıt C++ sınıfı.....	1772
ImqBinary C++ sınıfı.....	1774
ImqCache C++ sınıfı.....	1776
ImqChannel C++ sınıfı.....	1779
ImqCICSBridgeHeader C++ sınıfı.....	1784
ImqDeadLetterHeader C++ sınıfı.....	1790
ImqDistributionListe C++ sınıfı.....	1793
ImqError C++ sınıfı.....	1794
ImqGetMessageOptions C++ sınıfı.....	1795
ImqHeader C++ sınıfı.....	1799
ImqIMSBridgeHeader C++ sınıfı.....	1800
ImqItem C++ sınıfı.....	1803
ImqMessage C++ sınıfı.....	1805
ImqMessageİzleyici C++ sınıfı.....	1811
ImqNamelist C++ sınıfı.....	1814
ImqObject C++ sınıfı.....	1816
ImqProcess C++ sınıfı.....	1821
ImqPutMessageOptions C++ sınıfı.....	1823
ImqQueue C++ sınıfı.....	1825
ImqQueueManager C++ sınıfı.....	1835
ImqReferenceHeader C++ sınıfı.....	1851
ImqString C++ sınıfı.....	1854
ImqTrigger C++ sınıfı.....	1859
ImqWorkÜstbilgi C++ sınıfı.....	1862
IBM MQ classes for JMS nesnelerinin özellikleri.....	1864
IBM MQ classes for JMS nesnelerinin özellikleri arasındaki bağımlılıklar.....	1867
UYGULAMAADI.....	1869
ZAMANUYUMSUZ KURAL Dışı DURUMU.....	1869
BALOPTIONS.....	1870
BALTIP TIPI.....	1871
BALTIMEOUT.....	1872
BROKERCDURSUBQ.....	1872
BROKERCCSUBQ.....	1873
BROKERCONQ.....	1873
BROKERDURSUBQ.....	1873
BROKERPUBQ.....	1874
BROKERPUBQMGR.....	1874
BROKERQMGR.....	1875
BROKERSUBQ.....	1875
ARALIKÇI.....	1876
CCDTURL.....	1876
CCSID.....	1877
Kanal.....	1877
Temizle.....	1878
TEMİZLEME.....	1878
BAĞLANTI ADAMI.....	1879
CLIENTRECONNECTOPTIONS.....	1879
CLIENTRECONNECTTIMEOUT.....	1880

İSTEMCIID.....	1881
KLONESUPP.....	1881
TAMAMLANDI.....	1881
ŞİRKET.....	1882
BİRLEŞTİR.....	1882
CONNTAG (ETIKET).....	1883
TANIM.....	1884
DIRECTAUTH.....	1884
Kodlama.....	1885
Son kullanma tarihi.....	1886
FAILIFQUIESCE.....	1886
ANASİSTEM ADI.....	1887
YERELADRES.....	1887
MAPNAMESTYLE.....	1888
MAXBUFFSIZE.....	1889
MDREAD.....	1889
MDYAZMA.....	1890
MDMSGCTX.....	1890
MSGBATCHSZ.....	1891
MSGBODY (DOSYA).....	1891
MSGREÇEKİŞME.....	1892
MSGSELECTION.....	1892
ÇOKLU YAYIN.....	1893
OPTIMIST YAYIN.....	1893
OUTCOMENOTIFICATION.....	1894
Kalıcılık.....	1894
POLLINGİT.....	1895
PORT.....	1895
Öncelik.....	1896
SÜREÇ SÜRESİ.....	1896
PROVIDERVERSION.....	1897
PROXYHOSTNAME.....	1899
PROXYPORT.....	1899
GENEL.....	1900
PUTASYNCALKUTSAL.....	1900
YÖNETİCİ.....	1901
kuyruk.....	1901
READAHEADALLOWED.....	1902
READAHEADCLOSEPOLICY.....	1902
RECEIVECCSID.....	1903
ALAVECONVERSION.....	1903
ALMA YALITMA.....	1904
YENİDEN ÇIK.....	1904
RECEXITINIT.....	1905
REPLYTOSTYLE.....	1905
İLK DURUMUNA GEÇİR.....	1906
SECEXIT.....	1906
SECEXITINIT.....	1907
SENDCHECKCOUNT.....	1907
SENDEXIT.....	1908
SENDEXITINIT.....	1908
FARKLILIKLI.....	1909
SPARSESUB.....	1909
ŞİFRELEME TAKIMI.....	1910
SSLCRL.....	1910
SSLFIPSREQUIRED.....	1911
SSLPEERNAME.....	1911
SSLRESETCOUNT.....	1912

STATREFRESHINT.....	1912
ALT MAĞAZA.....	1913
SYNCPOINTALLYAKALAR.....	1913
HEDEF.....	1914
TARGCLIENTEŞLEŞME.....	1914
TEMPMODEL.....	1915
TEMPQPREFIX.....	1915
TEMPTOPICPREFIX.....	1916
Konu.....	1916
Aktarım.....	1917
WILDCARDFORMAT.....	1917
ENCODING özelliği.....	1918
JMS nesnelerinin TLS özellikleri.....	1918
IBM MQ Message Service Client (XMS) for .NET başvuru.....	1919
.NET arabirimler.....	1919
XMS nesnelerinin özellikleri.....	2000
Managed File Transfer uygulama başvurusu geliştirilmesi.....	2064
Programları başlatmak için fteCreateTransfer yönteminin kullanılmasına ilişkin örnekler.....	2064
fteAnt : Ant görevlerini MFT içinde çalıştırma.....	2066
Özelleştirme başvurusu için MFT kullanıcı çıktıları.....	2090
MFT Agent komut kuyruğuna koyabileceğiniz iletilere ilişkin ileti biçimleri.....	2131
İleti alışverişi REST API başvurusu.....	2131
REST API Kaynak.....	2131
Özel notlar.....	2163
Programlama arabirimi bilgileri.....	2164
Ticari Markalar.....	2164

Uygulama başvurusu geliştirilmesi

IBM MQ uygulamalarınızı geliştirmenize yardımcı olması için bu bölümde sağlanan bağlantıları kullanın.

- [“MQI uygulamaları başvurusu” sayfa 7](#)
- [IBM i “IBM i Uygulama Programlama Başvurusu \(ILE/RPG\)” sayfa 980](#)
- [IBM i “IBM i üzerinde veri dönüştürme” sayfa 1412](#)
- [“Kullanıcı çıkışları, API çıkışları ve kurulabilir hizmetler başvurusu” sayfa 1431](#)
- [“IBM MQ .NET sınıfları ve arabirimleri” sayfa 1694](#)
- [“IBM MQ C++ sınıfları” sayfa 1756](#)
- [“IBM MQ classes for JMS nesnelerinin özellikleri” sayfa 1864](#)
- [“İleti alışverişi REST API başvurusu” sayfa 2131](#)

İlgili görevler

[Uygulama geliştirilmesi](#)

İlgili başvurular

[Java kitaplıkları için IBM MQ sınıfları](#)

[JMS için IBM MQ sınıfları](#)

MQI uygulamaları başvurusu

İleti Kuyruğu Arabirimi (MQI) uygulamalarınızı geliştirmenize yardımcı olması için bu kısımda sağlanan bağlantıları kullanın.

- [“Kod örnekleri” sayfa 8](#)
- [“Değişmezler” sayfa 61](#)
- [“MQI 'da kullanılan veri tipleri” sayfa 234](#)
- [“İşlev çağrıları” sayfa 618](#)
- [“Nesnelerin öznitelikleri” sayfa 789](#)
- [“Dönüş kodları” sayfa 863](#)
- [“MQI seçeneklerinin geçerliliğini denetleme kuralları” sayfa 864](#)
- [“Makine kodlamaları” sayfa 888](#)
- [“Rapor seçenekleri ve ileti işaretleri” sayfa 891](#)
- [“Veri dönüştürme çıkışı” sayfa 895](#)
- [“MQRFH2 öğeleri olarak belirtilen özellikler” sayfa 917](#)
- [“Kod sayfası dönüştürme” sayfa 926](#)

İlgili kavramlar

[“Kullanıcı çıkışları, API çıkışları ve kurulabilir hizmetler başvurusu” sayfa 1431](#)

Kullanıcı çıkışlarınızı, API çıkışlarınızı ve kurulabilir hizmet uygulamalarını geliştirmenize yardımcı olması için bu bölümdeki linformation yardımcı programını kullanın:

İlgili görevler

[Uygulama geliştirilmesi](#)

İlgili başvurular

[“IBM MQ .NET sınıfları ve arabirimleri” sayfa 1694](#)

IBM MQ .NET sınıfları ve arabirimleri alfabetik olarak listelenir. Özellikler, yöntemler ve oluşturucular açıklanır.

“IBM MQ C++ sınıfları” sayfa 1756

IBM MQ C++ sınıfları IBM MQ Message Queue Interface (MQI) olanağını kapsıyor. Bu sınıfların tümünü kapsayan tek bir C++ üstbilgi dosyası (**imqi.hpp**) vardır.

[Java kitaplıkları için IBM MQ sınıfları](#)

[IBM MQ İlgili sınıflar JMS](#)

Kod örnekleri

İş gereksinimlerinizi karşılayan görevleri gerçekleştirmek için bu bölümdeki başvuru bilgilerini kullanın.

C dili örnekleri

Bu konu derlemi çoğunlukla IBM MQ for z/OS örnek uygulamalarından alınır. Bunlar, belirtildiği durumlar dışında, tüm platformlar için geçerlidir.

Kuyruk yöneticisine bağlanma

Bu örnek, z/OS toplu iş kümesinde bir programı kuyruk yöneticisine bağlamak için MQCONN çağrısının nasıl kullanılacağını gösterir.

Bu alma işlemi, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından (program CSQ4BCA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```
#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;
    }
    ...
}
```

Kuyruk yöneticisiyle bağlantıyı kesme

Bu örnek, z/OS toplu iş kümesinde bir programın kuyruk yöneticisiyle bağlantısını kesmek için MQDISC çağrılarının nasıl kullanılacağını gösterir.

Bu kod alımında kullanılan değişkenler, “Kuyruk yöneticisine bağlanma” sayfa 8 içinde ayarlanan değişkenlerdir. Bu alma işlemi, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından

(program CSQ4BCA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. Örnek yordam programları (z/OS dışında platformlar).

```
:
/*
/* 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;
}
:
```

Dinamik kuyruk yaratılması

Bu örnek, dinamik kuyruk yaratmak için MQOPEN çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ for z/OS ile birlikte gönderilen Mail Manager örnek uygulamasından (program CSQ4TCD1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. Örnek yordam programları (z/OS dışında platformlar).

```
:
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 ObjDesc; /* Object descriptor */
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;
}
...

```

Var olan bir kuyruğun açılması

Bu örnek, önceden tanımlanmış bir kuyruğu açmak için MQOPEN çağrısının nasıl kullanılacağını gösterir.

Bu alma işlemi, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından (program CSQ4BCA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

#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 };
/* Object descriptor */
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))
{
printf(pBuff, MESSAGE_4_E,
      ERROR_IN_MQOPEN, CompCode, Reason);
PrintLine(pBuff);
RetCode = CSQ4_ERROR;
goto AbnormalExit1; /* disconnect processing */
}
...
} /* end of main */

```

Kuyruğun kapatılması

Bu örnek, bir kuyruğu kapatmak için MQCLOSE çağrısının nasıl kullanılacağını gösterir.

Bu alma işlemi, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından (program CSQ4BCA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
/*                                     */
/* Close the queue.                   */
/* Test the output of the close 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;
}
:

```

MQPUT kullanılarak ileti konması

Bu örnek, kuyruğa ileti koymak için MQPUT çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz. Örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışındaki platformlar\)](#) **z/OS** ve [IBM MQ for z/OS için örnek programlar](#).

```

:
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 */
}
}
}

```

MQPUT1 kullanılarak ileti konması

Bu örnek, bir kuyruğu açmak, kuyruğa tek bir ileti koymak ve kuyruğu kapatmak için MQPUT1 çağrısının nasıl kullanılacağını gösterir.

Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Kredi Denetimi örnek uygulamasından (program CSQ4CCB5) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları](#) (z/OS dışında platformlar).

```

:
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      */
MQPMO    PutMsgOpts = {MQPMO_DEFAULT};
          /* Put Message Options   */
CSQ4BQRM PutBuffer;      /* Message structure      */
MQLONG   PutBuffLen = 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;
}
...

```

İleti alma

Bu örnek, kuyruktan ileti kaldırmak için MQGET çağrısını nasıl kullanacağınızı gösterir.

Bu alma işlemi, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından (program CSQ4BCA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

#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 */
    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 */

```

Bekleme seçeneğini kullanarak ileti alma

Bu örnek, MQGET çağrısının bekleme seçeneğinin nasıl kullanılacağını gösterir.

Bu kod kesilmiş iletileri kabul eder. Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Kredi Denetimi örnek uygulamasından (program CSQ4CCB5) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
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;
    MsgBuffLen = 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();
}
:

```

Sinyali kullanarak ileti alma

Sinyal yalnızca IBM MQ for z/OS ile kullanılabilir.

Bu örnek, bir kuyruğa uygun bir ileti geldiğinde size bildirim gönderilmesini sağlamak üzere bir sinyal ayarlamak için MQGET çağrısına nasıl geçileceğini gösterir. Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

:
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_ecb, work_ecb;
    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_ecb                    = 0;
    GetMsgOpts.Signal1      = &q_ecb;
    /*-----*/
    /* 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;
}
}

```

Bir nesnenin özniteliklerini sorma

Bu örnek, bir kuyruğun özniteliklerini sorgulamak için MQINQ çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ for z/OS ile birlikte sağlanan Kuyruk Öznitelikleri örnek uygulamasından (program CSQ4CCC1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

#include <cmqc.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,

```



```

/*      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 */

```

MQSTAT ile durum bilgileri alınıyor

Bu örnek, zamanuyumsuz bir MQPUT ' nin nasıl verileceğini ve MQSTAT ile durum bilgilerinin nasıl alınacağını gösterir.

Bu alma Calling MQSTAT örnek uygulamasından alınır (program amqsapt0) IBM MQ for Windows sistemleriyle birlikte sağlanır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

/*****
/*
/* 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
/*
/*
/*
/*****
/*
/*   AMQSAPTO 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 AMQSAPTO 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 AMQSAPTO                                         */
/*
/*
/*****
printf("Sample AMQSAPTO end\n");
return(0);
}

```

COBOL örnekleri

Bu konu derlemi, IBM MQ for z/OS örnek uygulamalarından alınır. Bunlar, belirtildiği durumlar dışında, tüm platformlar için geçerlidir.

Kuyruk yöneticisine bağlanma

Bu örnek, z/OS toplu iş kümesinde bir programı kuyruk yöneticisine bağlamak için MQCONN çağrısının nasıl kullanılacağını gösterir.

Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından (program CSQ4BVA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

* -----*
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.
:

```

Kuyruk yöneticisiyle bağlantıyı kesme

Bu örnek, z/OS toplu iş kümesinde bir programın kuyruk yöneticisiyle bağlantısını kesmek için MQDISC çağrılarının nasıl kullanılacağını gösterir.

Bu kod alımında kullanılan değişkenler, [“Kuyruk yöneticisine bağlanma”](#) sayfa 23’ünde ayarlanan değişkenlerdir. Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından (program CSQ4BVA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
*
*   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.
:

```

Dinamik kuyruk yaratılması

Bu örnek, dinamik kuyruk yaratmak için MQOPEN çağrısının nasıl kullanılacağını gösterir.

Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Kredi Denetimi örnek uygulamasından (program CSQ4CVB1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
* -----*
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 = MQ00-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
*

```

Var olan bir kuyruğun açılması

Bu örnek, var olan bir kuyruğu açmak için MQOPEN çağrısıyla nasıl kullanılacağını gösterir.

Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından (program CSQ4BVA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
* -----*

```

```

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

```

Kuyruğun kapatılması

Bu örnek, MQCLOSE çağrısının nasıl kullanılacağını gösterir.

Bu kod alımında kullanılan değişkenler, “Kuyruk yöneticisine bağlanma” sayfa 23’ünde ayarlanan değişkenlerdir. Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek uygulamasından (program CSQ4BVA1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
*
*   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.
*

```

MQPUT kullanılarak ileti konması

Bu örnek, bağlam kullanılarak MQPUT çağrısının nasıl kullanılacağını gösterir.

Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Kredi Denetimi örnek uygulamasından (program CSQ4CVB1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
* -----*
  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.

```

MQPUT1 kullanılarak ileti konması

Bu örnek, MQPUT1 çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ for z/OS ile birlikte sağlanan Kredi Denetimi örnek uygulamasından (program CSQ4CVB5) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
* -----*
* 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-SYNCPOINT +
                        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.
*

```

İleti alma

Bu örnek, kuyruktan ileti kaldırmak için MQGET çağrısını nasıl kullanacağınızı gösterir.

Bu veri alımı, IBM MQ for z/OS ile birlikte sağlanan Kredi Denetimi örnek uygulamasından (program CSQ4CVB1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

:

```

* -----*
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.

```

Bekleme seçeneğini kullanarak ileti alma

Bu örnek, MQGET çağrısıyla bekleme seçeneğinin nasıl kullanılacağını ve kesilmiş iletilerin nasıl kabul edileceğini gösterir.

Bu alma, IBM MQ for z/OS ile birlikte sağlanan Kredi Denetimi örnek uygulamasından (program CSQ4CVB5) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
* -----*
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 MQOD-OBJECTNAME TO M02-OBJECTNAME
  PERFORM RECORD-CALL-ERROR
END-IF.
:

```

Sinyali kullanarak ileti alma

Bu örnek, MQGET çağrısıyla sinyalin nasıl kullanılacağını gösterir. Bu alma, IBM MQ for z/OS ile birlikte sağlanan Kredi Denetimi örnek uygulamasından (program CSQ4CVB2) alınır.

Sinyal yalnızca IBM MQ for z/OS ile kullanılabilir.

```

:
* -----*
* 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 MQ0D-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
*
  :

```

Bir nesnenin özniteliklerini sorma

Bu örnek, bir kuyruğun özniteliklerini sorgulamak için MQINQ çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ for z/OS ile birlikte verilen Kuyruk Öznitelikleri örnek uygulamasından (program CSQ4CVC1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için bkz. [Örnek yordam programları \(z/OS dışında platformlar\)](#).

```

:
* -----*
WORKING-STORAGE SECTION.
* -----*
*
* W02 - MQM API fields

```


Bu alma, IBM MQ for z/OS ile birlikte verilen Kuyruk Öznitelikleri örnek uygulamasından (program CSQ4CVC1) alınır. Diğer altyapılardaki örnek uygulamaların adları ve yerleri için Örnek yordam programları (z/OS dışında platformlar) başlıklı konuya bakın.

```

:
* -----*
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 çevirici dili örnekleri

Bu konu derlemi çoğunlukla IBM MQ for z/OS örnek uygulamalarından alınır.

Kuyruk yöneticisine bağlanma

Bu örnek, z/OS toplu iş kümesinde bir programı kuyruk yöneticisine bağlamak için MQCONN çağrısının nasıl kullanılacağını gösterir.

Bu alma işlemi, IBM MQ for z/OS ile birlikte sağlanan Göz At örnek programından (CSQ4BAA1) alınır.

```

:
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 : MAINPARG *
*****
MAINPARG 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,MOVEPARG Move the data
*
*****
* EXECUTES *
*****
MOVEPARG 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
*

```

Kuyruk yöneticisiyle bağlantıyı kesme

Bu örnek, z/OS toplu iş kümesinde bir programın kuyruk yöneticisiyle bağlantısını kesmek için MQDISC çağrılarının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

:
*
*   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

```

Dinamik kuyruk yaratılması

Bu örnek, dinamik kuyruk yaratmak için MQOPEN çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

:
*
*   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(MQ00_OUTPUT) OPEN FOR OUTPUT AND
A R5,=AL4(MQ00_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
MQQA 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 ,(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L LIST FORM
OF CALL
MACRO
*
*
*
END

```

Var olan bir kuyruğun açılması

Bu örnek, önceden tanımlanmış bir kuyruğu açmak için MQOPEN çağrısının nasıl kullanılacağını gösterir.

İki seçeneğin nasıl belirtileceğini gösterir. Bu alma, IBM MQile verilen örnek uygulamalardan alınmaz.

```

*
*
* 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,MQ00_INPUT_EXCLUSIVE OPEN FOR MQGET CALLS
*
ST R5,OPTIONS
*
* ISSUE MQI OPEN REQUEST USING REENTRANT FORM
* OF CALL MACRO

```

```

*
      CALL MQOPEN,
          (HCONN,
           WOD,
           OPTIONS,
           HOBJ,
           COMPCODE,
           REASON),VL,MF=(E,CALLLST)
          X
          X
          X
          X
          X
          X
*
      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
      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
*
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

```

Kuyruğun kapatılması

Bu örnek, bir kuyruğu kapatmak için MQCLOSE çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

:
*
* 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,
          (HCONN,
           HOBJ,
           OPTIONS,
           COMPCODE,
           REASON),
          VL,MF=(E,CALLLST)
          X
          X
          X
          X
          X
*
      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),VL,MF=L
*
HCONN  DS    F
HOBJ   DS    F
OPTIONS DS    F
COMPCODE DS  F
REASON DS    F
*
*
LEG4   EQU  *-WKEG4
      END

```

MQPUT kullanılarak ileti konması

Bu örnek, kuyruğa ileti koymak için MQPUT çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

:
*      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

```

MQPUT1 kullanılarak ileti konması

Bu örnek, bir kuyruğu açmak, kuyruğa tek bir ileti koymak ve kuyruğu kapatmak için MQPUT1 çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

:
*
*   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
*
CALLLST  CALL ,(0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
        :
        END

```

İleti alma

Bu örnek, kuyruktan ileti kaldırmak için MQGET çağrısını nasıl kullanacağınızı gösterir.

Bu alma, IBM MQile verilen örnek uygulamalardan alınmaz.

```

:
*
*      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 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
L R5,=AL4(MQGMO_WAIT)
A R5,=AL4(MQGMO_ACCEPT_TRUNCATED_MSG)
ST R5,WGMO_OPTIONS
MVC WGMO_WAITINTERVAL,TWO_MINUTES WAIT UP TO TWO
MINUTES BEFORE
FAILING THE
CALL

*
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,CALLST)

*
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.

*
LA R5,MQRC_NO_MSG_AVAILABLE IT MUST BE AN ERROR.
IS IT DUE TO AN EMPTY
C R5,REASON QUEUE?
BE NOMSG YES, SO HANDLE THE ERROR
B BADCALL NO, SO GO TO ERROR ROUTINE

*
CHECK_W DS 0H
LA R5,MQRC_TRUNCATED_MSG_ACCEPTED IS THIS A
TRUNCATED
C R5,REASON MESSAGE?
BE GETOK YES, SO GO AND PROCESS.
B BADCALL NO, SOME OTHER WARNING

*
NOMSG DS 0H
:
GETOK DS 0H
:

BADCALL DS 0H
:
*
* CONSTANTS
*
CMQMDA DSECT=NO,LIST=YES
CMQGMOA 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
WGMO     CMQGMOA DSECT=NO,LIST=NO
*
CALLLST  CALL , (0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*
      :
      END

```

Sinyali kullanarak ileti alma

Bu örnek, bir kuyruğa uygun bir ileti geldiğinde size bildirim gönderilmesini sağlamak üzere bir sinyal ayarlamak için MQGET çağrısına nasıl geçileceğini gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

:
*
*   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,MQGMO_SET_SIGNAL
ST     R5,WGMO_OPTIONS
MVC    WGMO_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,CALLST)
*
    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?
    B0 CHECKSIG              YES, SO GO AND CHECK WHY
    B  DOWORK                NO, SO GO AND DO MORE WORK
*
NOMSG DS 0H
    TM SIG_ECB,X'40'
*
GETOK DS 0H
    TM SIG_ECB,X'40'
*
BADCALL DS 0H
    TM SIG_ECB,X'40'
*
*
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 CMQMOA DSECT=NO,LIST=NO
*
CALLST CALL ,(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),VL,MF=L
*

```



```

*
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?
        :

```

PL/I örnekleri

PL/I kullanımı yalnızca z/OS tarafından desteklenir. Bu konu derlemesi, PL/I örneklerini kullanan teknikleri gösterir.

Kuyruk yöneticisine bağlanma

Bu örnek, z/OS toplu iş kümesinde bir programı kuyruk yöneticisine bağlamak için MQCONN çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

%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_MQMNAME CHAR(48);
        :
        /*****
        /* WORKING STORAGE DECLARATIONS          */
        /*****
        DCL MQMNAME          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                       */
        /*****
        MQMNAME = ' ';
        MQMNAME = SUBSTR(PARAM_MQMNAME,1,PARAM_LENGTH);
        :
        /*****
        /* CONNECT FROM THE QUEUE MANAGER        */
        /*****
        CALL MQCONN (MQMNAME,          /* 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;

```

Kuyruk yöneticisiyle bağlantıyı kesme

Bu örnek, z/OS toplu iş kümesinde bir programın kuyruk yöneticisiyle bağlantısını kesmek için MQDISC çağrılarının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

%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;

```

Dinamik kuyruk yaratılması

Bu örnek, dinamik kuyruk yaratmak için MQOPEN çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```
%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;
```

Var olan bir kuyruğun açılması

Bu örnek, var olan bir kuyruğu açmak için MQOPEN çağrısıyla nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```
%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;

```

Kuyruğun kapatılması

Bu örnek, MQCLOSE çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

%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;

```

MQPUT kullanılarak ileti konması

Bu örnek, bağlam kullanılarak MQPUT çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

%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;

```

MQPUT1 kullanılarak ileti konması

Bu örnek, MQPUT1 çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```
%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 = ' ';
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;

```

İleti alma

Bu örnek, kuyruktan ileti kaldırmak için MQGET çağrısını nasıl kullanacağınızı gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```
%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,
            BUFFLEN,
            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;
```

Bekleme seçeneğini kullanarak ileti alma

Bu örnek, MQGET çağrısıyla bekleme seçeneğinin nasıl kullanılacağını ve kesilmiş iletilerin nasıl kabul edileceğini gösterir.

Bu alma, IBM MQ'ile verilen örnek uygulamalardan alınmaz.

```
%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;

```

Sinyali kullanarak ileti alma

MQGET çağrısıyla sinyalin nasıl kullanılacağını gösteren bir kod ayıklaması.

Sinyal yalnızca IBM MQ for z/OS ile kullanılabilir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

        %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,
            BUFFLEN,
            BUFFER,
            DATALEN,
            COMPCODE,
            REASON);

END GET_MSG;

NO_MSG: PROC;
:
END NO_MSG;

```

Bir nesnenin özniteliklerini sorma

Bu örnek, bir kuyruğun özniteliklerini sorgulamak için MQINQ çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

%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;

```

Kuyruğun özniteliklerini ayarlama

Bu örnek, bir kuyruğun özniteliklerini değiştirmek için MQSET çağrısının nasıl kullanılacağını gösterir.

Bu alma, IBM MQ ile verilen örnek uygulamalardan alınmaz.

```

%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;

```

Değişmezler

İş gereksinimlerinizi karşılayan görevleri gerçekleştirmek için bu bölümdeki başvuru bilgilerini kullanın.

IBM MQ COPY, üstbilgi, içerme ve modül dosyaları

Bu bilgiler, genel kullanım programlama arabirimi bilgileridir.

Bu kısım, MQI 'ı çeşitli programlama dilleri için kullanmanıza yardımcı olacak bilgileri içerir.

C üstbilgi dosyaları

Üstbilgi dosyaları, MQI kullanan C uygulama programlarını yazmanıza yardımcı olmak için sağlar.

C üstbilgi dosyaları aşağıdaki tabloda özetlenir:

<i>Çizelge 1. C üstbilgi dosyaları-çağrı prototipleri, veri tipleri, dönüş kodları, sabitler ve yapılar</i>					
Dosya adı	Açıklama	IBM i	AIX and Linux® sistemleri	Windows	z/OS
Çağrı prototipleri, veri tipleri, dönüş kodları, sabitler ve yapılar					
CMQC	MQI tanımlamaları	C	C	C	C
CMQBC	MQAI tanımlamaları	C	C	C	
CMQEC	Arabirim Giriş Noktaları tanımı (CMQC, CMQXC ve CMQZC içerir)		C	C	
CMQFC	PCF tanımlamaları	C	C	C	C
CMQPS	Tanımları yayınlama/abone ol	C	C	C	C
CMQXC	Kanal ve çıkış tanımları	C	C	C	C
CMQZC	Kurulabilir hizmet tanımlamaları	C	C	C	
Anahtar: C= Sağlanan dosyalar					

COBOL COPY dosyaları

MQI kullanan COBOL uygulama programlarını yazmanıza yardımcı olmak için çeşitli COPY dosyaları sağlar.

<i>Çizelge 2. COBOL kopya dosyaları-dönüş kodları, sabitler ve yapılar</i>					
Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
Dönüş kodları ve değişmezleri					
CMQx	MQI tanımlamaları	V	V	V	V
CMQCFx	PCF tanımlamaları	V	V	V	V
CMQPSx	Tanımları yayınlama/abone ol	V	V	V	V
CMQXx	Kanal ve çıkış tanımları	V	V	V	V
Yapılar					
CMQAIRx	MQAIR-Kimlik doğrulama bilgileri kaydı		V L	V L	
CMQBOX	MQBO-Başlangıç seçenekleri	V L	V L	V L	

Çizelge 2. COBOL kopya dosyaları-dönüş kodları, sabitler ve yapılar (devamı var)

Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
CMQCDx	MQCD-Kanal tanımlaması	V L	V L	V L	V L
CMQCFBFx	MQCFBF-PCF byte dizgisi süzgeç değiştirgesi	V L	V L	V L	V L
CMQCFBSx	MQCFBS-PCF byte dizgisi değiştirgesi	V L	V L	V L	V L
CMQCFGRX	MQCFGR-PCF grup değiştirgesi	V L	V L	V L	V L
CMQCFHx	MQCFH-PCF üstbilgisi	V L	V L	V L	V L
CMQCFIFx	MQCFIF-PCF tamsayı süzgeci değiştirgesi	V L	V L	V L	V L
CMQCFILx	MQCFIL-PCF tamsayı listesi değiştirgesi	V L	V L	V L	V L
CMQCFINx	MQCFIN-PCF tamsayı değiştirgesi	V L	V L	V L	V L
CMQCFSFx	MQCFSF-PCF dizgi süzgeci değiştirgesi	V L	V L	V L	V L
CMQCFSLx	MQCFSL-PCF dizgi listesi değiştirgesi	V L	V L	V L	V L
CMQCFSTx	MQCFST-PCF dizgi değiştirgesi	V L	V L	V L	V L
CMQCFXLx	MQCFIL64 -PCF 64 bit tamsayı listesi değiştirgesi	V L	V L	V L	V L
CMQCFXNx	MQCFIN64 -PCF 64 bit tamsayı değiştirgesi	V L	V L	V L	V L
CMQCHRVx	MQCHARV-Değişken uzunluklu dizgi	V L	V L	V L	V L
CMQCIHx	MQCIH- CICS bridge üstbilgisi	V L	V L	V L	V L
CMQCNOx	MQCNO-Bağlantı seçenekleri	V L	V L	V L	V L
CMQCSPx	MQCSP-Güvenlik değiştirgeleri	V L	V L	V L	V L
CMQCXPx	MQCXP-Kanal çıkış değiştirgeleri	V L			V L
CMQDHx	MQDH-Dağıtım üstbilgisi	V L	V L	V L	V L
CMQDLHx	MQDLH-Ölmeyen harf üstbilgisi	V L	V L	V L	V L
CMQDXPx	MQDXP-Veri dönüştürme çıkış parametreleri	V L		V L	
CMQEPHx	MQEPH-Gömülü PCF üstbilgisi	V L	V L	V L	V L
CMQGMox	MQGMO-İleti seçeneklerini al	V L	V L	V L	V L
CMQIIHx	MQIIH- IMS bilgi üstbilgisi	V L	V L	V L	V L
CMQMDx	MQMD-İleti tanımlayıcı	V L	V L	V L	V L
CMQMD1x	MQMD1 -İleti tanımlayıcı sürüm 1	V L	V L	V L	V L
CMQMD2x	MQMD2 -İleti tanımlayıcı sürüm 2	V L	V L	V L	V L
CMQMDEx	MQMDE-İleti tanımlayıcı genişletildi	V L	V L	V L	V L

Çizelge 2. COBOL kopya dosyaları-dönüş kodları, sabitler ve yapılar (devamı var)

Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
CMQODx	MQOD-Nesne tanımlayıcı	V L	V L	V L	V L
CMQORx	MQOR-Nesne kaydı	V L	V L	V L	V L
CMQPMOx	MQPMO-İleti koyma seçenekleri	V L	V L	V L	V L
CMQRFHx	MQRFH-Kurallar ve biçimlendirme üstbilgisi	V L	V L	V L	V L
CMQRFH2x	MQRFH2 -Kurallar ve biçimlendirme üstbilgisi 2	V L	V L	V L	V L
CMQRMHx	MQRMH-Başvuru iletisi üstbilgisi	V L	V L	V L	V L
CMQRRx	MQRR-Yanıt kaydı	V L	V L	V L	
CMQSCOx	MQSCO-TLS yapılandırma seçenekleri		V L	V L	
CMQTMx	MQTM-Tetikleyici iletisi	V L		V L	V L
CMQTMcx	MQTMc-Tetikleyici ileti karakteri	V L	V L		
CMQTM2x	MQTM2 -Tetikleyici ileti 2 karakteri	V L	V L	V L	V L
CMQWIHx	MQWIH-İş bilgileri üstbilgisi	V L	V L	V L	V L
CMQXQHx	MQXQH-İletim kuyruğu üstbilgisi	V L	V L	V L	V L

Anahtar:

- İlk değerleri sağlanan dosyalar, x = V
- Başlangıç değeri sağlanmayan dosyalar, x = L

z/OS PL/I içerme dosyaları

PL/I programlama dili için bir dizi INCLUDE dosyası sağlar. Bu dosyalar yalnızca z/OS üzerinde kullanılabilir.

Çizelge 3. PL/I include files-veri tipleri, dönüş kodları, sabitler ve yapılar

Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
Veri tipleri, dönüş kodları, sabitler ve yapılar					
CMQP	MQI tanımlamaları				P
CMQCFP	PCF tanımlamaları				P
CMQEPP	Giriş noktası tanımlamaları				P
CMQPSP	Tanımları yayınla/abone ol				P
CMQXP	Kanal ve çıkış tanımları				P

Anahtar: P= Dosya sağlandı

IBM i RPG kopyalama dosyaları

RPG COPY dosyaları RPG programlama dili için sağlar. Bu dosyalar yalnızca IBM üzerinde kullanılabilir.

Çizelge 4. RPG kopya dosyaları-dönüş kodları, sabitler ve yapılar

Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
Dönüş kodları ve değişmezleri					
CMQx	MQI tanımlamaları	G R			
CMQCFx	PCF tanımlamaları	G			
CMQPSx	Tanımları yayınla/abone ol	G			
CMQXx	Kanal ve çıkış tanımları	G R			
Yapılar					
CMQBOX	MQBO-Başlangıç seçenekleri	G H			
CMQCDx	MQCD-Kanal tanımlaması	G H R			
CMQCFBFx	MQCFBF-PCF byte dizgisi süzgeç değiştirgesi	G H			
CMQCFBSx	MQCFBS-PCF byte dizgisi değiştirgesi	G H			
CMQCFGRX	MQCFGR-PCF grup değiştirgesi	G H			
CMQCFHx	MQCFH-PCF üstbilgisi	G H			
CMQCFIFx	MQCFIF-PCF tamsayı süzgeci değiştirgesi	G H			
CMQCFILx	MQCFIL-PCF tamsayı listesi değiştirgesi	G H			
CMQCFINx	MQCFIN-PCF tamsayı değiştirgesi	G H			
CMQCFSFx	MQCFSF-PCF dizgi süzgeci değiştirgesi	G H			
CMQCFSLx	MQCFSL-PCF dizgi listesi değiştirgesi	G H			
CMQCFSTx	MQCFST-PCF dizgi değiştirgesi	G H			
CMQCFXLx	MQCFIL64 -PCF 64 bit tamsayı listesi değiştirgesi	G H			
CMQCFXNx	MQCFIN64 -PCF 64 bit tamsayı değiştirgesi	G H			
CMQCHARVx	MQCHARV-Değişken uzunluklu dizgi	G H			
CMQCIHx	MQCIH- CICS bridge üstbilgisi	G H			
CMQCNOx	MQCNO-Bağlantı seçenekleri	G H			
CMQCSPx	MQCSP-Güvenlik değiştirgeleri	G H			
CMQCXPx	MQCXP-Kanal çıkış değiştirgeleri	G H R			
CMQDHx	MQDH-Dağıtım üstbilgisi	G H R			
CMQDLHx	MQDLH-Ölmeyen harf üstbilgisi	G H R			
CMQDXPx	MQDXP-Veri dönüştürme çıkış parametreleri	G H R			

Çizelge 4. RPG kopya dosyaları-dönüş kodları, sabitler ve yapılar (devamı var)

Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
CMQEPHx	MQEPH-Gömülü PCF üstbilgisi	G H			
CMQGMOx	MQGMO-İleti seçeneklerini al	G H R			
CMQIIHx	MQIIH- IMS bilgi üstbilgisi	G H R			
CMQMDx	MQMD-İleti tanımlayıcı	G H R			
CMQMD1x	MQMD1 -İleti tanımlayıcı sürüm 1	G H R			
CMQMD2x	MQMD2 -İleti tanımlayıcı sürüm 2	G H			
CMQMDEx	MQMDE-İleti tanımlayıcı genişletildi	G H R			
CMQODx	MQOD-Nesne tanımlayıcı	G H R			
CMQORx	MQOR-Nesne kaydı	G H R			
CMQPMOx	MQPMO-İleti koyma seçenekleri	G H R			
CMQXPx	MQXP-Yayınla/Abone Ol yönlendirme çıkış parametreleri	G H			
CMQRFHx	MQRFH-Kurallar ve biçimlendirme üstbilgisi	G H			
CMQRFH2x	MQRFH2 -Kurallar ve biçimlendirme üstbilgisi 2	G H			
CMQRMHx	MQRMH-Başvuru iletisi üstbilgisi	G H R			
CMQRRx	MQRR-Yanıt kaydı	G H R			
CMQTMx	MQTM-Tetikleyici iletisi	G H R			
CMQTMCx	MQTMC-Tetikleyici ileti karakteri	G H R			
CMQTM2x	MQTM2 -Tetikleyici ileti 2 karakteri	G H R			
CMQWIHx	MQWIH-İş bilgileri üstbilgisi	G H			
CMQXQHx	MQXQH-İletim kuyruğu üstbilgisi	G H R			

Anahtar:

- Durağan bağlantı için dosya, kullanıma hazırlandı, sağlanan x = G
- Durağan bağlantı için dosya kullanıma hazırlanmadı, sağlanan x = H
- Dinamik bağlantı için dosya, kullanıma hazırlandı, sağlandı, x = R

Windows Visual Basic modül dosyaları

Üstbilgi (ya da form) dosyaları, MQI kullanan Visual Basic uygulama programlarını yazmanıza yardımcı olmak için sağlanır. Bu üstbilgi dosyaları yalnızca 32 bit sürümlerinde sağlanır.

Çizelge 5. Visual Basic modülü dosyaları-çağrı bildirimleri, veri tipleri, dönüş kodları, sabitler ve yapılar

Dosya adı	Açıklama	IBM i	AIX and Linux sistemleri	Windows	z/OS
Çağrı bildirimleri, veri tipleri, dönüş kodları, sabitler ve yapılar					
CMQB	MQI tanımlamaları			B	

Çizelge 5. Visual Basic modülü dosyaları-çağrı bildirimleri, veri tipleri, dönüş kodları, sabitler ve yapılar (devamı var)

Dosya adı	Açıklama	IBM i	AIX and Linux sistemleri	Windows	z/OS
CMQBB	MQAI tanımlamaları			B	
CMQCFB	PCF tanımlamaları			B	
CMQXB	Kanal ve çıkış tanımları			B	

Anahtar: B= Dosya sağlandı

z/OS z/OS çevirici COPY dosyaları

MQI kullanan z/OS Assembler uygulama programlarını yazmanıza yardımcı olmak için çeşitli COPY dosyaları sağlanır.

Çizelge 6. z/OS Çevirici kopyalama dosyaları-veri tipleri, dönüş kodları, sabitler ve yapılar

Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
Veri tipleri, dönüş kodları ve değişmezler					
CMQA	MQI tanımlamaları				A
CMQCFA	PCF tanımlamaları				A
CMQPSA	Tanımları yayınla/abone ol				A
CMQVERA	Yapı sürümü denetimi				A
CMQXA	Kanal ve çıkış tanımları				A
Yapılar					
CMQCDA	MQCD-Kanal tanımlaması				
CMQCFBFA	MQCFBF-PCF byte dizgisi süzgeç değiştirgesi				
CMQCFBSA	MQCFBS-PCF byte dizgisi değiştirgesi				A
CMQCFGRA	MQCFGR-PCF grup değiştirgesi				A
CMQCFHA	MQCFH-PCF üstbilgisi				A
CMQCFIFA	MQCFIF-PCF tamsayı süzgeci değiştirgesi				A
CMQCFILA	MQCFIL-PCF tamsayı listesi değiştirgesi				A
CMQCFINA	MQCFIN-PCF tamsayı değiştirgesi				A
CMQCFSA	MQCFSF-PCF dizgi süzgeci değiştirgesi				A
CMQCFSLA	MQCFSL-PCF dizgi listesi değiştirgesi				A
CMQCFSTA	MQCFST-PCF dizgi değiştirgesi				A

Çizelge 6. z/OS Çevirici kopyalama dosyaları-veri tipleri, dönüş kodları, sabitler ve yapılar (devamı var)

Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
CMQCFXLA	MQCFIL64 -PCF 64 bit tamsayı listesi değiştirgesi				A
CMQCFXNA	MQCFIN64 -PCF 64 bit tamsayı değiştirgesi				A
CMQCHARVA	MQCHARV-Değişken uzunluklu dizgi				A
CMQCIHA	MQCIH- CICS bridge üstbilgisi				A
CMQCNOA	MQCNO-Bağlantı seçenekleri				A
CMQCSPA	MQCSP-Güvenlik değiştirgeleri				A
CMQCXPA	MQCXP-Kanal çıkış değiştirgeleri				A
CMQDHA	MQDH-Dağıtım üstbilgisi				A
CMQDLHA	MQDLH-Ölmeyen harf üstbilgisi				A
CMQDXPA	MQDXP-Veri dönüştürme çıkış parametreleri				A
CMQEPHA	MQEPH-Gömülü PCF üstbilgisi				A
CMQGMOA	MQGMO-İleti seçeneklerini al				A
CMQIIHA	MQIIH- IMS bilgi üstbilgisi				A
CMQMDA	MQMD-İleti tanımlayıcı				A
CMQMD1A	MQMD1 -İleti tanımlayıcı sürüm 1				A
CMQMD2A	MQMD2 -İleti tanımlayıcı sürüm 2				A
CMQMDEA	MQMDE-İleti tanımlayıcı genişletildi				A
CMQODA	MQOD-Nesne tanımlayıcı				A
CMQORA	MQOR-Nesne kaydı				A
CMQPMOA	MQPMO-İleti koyma seçenekleri				A
CMQRFHA	MQRFH-Kurallar ve biçimlendirme üstbilgisi				A
CMQRFH2A	MQRFH2 -Kurallar ve biçimlendirme üstbilgisi 2				A
CMQRMHA	MQRMH-Başvuru iletisi üstbilgisi				A
CMQTMMA	MQTM-Tetikleyici iletisi				A
CMQTMCA	MQTMCA -Tetikleyici iletisi 2 karakteri				A
CMQWCRA	MQWCR-Küme iş yükü küme kaydı				A
CMQWDRA	MQWDR-Küme iş yükü hedef kaydı				A
CMQWDR1A	MQWDR1 -Küme iş yükü hedef kayıt sürüm 1				A
CMQWDR2A	MQWDR2 -Küme iş yükü hedef kaydı sürüm 2				A

Çizelge 6. z/OS Çevirici kopyalama dosyaları-veri tipleri, dönüş kodları, sabitler ve yapılar (devamı var)

Dosya adı	Açıklama	IBM i	AIX and Linux	Windows	z/OS
CMQWIHA	MQWIH-İş bilgileri üstbilgisi				A
CMQWQRA	MQWQR-Küme iş yükü kuyruğu kaydı				A
CMQWQR1A	MQWQR1 -Küme iş yükü kuyruğu kayıt sürüm 1				A
CMQWQR2A	MQWQR2 -Küme iş yükü kuyruğu kaydı sürüm 2				A
CMQWXP	MQWXP-Küme iş yükü çıkış parametreleri				A
CMQWXP1A	MQWXP1 -Küme iş yükü çıkış parametreleri sürüm 1				A
CMQWXP2A	MQWXP2 -Küme iş yükü çıkış parametreleri sürüm 2				A
CMQWXP3A	MQWXP3 -Küme iş yükü çıkış parametreleri sürüm 3				A
CMQXPA	MQXP- CICS API-çapraz geçiş çıkış parametreleri				A
CMQXQHA	MQXQH-İletim kuyruğu üstbilgisi				A
CMQXWDA	MQXWD-Çıkış bekleme tanımlayıcısı				A

Anahtar: A= Dosya sağlandı

MQ_* (Dizgi Uzunlukları)

Çizelge 7. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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'

Çizelge 7. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_DOSYA_ADI_UZUNLUĞU	8	X'00000008'
MQ_CLIENT_ID_LENGTH	23	X'00000017'
MQ_CLUSTER_NAME_LENGTH	48	X'00000030'
MQ_CONN_NAME_LENGTH	264	X'00000108'
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_AYIRT edici ad_adi_uzunluğu	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_AD_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_AD_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_AD_UZUNLUK	8	X'00000008'

Çizelge 7. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'00000FFF'
MQ_MAX_USER_ID_LENGTH	64	X'00000040'
MQ_MCA_JOB_NAME_LENGTH	28	X'0000001C'
MQ_MCA_AD_UZUNLUK	20	X'00000014'
MQ_MCA_USER_DATA_LENGTH	32	X'00000020'
MQ_MCA_USER_ID_LENGTH	(value differs by platform or version)	(value differs by platform or version)
MQ_MFS_MAP_AD_UZUNLUK	8	X'00000008'
MQ_MODE_AD_UZUNLUK	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'
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_ADI_UZUNLUĞU	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_AD_UZUNLUK	48	X'00000030'
MQ_Q_AD_UZUNLUK	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'

Çizelge 7. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQ_SERVICE_ARGS_LENGTH	255	X'000000FF'
MQ_SERVICE_COMMAND_LENGTH	255	X'000000FF'
MQ_SERVICE_DESC_LENGTH	64	X'00000040'
MQ_HIZMET_ADI_UZUNLUĞU	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 (mq_ssl_anahtar_havuz_uzunluğu)	256	X'00000100'
MQ_SSL_PEER_NAME_LENGTH	1024	X'00000400'
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_YOK	0	X'00000000'
MQ_TCP_NAME_LENGTH	8	X'00000008'
MQ_ZAMAN_UZUNLUĞU	8	X'00000008'
MQ_TOPIC_DESC_LENGTH	64	X'00000040'
MQ_TOPIC_AD_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_ADI_UZUNLUK	8	X'00000008'
MQ_TRIGGER_TERM_ID_LENGTH	4	X'00000004'

Çizelge 7. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi Dizgi Uzunlukları)

Çizelge 8. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQ_ARCHIVE_PFX_LENGTH	36	X'00000024'
MQ_ARCHIVE_UNIT_LENGTH	8	X'00000008'
MQ_ASID_LENGTH	4	X'00000004'
MQ_AUTH_PROFIL_ADI_UZUNLUĞU	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'
MQ_DB2_NAME_LENGTH	4	X'00000004'
MQ_DSG_AD_UZUNLUK	8	X'00000008'
MQ_ENTITY_NAME_LENGTH	1024	X'0000400'
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'0000400'
MQ_LRSN_LENGTH	12	X'0000000C'
MQ_ORIGIN_NAME_LENGTH	8	X'00000008'
MQ_PSB_AD_UZUNLUK	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_SISTEM_ADI_UZUNLUĞU	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'

Çizelge 8. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQ_VOLSER_LENGTH	6	X'00000006'

MQACH_* (API çıkış zinciri alanı üstbilgi yapısı)

Çizelge 9. Sabitlerin yapıları	
Ad	Yapı
MQACH_STRUC_ID	"ACH↵"
MQACH_STRUC_ID_ARRAY	'A', 'C', 'H', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 10. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Muhasebe Simgesi)

Çizelge 11. Sabit adlar ve değerler	
Ad	Değer
MQACT_NONE	X'00...00' (32 boş değer)
MQACT_NONE_ARRAY	'\0', '\0', ... (32 boş değer)

MQACT_* (Komut biçimi İşlem Seçenekleri)

Çizelge 12. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQACT_FORCE_REMOVE	1	X'00000001'
MQACT_AVANSCE_LOG	2	X'00000002'
MQACT_COLLECT_STATISTICS	3	X'00000003'
MQACT_PUBSUB	4	X'00000004'

MQACTP_* (İşlem)

Çizelge 13. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQACTP_YENI	0	X'00000000'
MQACTP_FORWARD	1	X'00000001'
MQACTP_REPLY	2	X'00000002'
MQACTP_REPORT	3	X'00000003'

MQACTT_* (Muhasebe Simgesi Tipleri)

Çizelge 14. Değişmezlerin değerleri	
Ad	Onaltılı değer
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_* (Yeni MCA Denetimlerini Benimse ve Yeni MCA Tiplerini Benimse)

Yeni MCA Denetimlerini Benimse

Çizelge 15. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQADOPT_CHECK_NONE	0	X'00000000'
MQADOPT_CHECK_ALL	1	X'00000001'
MQADOPT_CHECK_Q_MGR_NAME	2	X'00000002'
MQADOPT_CHECK_NET_ADDR	4	X'00000004'

Yeni MCA Tiplerini Benimse

Çizelge 16. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Kimlik doğrulama bilgileri kayıt yapısı)

Çizelge 17. Sabitlerin yapıları	
Ad	Yapı
MQAIR_STRUC_ID	"AIR↵"
MQAIR_STRUC_ID_ARRAY	'A', 'I', 'R', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 18. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQAIR_VERSION_1	1	X'00000001'
MQAIR_VERSION_2	2	X'00000002'
MQAIR_CURRENT_VERSION	2	X'00000002'

MQAIT_ * (Kimlik Doğrulama Bilgi Tipi)

Çizelge 19. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQAIT_Tümü	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_ * (Komut biçimi Zamanuyumsuz Durum Değerleri)

Çizelge 20. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQAS_NONE	0	X'00000000'
MQAS_BAŞLATILDI	1	X'00000001'
MQAS_START_WAIT	2	X'00000002'
MQAS_DURDURULDU	3	X'00000003'
MQAS_ASKIYA ALINDI	4	X'00000004'
MQAS_SUSPENDED_TEMPORARY	5	X'00000005'
MQAS_ETKİNLEŞTİRİCİ	6	X'00000006'
MQAS_ETKİN DEĞİL	7	X'00000007'

MQAT_ * (Put Uygulama Tipleri)

Çizelge 21. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQAT_BILINMIYOR	-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'

Çizelge 21. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_KULLANICISI	25	X'00000019'
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_* (Komut biçimi Yetki Değerleri)

Çizelge 22. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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'

Çizelge 22. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_ABONE	20	X'00000014'
MQAUTH_RESUME	21	X'00000015'
MQAUTH_SYSTEM	22	X'00000016'

MQAUTHOPT_* (Komut biçimi Yetki Seçenekleri)

Çizelge 23. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQAUTHOPT_CUMULATIVE	256	X'00000100'
MQAUTHOPT_ENTITY_EXPLICIT	1	X'00000001'
MQAUTHOPT_ENTITY_SET	2	X'00000002'
MQAUTHOPT_NAME_ALL_EŞLEŞME	32	X'00000020'
MQAUTHOPT_NAME_AS_WILDCARD	64	X'00000040'
MQAUTHOPT_NAME_EXPLICIT	16	X'00000010'

MQAXC_* (API çıkış bağlamı yapısı)

Çizelge 24. Sabitlerin yapıları

Ad	Yapı
MQAXC_STRUC_ID	"AXC↵"
MQAXC_STRUC_ID_ARRAY	'A', 'X', 'C', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 25. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQAXC_VERSION_1	1	X'00000001'
MQAXC_CURRENT_VERSION	1	X'00000001'

MQAXP_* (API çıkış deęiřtirgesi yapısı)

Çizelge 26. Sabitlerin yapıları	
Ad	Yapı
MQAXP_STRUC_ID	"AXP~"
MQAXP_STRUC_ID_ARRAY	'A', 'X', 'P', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 27. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQAXP_VERSION_1	1	X'00000001'
MQAXP_VERSION_2	2	X'00000002'
MQAXP_CURRENT_VERSION	2	X'00000002'

MQBA_* (Byte Öznitelik Seçicileri)

Çizelge 28. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQBA_FIRST	6001	X'00001771'
MQBA_LAST	8000	X'00001F40'

MQBACF_* (Komut biçimi Byte Parametre Tipleri)

Çizelge 29. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
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_KULLANILAN	7016	X'00001B68'

MQBL_* (mqAddDizesi ve mqSetDizesi için Arabellek Uzunluğu)

Çizelge 30. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQBL_NULL_SONLANDIRILDI	-1	X'FFFFFFFF'

MQBMHO_* (İleti işleme seçenekleri ve yapısı için arabellek)

İleti işleme seçenekleri yapısı için arabellek

Çizelge 31. Sabitlerin yapıları	
Ad	Yapı
MQBMHO_STRUC_ID	"BMHO"
MQBMHO_STRUC_ID_ARRAY	'B','M','H','O'

Not: – simgesi, tek bir boş karakteri temsil eder.

Çizelge 32. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQBMHO_VERSION_1	1	X'00000001'
MQBMHO_CURRENT_VERSION	1	X'00000001'

İleti İşleyecek Arabellek Seçenekleri

Çizelge 33. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQBMHO_NONE	0	X'00000000'
MQBMHO_DELETE_PROPERTIES	1	X'00000001'

MQBND_* (Varsayılan Bağ Tanımları)

Çizelge 34. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQBND_BIND_ON_OPEN	0	X'00000000'
MQBND_BIND_NOT_FIXED	1	X'00000001'
MQBND_BIND_ON_GROUP	2	X'00000002'

MQBO_* (Başlangıç seçenekleri ve yapısı)

Başlangıç seçenekleri yapısı

Çizelge 35. Sabitlerin yapıları	
Ad	Yapı
MQBO_STRUC_ID	"BO--"
MQBO_STRUC_ID_ARRAY	'B','O','-','-'

Not: – simgesi, tek bir boş karakteri temsil eder.

<i>Çizelge 36. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQBO_VERSION_1	1	X'00000001'
MQBO_CURRENT_VERSION	1	X'00000001'

Başlangıç Seçenekleri

<i>Çizelge 37. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQBO_NONE	0	X'00000000'

MQBT_* (Komut biçimi Köprü Tipleri)

<i>Çizelge 38. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQBT_OTMA	1	X'00000001'

MQCA_* (Karakter Özniteliği Seçicileri)

<i>Çizelge 39. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQCA_ADMIN_TOPIC_NAME	2105	X'00000839'
MQCA_ALTERATION_DATE	2027	X'000007EB'
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'

Çizelge 39. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 39. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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_* (Komut biçimi Karakter Değişirgesi Tipleri)

Çizelge 40. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'
MQCACF_ESCAPE_METIN	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_BAŞVURU_ADI	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_KONU	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'

Çizelge 40. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_NESNE_ADI	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'
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_VERI_KÜMESI_ADI	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_PROFILI_ADI	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'

Çizelge 40. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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_SISTEM_ADI	3100	X'00000C1C'
MQCACF_STRUC_BACKUP_START	3101	X'00000C1D'
MQCACF_CF_STRUC_BACKUP_END	3102	X'00000C1E'
MQCACF_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'

Çizelge 40. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_ADI	3125	X'00000C35'
MQCACF_FROM_SERVICE_NAME	3126	X'00000C36'
MQCACF_TO_HIZMET_ADI	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'
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'

Çizelge 40. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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_ANASISTEM_ADI	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'

Çizelge 40. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_KULLANILAN	3211	X'00000C8B'

MQCACH_* (Komut biçimi Karakter Kanal Değiştirgesi Tipleri)

Çizelge 41. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQCACH_FIRST	3501	X'00000DAD'
MQCACH_CHANNEL_NAME	3501	X'00000DAD'
MQCACH_DESC	3502	X'00000DAE'
MQCACH_MODE_NAME	3503	X'00000DAF'
MQCACH_TP_NAME	3504	X'00000DB0'
MQCACH_XMIT_Q_NAME	3505	X'00000DB1'
MQCACH_CONNECTION_NAME	3506	X'00000DB2'
MQCACH_MCA_NAME	3507	X'00000DB3'
MQCACH_SEC_EXIT_NAME	3508	X'00000DB4'
MQCACH_MSG_EXIT_NAME	3509	X'00000DB5'
MQCACH_SEND_EXIT_NAME	3510	X'00000DB6'
MQCACH_RCV_EXIT_NAME	3511	X'00000DB7'
MQCACH_CHANNEL_ADLARI	3512	X'00000DB8'
MQCACH_SEC_EXIT_USER_DATA	3513	X'00000DB9'
MQCACH_MSG_EXIT_USER_DATA	3514	X'00000DBA'
MQCACH_SEND_EXIT_USER_DATA	3515	X'00000DBB'
MQCACH_RCV_EXIT_USER_DATA	3516	X'00000DBC'
MQCACH_USER_ID	3517	X'00000DBD'
MQCACH_PASSWORD	3518	X'00000DBE'
MQCACH_LOCAL_ADDRESS	3520	X'00000DC0'
MQCACH_LOCAL_NAME	3521	X'00000DC1'
MQCACH_LAST_MSG_TIME	3524	X'00000DC4'
MQCACH_LAST_MSG_DATE	3525	X'00000DC5'
MQCACH_MCA_USER_ID	3527	X'00000DC7'

Çizelge 41. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQCACH_CHANNEL_START_TIME	3528	X'00000DC8'
MQCACH_CHANNEL_START_DATE	3529	X'00000DC9'
MQCACH_MCA_JOB_NAME	3530	X'00000DCA'
MQCACH_LAST_LUWID	3531	X'00000DCB'
MQCACH_CURRENT_LUWID	3532	X'00000DCC'
MQCACH_FORMAT_ADı	3533	X'00000DCD'
MQCACH_MR_EXIT_NAME	3534	X'00000DCE'
MQCACH_MR_EXIT_USER_DATA	3535	X'00000DCF'
MQCACH_SSL_CIPHER_SPEC	3544	X'00000DD8'
MQCACH_SSL_PEER_NAME	3545	X'00000DD9'
MQCACH_SSL_HANDSHAKE_STAGE	3546	X'00000DDA'
MQCACH_SSL_SHORT_PEER_NAME	3547	X'00000ddb'
MQCACH_REMOTE_APPL_TAG	3548	X'00000DDC'
MQCACH_SSL_CERT_USER_ID	3549	X'00000DDD'
MQCACH_SSL_CERT_ISSUER_NAME	3550	X'00000DDE'
MQCACH_LU_ADı	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_KULLANILAN	3559	X'00000DE7'

MQCADSD_* (CICS bilgi üstbilgisi ADS Tanımlayıcıları)

Çizelge 42. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCADSD_NONE	0	X'00000000'
MQCADSD_SEND	1	X'00000001'
MQCADSD_RECV	16	X'00000010'
MQCADSD_MSGFORMAT	256	X'00000100'

MQCAFTY_* (Bağlantı Yakınlığı Değerleri)

Çizelge 43. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCAFTY_NONE	0	X'00000000'
MQCAFTY_PREFERRED	1	X'00000001'

MQCAMO_* (Komut biçimi Karakter İzleme Değişirgesi Tipleri)

Çizelge 44. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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'
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_KULLANILAN	2712	X'00000A98'

MQCBC_* (MQCBC sabitleri yapısı)

Çizelge 45. Sabitlerin yapıları

Ad	Yapı
MQCBC_STRUC_ID	"CBC~"
MQCBC_STRUC_ID_ARRAY	'C', 'B', 'C', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 46. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQCBC_VERSION_1	1	X'00000001'
MQCBC_CURRENT_VERSION	1	X'00000001'

MQCBCF_* (MQCBC değişmezleri işaretleri)

Çizelge 47. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQCBCF_NONE	0	X'00000000'
MQCBCF_READA_BUFFER_EMPTY	1	X'00000001'

MQCBCT_* (MQCBC değişmezleri geri çağırma tipi)

Çizelge 48. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQCBCT_START_CALL	1	X'00000001'
MQCBCT_STOP_CALL	2	X'00000002'
MQCBCT_REGISTER_CALL	3	X'00000003'

Çizelge 48. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQCBCT_DEREGISTER_CALL	4	X'00000004'
MQCBCT_EVENT_CALL	5	X'00000005'
MQCBCT_MSG_REMOVED	6	X'00000006'
MQCBCT_MSG_NOT_REMOVED	7	X'00000007'

MQCBBD_* (MQCBBD değişmezleri yapısı)

Çizelge 49. Sabitlerin yapıları	
Ad	Yapı
MQCBBD_STRUC_ID	"CBD↵"
MQCBBD_STRUC_ID_ARRAY	'C','B','D','↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 50. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCBBD_VERSION_1	1	X'00000001'
MQCBBD_CURRENT_VERSION	1	X'00000001'

MQCBDO_* (MQCBDO değişmezleri geri çağırma seçenekleri)

Çizelge 51. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (mqCreateBag için Create-Bag Seçenekleri)

Çizelge 52. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_INENGELLEME	0	X'00000000'
MQCBO_REORDER_AS_REQUIRED	4	X'00000004'
MQCBO_DO_NOT_REORDER	0	X'00000000'

Çizelge 52. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQCBO_CHECK_SELECTORS	8	X'00000008'
MQCBO_DO_NOT_CHECK_SELECTORS	0	X'00000000'

MQCBT_* (MQCBD değişmezleri Bu, Geri Çağırma İşlevinin tipidir)

Çizelge 53. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCBT_MESSAGE_CONSUMER	1	X'00000001'
MQCBT_EVENT_HANDLER	2	X'00000002'

MQCC_* (tamamlanma kodları)

Çizelge 54. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCC_OK	0	X'00000000'
MQCC_UYARISI	1	X'00000001'
MQCC_FAILED	2	X'00000002'
MQCC_UNKNOWN	-1	X'FFFFFFFF'

MQCCSI_* (Kodlanmış Karakter Kümesi Tanıtıcıları)

Çizelge 55. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCCSI_AYRILMAMIŞ	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 bilgi üstbilgisi Conversational Task Options)

Çizelge 56. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCCT_YES	1	X'00000001'
MQCCT_NO	0	X'00000000'

MQCD_* (Kanal tanımlaması yapısı)

Çizelge 57. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCD_VERSION_1	1	X'00000001'
MQCD_VERSION_2	2	X'00000002'
MQCD_VERSION_3	3	X'00000003'
MQCD_VERSION_4	4	X'00000004'

Çizelge 57. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
 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_* (Kanal Verileri Dönüştürme)

Çizelge 58. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQCDC_SENDER_CONVERSION	1	X'00000001'
MQCDC_NO_SENDER_CONVERSION	0	X'00000000'

MQCERT_* (Sertifika Doğrulama İlkesi Tipi)

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_* (Yetenek İşaretleri)

Çizelge 59. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCF_NONE	0	X'00000000'
MQCF_DIST_LIST	1	X'00000001'

MQCFAC_* (CICS bilgi üstbilgisi Olanığı)

Çizelge 60. Sabit adlar ve değerler	
Ad	Onaltılı değer
MQCFAC_NONE	X'00...00' (8 boş değer)
MQCFAC_NONE_ARRAY	'\0', '\0', ... (8 boş değer)

MQCFBF_* (Komut biçimi byte dizgisi süzgeci değiştirgesi yapısı)

Çizelge 61. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFBF_STRUC_LENGTH_FIXED	20	X'00000014'

MQCFBS_* (Komut biçimi byte dizgisi değiştirgesi yapısı)

Çizelge 62. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFBS_STRUC_LENGTH_FIXED	16	X'00000010'

MQCF_* (Komut biçimi üstbilgi Denetim Seçenekleri)

Çizelge 63. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCF_LAST	1	X'00000001'
MQCF_NOT_LAST	0	X'00000000'

MQCFGR_* (Komut biçimi grup parametresi yapısı)

Çizelge 64. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFGR_STRUC_LENGTH	16	X'00000010'

MQCFH_* (Komut biçimi üstbilgi yapısı)

Çizelge 65. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi tamsayı süzgeci değiştirgesi yapısı)

Çizelge 66. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQC50_STRUC_LENGTH	20	X'00000014'

MQCFIL_* (Komut biçimi tamsayı listesi değiştirgesi yapısı)

Çizelge 67. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFIL_STRUC_LENGTH_FIXED	16	X'00000010'

MQCFIL64_* (Komut biçimi 64 bit tamsayı listesi değiştirge yapısı)

Çizelge 68. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFIL64_STRUC_LENGTH_FIXED	16	X'00000010'

MQCFIN_* (Komut biçimi tamsayı değiştirgesi yapısı)

Çizelge 69. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFIN_STRUC_LENGTH	16	X'00000010'

MQCFIN64_* (Komut biçimi 64 bitlik tamsayı değiştirgesi yapısı)

Çizelge 70. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFIN64_STRUC_LENGTH	24	X'00000018'

MQCFO_* (Komut biçimi Havuz Yenileme Seçenekleri ve Komut biçimi Kuyrukları Kaldırma Seçenekleri)

Komut biçimi Havuzu Yenile Seçenekleri

Çizelge 71. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFO_REFRESH_HAVUZ_EVET	1	X'00000001'
MQCFO_REFRESH_HAVUZ_NO	0	X'00000000'

Komut biçimi Kuyrukları Kaldırma Seçenekleri

Çizelge 72. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFO_REMOVE_QUEUES_YES	1	X'00000001'
MQCFO_REMOVE_QUEUES_NO	0	X'00000000'

MQCFOP_* (Komut biçimi Süzgeç İşleçleri)

Çizelge 73. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFOP_LESS	1	X'00000001'
MQCFOP_EQUAL	2	X'00000002'
MQCFOP_GREATER	4	X'00000004'
MQCFOP_NOT_LESS	6	X'00000006'
MQCFOP_NOT_EQUAL	5	X'00000005'
MQCFOP_NOT_GREATER	3	X'00000003'
MQCFOP_LIKE	18	X'00000012'
MQCFOP_NOT_LIKE	21	X'00000015'
MQCFOP_CONTAINS	10	X'0000000A'
MQCFOP_IÇERMEZ	13	X'0000000D'
MQCFOP_CONTAINS_GEN	26	X'0000001A'
MQCFOP_EXCLUDES_GEN	29	X'0000001D'

MQCFR_* (CF Kurtarıbilirliği)

Çizelge 74. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFR_YES	1	X'00000001'
MQCFR_NO	0	X'00000000'

MQCFSF_* (Komut biçimi dizgi süzgeci değiştirgesi yapısı)

Çizelge 75. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFSF_STRUC_LENGTH_FIXED	24	X'00000018'

MQCFSL_* (Komut biçimi dizgi listesi değiştirgesi yapısı)

Çizelge 76. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFSL_STRUC_LENGTH_FIXED	24	X'00000018'

MQCFST_* (Komut biçimi dizgi değiştirgesi yapısı)

Çizelge 77. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFST_STRUC_LENGTH_FIXED	20	X'00000014'

MQCFSTATUS_* (Komut biçimi CF Durumu)

Çizelge 78. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFSTATUS_NOT_FOUND	0	X'00000000'

Çizelge 78. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQCFSTATUS_ETKINLEŞTİRICI	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_TAMAMLANMAMIŞ	20	X'00000014'
MQCFSTATUS_NEVER_KULLANILDI	21	X'00000015'
MQCFSTATUS_NO_BACKUP	22	X'00000016'
MQCFSTATUS_NOT_FAILED	23	X'00000017'
MQCFSTATUS_NOT_RECOVERABLE	24	X'00000018'
MQCFSTATUS_XES_HATA	25	X'00000019'

MQCFT_* (Komut biçimi Yapı Tipleri)

Çizelge 79. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQCFT_NONE	0	X'00000000'
MQCFT_COMMAND	1	X'00000001'
MQCFT_YANITI	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_STATISTIK	21	X'00000015'
MQCFT_MUHASEBE	22	X'00000016'
MQCFT_INTEGER64	23	X'00000017'

Çizelge 79. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQCFT_INTEGER64_LIST	25	X'00000019'

MQCFTYPE_* (Komut biçimi CF Tipleri)

Çizelge 80. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCFTYPE_APPL	0	X'00000000'
MQCFTYPE_ADMIN	1	X'00000001'

MQCFUNC_* (CICS bilgi üstbilgisi İşlevleri)

Çizelge 81. Sabitlerin yapıları	
Ad	Yapı
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	'~','~','~','~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

MQCGWI_* (CICS bilgi üstbilgisi Bekleme Aralığı AI)

Çizelge 82. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCGWI_DEFAULT	-2	X'FFFFFFE'

MQCHAD_* (Kanal Otomatik Tanımlaması)

Çizelge 83. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCHAD_DISABLED	0	X'00000000'
MQCHAD_ENABLED	1	X'00000001'

MQCHIDS_* (Komut biçimi Indoubt Status)

Çizelge 84. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCHIDS_NOT_INDOUBT	0	X'00000000'
MQCHINDOUBT	1	X'00000001'

MQCHLD_* (Komut biçimi Kanal Düzenleme)

Çizelge 85. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi Kanal Durumu)

Çizelge 86. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi Kanal Paylaşılan Yeniden Başlatma Seçenekleri)

Çizelge 87. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCHSH_RESTART_NO	0	X'00000000'
MQCHSH_RESTART_YES	1	X'00000001'

MQCHSR_* (Komut biçimi Kanal Durdurma Seçenekleri)

Çizelge 88. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCHSR_STOP_STOP_NOT_ISTENEN	0	X'00000000'
MQCHSR_STOP_ISTENEN	1	X'00000001'

MQCHSSTATE_* (Komut biçimi Kanal Alt Durumu)

Çizelge 89. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCHSSTATE_DİĞER	0	X'00000000'
MQCHSSTATE_END_OF_BATCH	100	X'00000064'
MQCHSSTATE_XX_ENCODE_CASE_CAPS_LOCK_OFF GÖNDERİYOR	200	X'000000C8'
MQCHSSTATE_ALICI	300	X'0000012C'
MQCHSSTATE_SERIALIZING	400	X'00000190'
MQCHSSTATE_RESYNCHING	500	X'000001F4'
MQCHSSTATE_HEARTSAĞLIKLİ işletim bildirimi	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_BAĞLANIYOR	1250	X'000004E2'
MQCHSSTATE_SSL_HANDSHAKING	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_SIKIŞTIRMA	1800	X'00000708'

MQCHT_* (Kanal Tipleri)

Çizelge 90. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCHT_SENDER	1	X'00000001'
MQCHT_SERVER	2	X'00000002'
MQCHT_ALICI	3	X'00000003'
MQCHT_REQUESTER	4	X'00000004'
MQCHT_Tümü	5	X'00000005'
MQCHT_CLNTCONN	6	X'00000006'
MQCHT_SVRCONN	7	X'00000007'
MQCHT_CLUSRCVR	8	X'00000008'
MQCHT_CLUSSDR	9	X'00000009'

MQCHTAB_* (Komut biçimi Kanal Çizelgesi Tipleri)

Çizelge 91. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCHTAB_Q_MGR	1	X'00000001'

Çizelge 91. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQCHTAB_CLNTCONN	2	X'00000002'

MQCI_* (İlinti Tanıtıcısı)

Çizelge 92. Sabit adlar ve değerler	
Ad	Değer
MQCI_NONE	X'00...00' (24 boş değer)
MQCI_NONE_ARRAY	'\0', '\0', ... (24 boş değer)
MQCI_YENI_OTURUM	X'414D5121...'
MQCI_NEW_SESSION_ARRAY	'\x41', '\x4D', '\51', '\x21', ...

MQCIH_* (CICS bilgi üstbilgisi yapısı ve işaretleri)

CICS bilgi üstbilgisi yapısı

Çizelge 93. Sabitlerin yapıları	
Ad	Yapı
MQCIH_STRUC_ID	"CIH~"
MQCIH_STRUC_ID_ARRAY	'C', 'I', 'H', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 94. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

CICS bilgi üstbilgisi İşaretleri

Çizelge 95. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCIH_YOK	0	X'00000000'
MQCIH_PASS_SÜRE SONU	1	X'00000001'
MQCIH_UNLIMITED_SON kullanma tarihi	0	X'00000000'
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_* (Küme Önbellek Tipleri)

Çizelge 96. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCLCT_STATIC	0	X'00000000'
MQCLCT_DYNAMIC	1	X'00000001'

MQCLRS_* (Komut biçimi Konu Dizesi Kapsamını Temizle)

Çizelge 97. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCLRS_LOCAL	1	X'00000001'
MQCLRS_GENEL	2	X'00000002'

MQCLRT_* (Komut biçimi Konu Dizesi Tipini Temizle)

Çizelge 98. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCLRT_KORUNDU	1	X'00000001'

MQCLT_* (CICS bilgi üstbilgisi Bağlantı Tipleri)

Çizelge 99. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCLT_PROGRAM	1	X'00000001'
MQCLT_TRANSACTION	2	X'00000002'

MQCLWL_* (Küme İş Yüğü)

Çizelge 100. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCLWL_USEQ_LOCAL	0	X'00000000'
MQCLWL_USEQ_ANY	1	X'00000001'
MQCLWL_USEQ_AS_Q_MGR	-3	X'FFFFFFFD'

MQCLXQ_* (Küme iletim kuyruğu tipi)

MQCLXQ_* , DEFCLXQ kuyruk yöneticisi özniteliğinde ayarlayabileceğiniz değerlerdir. **DEFCLXQ** özniteliği, iletileri küme alıcı kanallarına göndermek, iletileri almak için küme gönderen kanalları tarafından varsayılan olarak hangi iletim kuyruğunun seçileceğini denetler.

Çizelge 101. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCLXQ_SCTQ	0	X'00000000'
MQCLXQ_CHANNEL	1	X'00000001'

İlgili başvurular

“DefClusterXmitQueueType (MQLONG)” sayfa 807

DefClusterXmitQueueType özniteliği, iletileri küme alıcı kanallarına göndermek için küme gönderen kanalları tarafından varsayılan olarak hangi iletim kuyruğunun seçileceğini denetler.

Kuyruk Yöneticisini Değiştir

Sorma Kuyruğu Yöneticisi

Sorma Kuyruğu Yöneticisi (Yanıt)

“MQINQ-Nesne özniteliklerini sorgu” sayfa 696

MQINQ çağırısı, bir nesnenin özniteliklerini içeren bir dizi tamsayı ve karakter dizgisi döndürür.

MQCMD_* (Komut Kodları)

<i>Çizelge 102. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 102. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_SUBSCRIBER	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'
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'

Çizelge 102. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_STRUC	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_STRUC	105	X'00000069'
MQCMD_CREATE_BUFFER_POOL	106	X'0000006A'
MQCMD_CREATE_PAGE_SET	107	X'0000006B'
MQCMD_CREATE_CF_STRUC	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_STRUC	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'
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'

Çizelge 102. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQCMD_RECOVER_CF_STRUC	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'
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'

Çizelge 102. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_ABONELIK	176	X'000000B0'
MQCMD_CREATE_SUBSCRIPTION	177	X'000000B1'
MQCMD_CHANGE_ABONELIK	178	X'000000B2'
MQCMD_DELETE_SUBSCRIPTION	179	X'000000B3'
MQCMD_COPY_ABONELIK	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_ * (Komut biçimi Komut Bilgileri Değerleri)

Çizelge 103. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_BASLATILDI	11	X'0000000B'
MQCMDI_BACKUP_BASLATILDI	12	X'0000000C'
MQCMDI_RECOVER_COMPLETED	13	X'0000000D'
MQCMDI_SEC_TIMER_ZERO	14	X'0000000E'
MQCMDI_REFRESH_CONFIGURATION	16	X'00000010'
MQCMDI_SEC_SIGNOFF_HATA	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_* (Komut Düzeyleri)

Çizelge 104. Sabit adlar ve değerler	
Ad	Değer
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_* (İleti tanıtıcısı seçenekleri ve yapısı yarat)

İleti tanıtıcısı seçenekleri yapısı yarat

Çizelge 105. Sabitlerin yapıları	
Ad	Yapı
MQCMHO_STRUC_ID	"CMHO"
MQCMHO_STRUC_ID_ARRAY	'C', 'M', 'H', 'O'

Not: ↪ simgesi, tek bir boş karakteri temsil eder.

Çizelge 106. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCMHO_VERSION_1	1	X'00000001'
MQCMHO_CURRENT_VERSION	1	X'00000001'

İleti Yaratma İşleyecek Seçenekleri

Çizelge 107. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCMHO_DEFAULT_VALIDATION	0	X'00000000'
MQCMHO_NO_VALIDATION	1	X'00000001'
MQCMHO_VALIDATE	2	X'00000002'
MQCMHO_NONE	0	X'00000000'

MQCNO_* (Bağlantı seçenekleri ve yapısı)

Bağlantı seçenekleri yapısı

Çizelge 108. Sabitlerin yapıları	
Ad	Yapı
MQCNO_STRUC_ID	"CNO↔"
MQCNO_STRUC_ID_ARRAY	'C','N','O','↔'

Not: ↔ simgesi, tek bir boş karakteri temsil eder.

Çizelge 109. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

Bağlantı Seçenekleri

Çizelge 110. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

Çizelge 110. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
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_PAYLAŞIMI	65536	X'00010000'
MQCNO_ALL_CONVS_SHARE	262144	X'00040000'
YALNIZCA 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_* (Kapatma Seçenekleri)

Çizelge 111. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'
MQCO_QUIESCE	32	X'00000020'

MQCODL_* (CICS bilgi üstbilgisi Çıkış Verileri Uzunluğu)

Çizelge 112. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCODL_AS_INPUT	-1	X'FFFFFFFF'

MQCOMPRESS_* (Kanal Sıkıştırma)

Çizelge 113. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCOMPRESS_NOT_VAR	-1	X'FFFFFFFF'
MQCOMPRESS_NONE	0	X'00000000'
MQCOMPRESS_RLE	1	X'00000001'
MQCOMPRESS_ZLIBFAST	2	X'00000002'
MQCOMPRESS_ZLIBHIGH	4	X'00000004'

Çizelge 113. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQCOMPRESS_SYSTEM	8	X'00000008'
MQCOMPRESS_ANY	268435455	X'0FFFFFFF'

MQCONNID_* (Bağlantı Tanıtıcısı)

Çizelge 114. Sabit adlar ve değerler

Ad	Değer
MQCONNID_NONE	X'00...00' (24 boş değer)
MQCONNID_NONE_ARRAY	'\0', '\0', ... (24 boş değer)

MQCOPY_* (Özellik Kopyalama Seçenekleri)

Çizelge 115. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_* (Küme Kuyruğu Tipleri)

Çizelge 116. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQCQT_LOCAL_Q	1	X'00000001'
MQCQT_ALIAS_Q	2	X'00000002'
MQCQT_REMOTE_Q	3	X'00000003'
MQCQT_Q_MGR_ALIAS	4	X'00000004'

MQCRC_* (CICS bilgi üstbilgisi Dönüş Kodları)

Çizelge 117. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQCRC_OK	0	X'00000000'
MQCRC_CICS_EXEC_ERROR (MQCRC_CICS_HATASI)	1	X'00000001'
MQCRC_MQ_API_ERROR	2	X'00000002'
MQCRC_BRIDGE_HATA	3	X'00000003'
MQCRC_BRIDGE_ABEND	4	X'00000004'
MQCRC_APPLICATION_ABEND	5	X'00000005'
MQCRC_SECURITY_HATA	6	X'00000006'
MQCRC_PROGRAM_YOK	7	X'00000007'
MQCRC_BRIDGE_TIMEOUT	8	X'00000008'

<i>Çizelge 117. Değişmezlerin değerleri (devamı var)</i>		
Ad	Ondalık değer	Onaltılı değer
MQCRC_TRANSID_NOT_KULLANILAMIYOR	9	X'00000009'

MQCS_* (MQCBC değişmezleri Tüketici durumu)

<i>Çizelge 118. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQCS_NONE	0	X'00000000'
MQCS_SUSPENDED_TEMPORARY	1	X'00000001'
MQCS_SUSPENDED_USER_ACTION	2	X'00000002'
MQCS_ASKIYA_ALINDI	3	X'00000003'
MQCS_DURDURULDU	4	X'00000004'

MQCSC_* (CICS bilgi üstbilgisi Başlatma Kodları)

<i>Çizelge 119. Sabitlerin yapıları</i>	
Ad	Yapı
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	'-',',','-',',','-',',','-',',','-'

Not: - simgesi, tek bir boş karakteri temsil eder.

MQCSP_* (Bağlantı güvenlik değiştirgeleleri yapısı ve kimlik doğrulama tipleri)

Bağlantı güvenliği parametreleri yapısı

<i>Çizelge 120. Sabitlerin yapıları</i>	
Ad	Yapı
MQCSP_STRUC_ID	"CSP-
MQCSP_STRUC_ID_ARRAY	'C','S','P','-',',','-',',','-',',','-'

Not: - simgesi, tek bir boş karakteri temsil eder.

<i>Çizelge 121. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQCSP_VERSION_1	1	X'00000001'
MQCSP_VERSION_2	2	X'00000002'
MQCSP_VERSION_3	3	X'00000003'
MQCSP_CURRENT_VERSION	3	X'00000003'

Bağlantı güvenliği parametreleri Kimlik Doğrulama Tipleri

Çizelge 122. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCSP_AUTH_NONE	0	X'00000000'
MQCSP_AUTH_USER_ID_AND_PWD	1	X'00000001'
MQCSP_AUTH_ID_TOKEN	2	X'00000002'

MQCSRV_* (Komut Sunucusu Seçenekleri)

Çizelge 123. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Kuyruk Yöneticisi Bağlantı Etiketleri)

Çizelge 124. Sabit adlar ve değerler	
Ad	Değer
MQCT_NONE	X'00...00' (128 boş değer)
MQCT_NONE_ARRAY	'\0', '\0', ... (128 boş değer)

MQCTES_* (CICS bilgi üstbilgisi Görev Bitiş Durumu)

Çizelge 125. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCTES_NOSYNC	0	X'00000000'
MQCTES_COMMIT	256	X'00000100'
MQCTES_BACKOUT	4352	X'00001100'
MQCTES_ENDTASK	65536	X'00010000'

MQCTLO_* (MQCTL seçenekleri yapısı ve Tüketici Denetim Seçenekleri)

MQCTL seçenekleri yapısı

Çizelge 126. Sabitlerin yapıları	
Ad	Yapı
MQCTLO_STRUC_ID	"CTLO"
MQCTLO_STRUC_ID_ARRAY	'C', 'T', 'L', 'O'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 127. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCTLO_VERSION_1	1	X'00000001'
MQCTLO_CURRENT_VERSION	1	X'00000001'

MQCTL seçenekleri Tüketici Denetimi Seçenekleri

Çizelge 128. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQCTLO_NONE	0	X'00000000'
MQCTLO_THREAD_AFFINITY	1	X'00000001'
MQCTLO_FAIL_IF QUIESCING	8192	X'00002000'

MQCUOWC_* (CICS bilgi üstbilgisi İş Birimi Denetimleri)

Çizelge 129. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
YALNIZCA MQCUOWC_ONLY	273	X'00000111'
MQCUOWC_CONTINUE	65536	X'00010000'
MQCUOWC_FIRST	17	X'00000011'
MQCUOWC_MIDDLE	16	X'00000010'
MQCUOWC_SON	272	X'00000110'
MQCUOWC_COMMIT	256	X'00000100'
MQCUOWC_BACKOUT	4352	X'00001100'

MQCXP_* (Kanal çıkış değiştirgesi yapısı)

Çizelge 130. Sabitlerin yapıları	
Ad	Yapı
MQCXP_STRUC_ID	"CXP↯"
MQCXP_STRUC_ID_ARRAY	'C', 'X', 'P', '↯'

Not: ↯ simgesi, tek bir boş karakteri temsil eder.

Çizelge 131. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Hedef Sınıf)

Çizelge 132. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDC_YÖNETİLDİR	1	X'00000001'

Çizelge 132. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQDC_, SAĞLANAN	2	X'00000002'

MQDCC_* (Dönüştürme Seçenekleri ve Maskeler ve Katsayılar)

Dönüştürme Seçenekleri

Çizelge 133. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_REVERSED	32	X'00000020'
MQDCC_SOURCE_ENC_UNDEFINED	0	X'00000000'
MQDCC_TARGETC_NATIVE	(value differs by platform or version)	(value differs by platform or version)
MQDCC_TARGET_ENC_NORMAL	256	X'00000100'
MQDCC_TARGET_ENC_REVERSED	512	X'00000200'
MQDCC_TARGET_ENC_UNDEFINED	0	X'00000000'
MQDCC_NONE	0	X'00000000'

Dönüştürme Seçenekleri Maskeleri ve Katsayısı

Çizelge 134. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_* (Yayınlama/Abone Olma Silme Seçenekleri)

Çizelge 135. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQDELO_NONE	0	X'00000000'
MQDELO_LOCAL	4	X'00000004'

MQDH_* (Dağıtım üstbilgisi yapısı)

Çizelge 136. Sabitlerin yapıları

Ad	Yapı
MQDH_STRUC_ID	"DH--"

Çizelge 136. Sabitlerin yapıları (devamı var)	
Ad	Yapı
MQDH_STRUC_ID_ARRAY	'D', 'H', '-', '-'

Not: - simgesi, tek bir boş karakteri temsil eder.

Çizelge 137. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDH_VERSION_1	1	X'00000001'
MQDH_CURRENT_VERSION	1	X'00000001'

MQDHF_* (Dağıtım üstbilgisi işaretleri)

Çizelge 138. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDHF_NEW_MSG_IDS	1	X'00000001'
MQDHF_NONE	0	X'00000000'

MQDISCONNECT_* (Komut biçimi Bağlantı Kesme Tipleri)

Çizelge 139. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDISCONNECT_NORMAL	0	X'00000000'
MQDISCONNECT_IMPLICIT	1	X'00000001'
MQDISCONNECT_Q_MGR	2	X'00000002'

MQDL_* (Dağıtım Listeleri)

Çizelge 140. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDL_SUPPORTED	1	X'00000001'
MQDL_NOT_DESTEKLENMIYOR	0	X'00000000'

MQDLH_* (Onlu harfli üstbilgi yapısı)

Çizelge 141. Sabitlerin yapıları	
Ad	Yapı
MQDLH_STRUC_ID	"DLH-
MQDLH_STRUC_ID_ARRAY	'D', 'L', 'H', '-'

Not: - simgesi, tek bir boş karakteri temsil eder.

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

MQDLV_* (Kalıcı/Kalıcı Olmayan İleti Teslimi)

Çizelge 142. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDLV_AS_üst ÖğESI	0	X'00000000'
MQDLV_Tümü	1	X'00000001'
MQDLV_ALL_DUR	2	X'00000002'
MQDLV_ALL_AVAIL	3	X'00000003'

MQDMHO_* (İleti tanıtıcısı seçeneklerini ve yapısını sil)

İleti tanıtıcısı seçenekleri yapısını sil

Çizelge 143. Sabitlerin yapıları	
Ad	Yapı
MQDMHO_STRUC_ID	"DMHO"
MQDMHO_STRUC_ID_ARRAY	'D', 'M', 'H', 'O'

Not: – simgesi, tek bir boş karakteri temsil eder.

Çizelge 144. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDMHO_VERSION_1	1	X'00000001'
MQDMHO_CURRENT_VERSION	1	X'00000001'

İleti Silme Tanıtıcısı Seçenekleri

Çizelge 145. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDMHO_NONE	0	X'00000000'

MQDMPO_* (İleti özelliği seçeneklerini ve yapısını sil)

İleti özelliği seçenekleri yapısını sil

Çizelge 146. Sabitlerin yapıları	
Ad	Yapı
MQDMPO_STRUC_ID	"DMPO"
MQDMPO_STRUC_ID_ARRAY	'D', 'M', 'P', 'O'

Not: – simgesi, tek bir boş karakteri temsil eder.

Çizelge 147. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDMPO_VERSION_1	1	X'00000001'
MQDMPO_CURRENT_VERSION	1	X'00000001'

İleti Silme Özelliği Seçenekleri

Çizelge 148. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDMPO_DEL_FIRST	0	X'00000000'
MQDMPO_DEL_PROP_UNDER_CURSOR	1	X'00000001'
MQDMPO_NONE	0	X'00000000'

MQDNSWLM_* (DNS WLM)

Çizelge 149. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDNSWLM_NO	0	X'00000000'
MQDNSWLM_YES	1	X'00000001'

MQDT_* (Hedef Tipleri)

Çizelge 150. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDT_APPL	1	X'00000001'
MQDT_BROKER	2	X'00000002'

MQDXP_* (Dönüştürme çıkış değiştirgesi yapısı)

Çizelge 151. Sabitlerin yapıları	
Ad	Yapı
MQDXP_STRUC_ID	"DXP↵"
MQDXP_STRUC_ID_ARRAY	'D', 'X', 'P', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 152. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQDXP_VERSION_1	1	X'00000001'
MQDXP_VERSION_2	2	X'00000002'
MQDXP_CURRENT_VERSION	2	X'00000002'

MQEC_* (Sinyal Değerleri)

Çizelge 153. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQEC_MSG_TESLIM ALINDI	2	X'00000002'
MQEC_WAIT_INTERVAL_SÜRESI doldu	3	X'00000003'
MQEC_WAIT_İPTAL edildi	4	X'00000004'
MQEC_Q_MGR QUIESCING	5	X'00000005'
MQEC_CONNECTION QUIESCING	6	X'00000006'

MQEI_* (Süre Bitimi)

Çizelge 154. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQEI_UNLIMITED	-1	X'FFFFFFFF'

MQENC_* (Kodlama)

MQENC_* (Kodlama)

Çizelge 155. Platforma göre değişmezlerin değerleri			
Ad	Hizmet olarak sunulan	Ondalık değer	Onaltılı değer
MQENC_NATIVE	IBM i	273	X'00000111'
	Linux	546	X'00000222'
	Linux SPARC üzerinde	273	X'00000111'
	Linux on x86	546	X'00000222'
	AIX and Linux	273	X'00000111'
	Windows	546	X'00000222'
	Micro Focus COBOL on Windows	17	X'00000011'
	z/OS	785	X'00000311'

MQENC_* (Kodlama Maskeleri)

Çizelge 156. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (İkili Tamsayılar için Encodings)

Çizelge 157. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQENC_INTEGER_UNDEFINED	0	X'00000000'
MQENC_INTEGER_NORMAL	1	X'00000001'
MQENC_INTEGER_REVERSED	2	X'00000002'

MQENC_* (Paketlenmiş Ondalık Tamsayılar için Encodings)

Çizelge 158. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQENC_DECIMAL_UNDEFINED	0	X'00000000'
MQENC_DECIMAL_NORMAL	16	X'00000010'
MQENC_DECIMAL_REVERSED	32	X'00000020'

MQENC_* (Kayan Nokta Numaraları için Kodlama)

Çizelge 159. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Yerleşik komut biçimi üstbilgi yapısı ve İşaretleri)

Gömülü komut biçimi üstbilgi yapısı

Çizelge 160. Sabitlerin yapıları	
Ad	Yapı
MQEPH_STRUC_ID	"EPH↵"
MQEPH_STRUC_ID_ARRAY	'E', 'P', 'H', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 161. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQEPH_STRUC_LENGTH_FIXED	68	X'00000044'
MQEPH_VERSION_1	1	X'00000001'
MQEPH_CURRENT_VERSION	1	X'00000001'

Yerleşik komut biçimi üstbilgi İşaretleri

Çizelge 162. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQEPH_NONE	0	X'00000000'
MQEPH_CCSDID_EMBEDDED	1	X'00000001'

MQET_* (Komut biçimi Escape Tipleri)

Çizelge 163. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQET_MQSC	1	X'00000001'

MQEVO_* (Komut biçimi Olay Kökenleri)

Çizelge 164. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQEVO_DİĞER	0	X'00000000'
MQEVO_CONSOLE	1	X'00000001'
MQEVO_INIT	2	X'00000002'
MQEVO_MSG	3	X'00000003'

Çizelge 164. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi Olay Kaydı)

Çizelge 165. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQEVR_DISABLED	0	X'00000000'
MQEVR_ENABLED	1	X'00000001'
MQEVR_KURAL Dışı DURUMU	2	X'00000002'
MQEVR_NO_DISPLAY	3	X'00000003'

MQEXPI_* (Süre Bitimi Tarama Aralığı)

Çizelge 166. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQEXPI_OFF	0	X'00000000'

MQFB_* (Geribildirim Değerleri)

Çizelge 167. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQFB_NONE	0	X'00000000'
MQFB_SYSTEM_FIRST	1	X'00000001'
MQFB_QUIT	256	X'00000100'
MQFB_SÜRE SONU	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_HATASI	266	X'0000010A'
MQFB_APPL_TYPE_HATA	267	X'0000010B'
MQFB_STOPPED_BY_MSG_EXIT	268	X'0000010C'
MQFB_ACTIVITY	269	X'0000010D'
MQFB_XMIT_Q_MSG_HATA	271	X'0000010F'
MQFB_PAN	275	X'00000113'
MQFB_NAN	276	X'00000114'
MQFB_DURDURED_BY_CHAD_EXIT	277	X'00000115'

Çizelge 167. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQFB_DURDURED_BY_PUBSUB_EXIT	279	X'00000117'
MQFB_NOT_A_REPOSITORY_MSG	280	X'00000118'
MQFB_BIND_OPEN_CLUSRCVR_DEL	281	X'00000119'
MQFB_MAX_ACTIVITIES	282	X'0000011A'
MQFB_NOT_ILETILDI	283	X'0000011B'
MQFB_NOT_TESLIM EDİLDİ	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_NEGATIF	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_HATA	296	X'00000128'
MQFB_NOT_AUTHORIZED_FOR_IMS	298	X'0000012A'
MQFB_IMS_HATASI	300	X'0000012C'
MQFB_IMS_FIRST	301	X'0000012D'
MQFB_IMS_SON	399	X'0000018F'
MQFB_CICS_INTERNAL_HATA	401	X'00000191'
MQFB_CICS_NOT_YETKILI	402	X'00000192'
MQFB_CICS_BRIDGE_FAILURE	403	X'00000193'
MQFB_CICS_CORREL_ID_HATA	404	X'00000194'
MQFB_CICS_CCSID_HATA	405	X'00000195'
MQFB_CICS_ENCODING_ERROR	406	X'00000196'
MQFB_CICS_CIH_HATASI	407	X'00000197'
MQFB_CICS_UOW_HATASI	408	X'00000198'
MQFB_CICS_COMMAREA_HATA	409	X'00000199'
MQFB_CICS_APPL_NOT_STARTED	410	X'0000019A'
MQFB_CICS_APPL_ABENDED	411	X'0000019B'
MQFB_CICS_DLQ_HATA	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'

Çizelge 167. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQFB_APPL_LAST	999999999	X'3B9AC9FF'

MQFC_* (Komut biçimi Zorlama Seçenekleri)

Çizelge 168. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQFC_YES	1	X'00000001'
MQFC_NO	0	X'00000000'

MQFMT_* (Biçimler)

Çizelge 169. Sabit adlar ve değerler

Ad	Değer
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-"
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	"MQHRF1-"
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','-'

Çizelge 169. Sabit adlar ve değerler (devamı var)

Ad	Değer
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_DIZGI_DIZISI	'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_DIZGI_DIZISI	'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','-', '-'

Not: - simgesi, tek bir boş karakteri temsil eder.

MQFUN_ * (Uygulama İşlevi Tipleri)

Çizelge 170. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_ * (Grup Özniteliği Seçicileri)

Çizelge 171. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQGA_FIRST	8001	X'00001F41'
MQGA_SON	9000	X'00002328'

MQGACF_ * (Komut biçimi Grup Değiştirgesi Tipleri)

Çizelge 172. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQGACF_FIRST	8001	X'00001F41'
MQGACF_COMMAND_CONTEXT	8001	X'00001F41'

Çizelge 172. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
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_KULLANILAN	8012	X'00001F4C'

MQGI_* (Grup Tanıtıcısı)

Çizelge 173. Sabit adlar ve değerler	
Ad	Değer
MQGI_NONE	X'00...00' (24 boş değer)
MQGI_NONE_ARRAY	'\0', '\0', ... (24 boş değer)

MQGMO_* (İleti seçeneklerini ve yapısını al)

İleti alma seçenekleri yapısı

Çizelge 174. Sabitlerin yapıları	
Ad	Yapı
MQGMO_STRUC_ID	"GMO-"
MQGMO_STRUC_ID_ARRAY	'G', 'M', 'O', '-'

Not: - simgesi, tek bir boş karakteri temsil eder.

Çizelge 175. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

İleti alma seçenekleri

Çizelge 176. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQGMO_WAIT	1	X'00000001'

Çizelge 176. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'00000800'
MQGMO_BROWSE_HANDLE	17825808	X'01100010'
MQGMO_BROWSE_CO_OP	18874384	X'01200010'
MQGMO_MSG_UNDER_CURSOR	256	X'00000100'
MQGMO_KİLİK	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_VAR	131072	X'00020000'
MQGMO_ALL_SEGMENTS_VAR	262144	X'00040000'
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_* (Grup Durumu)

Çizelge 177. Sabit adlar ve değerler

Ad	Değer
MQGS_NOT_IN_GROUP	'-'
MQGS_MSG_IN_GROUP	'G'
MQGS_LAST_MSG_IN_GROUP	'L'

Not: – simgesi, tek bir boş karakteri temsil eder.

MQHA_ * (İşleyici Seçicileri)

<i>Çizelge 178. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQHA_FIRST	4001	X'00000FA1'
MQHA_BAG_HANDLE	4001	X'00000FA1'
MQHA_LAST_KULLANILI	4001	X'00000FA1'
MQHA_SON	6000	X'00001770'

MQHB_ * (Çanta Tanıtıcıları)

<i>Çizelge 179. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQHB_UNUSABLE_HBAG	-1	X'FFFFFFFF'
MQHB_NONE	-2	X'FFFFFFFE'

MQHC_ * (Bağlantı İşleri)

<i>Çizelge 180. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQHC_DEF_HCONN	0	X'00000000'
MQHC_UNUSABLE_HCONN	-1	X'FFFFFFFF'
MQHC_UNASSOCIATED_HCONN	-3	X'FFFFFFFD'

MQHM_ * (İleti tanıtıcısı)

<i>Çizelge 181. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQHM_HABLE_HMSG	-1	X'FFFFFFFF'
MQHM_NONE	0	X'00000000'

MQHO_ * (Nesne Tanıtıcısı)

<i>Çizelge 182. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQHO_UNUSABLE_HOBJ	-1	X'FFFFFFFF'
MQHO_NONE	0	X'00000000'

MQHSTATE_ * (Komut biçimi İşleyici Durumları)

<i>Çizelge 183. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQHSTATE_INACTIVE	0	X'00000000'
MQHSTATE_ACTIVE	1	X'00000001'

MQIA_* (Tamsayı Öznitelik Seçicileri)

Çizelge 184. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'
MQ Adv. VUE MQ Adv. VUE MQIA_ADVANCED_CAPABILITY	273	X'00000111'
MQIA_AMQP_CAPABILITY	265	X'00000109'
MQIA_APPL_TYPE	1	X'00000001'
MQIA_ARCHIVE	60	X'0000003C'
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'

Çizelge 184. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 184. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 184. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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_CFCNLOS	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'

Çizelge 184. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 184. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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_* (Komut biçimi Tamsayı değiştirgesi tipleri)

Çizelge 185. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_Tümü	1009	X'000003F1'
MQIACF_EVENT_APPL_TYPE	1010	X'000003F2'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQIACF_EVENT_ORIGIN	1011	X'000003F3'
MQIACF_PARAMETER_ID	1012	X'000003F4'
MQIACF_ERROR_ID	1013	X'000003F5'
MQIACF_HATA_TANITICISI	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_NEDEN_NITELEYICISI	1020	X'000003FC'
MQIACF_XX_ENCODE_CASE_ONE komutu	1021	X'000003FD'
MQIACF_OPEN_OPTIONS	1022	X'000003FE'
MQIACF_OPEN_TIP	1023	X'000003FF'
MQIACF_PROCESS_Tnt	1024	X'00000400'
MQIACF_THREAD_ID	1025	X'00000401'
MQIACF_Q_STATUS_ATTRS	1026	X'00000402'
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_NEDEN_KODU	1072	X'00000430'
MQIACF_BRIDGE_TYPE	1073	X'00000431'
MQIACF_SORGUSU	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'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_KUYRUKLARI_	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'
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'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQIACF_KAPSAM dışı bırakılan_aralık	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_KULLANILAN	1141	X'00000475'
MQIACF_CF_STRUC_ENTRIES_MAX	1142	X'00000476'
MQIACF_CF_STRUC_ENTRIES_KULLANILAN	1143	X'00000477'
MQIACF_CF_STRUC_BACKUP_SIZE	1144	X'00000478'
MQIACF_MOVE_TYPE	1145	X'00000479'
MQIACF_MOVE_TYPE_TAŞIMA	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'
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_KULLANICI_TÜRÜ	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'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_TAPE	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_ETKİN	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_GERI	1188	X'000004A4'
MQIACF_SYSP_EXIT_INTERVAL	1189	X'000004A5'
MQIACF_SYSP_EXIT_GÖREVLERİ	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_MUHASEBE	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_IKINCİL	1210	X'000004BA'
MQIACF_SYSP_PROTECT	1211	X'000004BB'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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_ACTIVITIES	1236	X'000004D4'
MQIACF_DISCONTINUITY_COUNT	1237	X'000004D5'
MQIACF_ROUTE_ACCUMULATION	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'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQIACF_PERSISTENCE	1252	X'000004E4'
MQIACF_PRIORITY	1253	X'000004E5'
MQIACF_NEDEN_KODU	1254	X'000004E6'
MQIACF_REPORT	1255	X'000004E7'
MQIACF_Sürümü	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_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'
MQIACF_TOPIC_ATTRS	1269	X'000004F5'
MQIACF_PUBSUB_PROPERTIES	1271	X'000004F7'
MQIACF_DESTINATION_CLASS	1273	X'000004F9'
MQIACF_DURABLE_ABONELIK	1274	X'000004FA'
MQIACF_SUBSCRIPTION_SCOPE	1275	X'000004FB'
MQIACF_VARIABLE_USER_ID	1277	X'000004FD'
YALNIZCA 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_YAYINI	1300	X'00000514'
MQIACF_TOPIC_STATUS_ATTRS	1301	X'00000515'
MQIACF_TOPIC_STATUS_TYPE	1302	X'00000516'
MQIACF_SUB_OPTIONS	1303	X'00000517'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
MQIACF_POINTER_SIZE	1397	X'00000575'
MQIACF_REMOVE_AUTHREC	1398	X'00000576'
MQIACF_XATTRS	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_TIP	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'

Çizelge 185. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQIACF_LOG_IN_USE	1420	X'0000058C'
MQIACF_LOG_UTILIZATION	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_DENGELI	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'
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'
V9.3.0 V9.3.0 MQIACF_LAST_KULLANILDI	1444	X'000005A4'

MQIACH_* (Komut biçimi Tamsayı Kanal Tipleri)

Çizelge 186. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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'

Çizelge 186. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 186. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 186. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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'
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'

Çizelge 186. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQIACH_CHANNEL_SUMMARY_ATTRS	1642	X'0000066A'
MQIACH_PROTOCOL	1643	X'0000066B'
MQIACH_AMQPKEEPALIVE	1644	X'0000066C'
MQIACH_SECURITY_PROTOCOL	1645	X'0000066D'
z/OS MQIACH_SPL_PROTECTION	1646	X'0000066E'
MQIACH_LAST_USED	1646	X'0000066E'

MQIAMO_* (Komut biçimi Tamsayı İzleme Parametre Tipleri)

Çizelge 187. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'
MQIAMO_BROWSE_MIN_BYTES	707	X'000002C3'
MQIAMO_BROWSES_FAILED	708	X'000002C4'
MQIAMO_KAPANIYOR	709	X'000002C5'
MQIAMO_COMMIT	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_TOPLU işler	720	X'000002D0'
MQIAMO_GENERATED_MSGS	721	X'000002D1'
MQIAMO_TR	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_TOPLU işler	726	X'000002D6'
MQIAMO_INQS	727	X'000002D7'
MQIAMO_MSGS	728	X'000002D8'
MQIAMO_NET_TIME_AVG	729	X'000002D9'

Çizelge 187. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQIAMO_NET_TIME_MAX	730	X'000002DA'
MQIAMO_NET_TIME_MIN	731	X'000002DB'
MQIAMO_OBJECT_COUNT	732	X'000002DC'
MQIAMO_AÇIK	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'
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'

Çizelge 187. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_GÖNDERILDI	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'
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_GÖNDERILDI	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'

Çizelge 187. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQIAMO_MSGS_TESLIM	819	X'00000333'
MQIAMO_PKTS_IŞLENDI	820	X'00000334'
MQIAMO_PKTS_DLVD	821	X'00000335'
MQIAMO_PKTS_BIRAKILDI	822	X'00000336'
MQIAMO_PKTS_YINELENDI	823	X'00000337'
MQIAMO_NACKS_OLUŞTURULDU	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_ONARILDI	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_IŞLENDI	833	X'00000341'
MQIAMO_TOTAL_MSGS_SELECTED	834	X'00000342'
MQIAMO_TOTAL_MSGS_EXPIRED	835	X'00000343'
MQIAMO_TOTAL_MSGS_TESLIM	836	X'00000344'
MQIAMO_TOTAL_MSGS_İADE	837	X'00000345'
MQIAMO_LAST_KULLANILAN	837	X'00000345'

MQIAMO64_* (Komut biçimi 64 bitlik Tamsayı İzleme Parametre Tipleri)

Çizelge 188. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_* (Tamsayı Sistem Seçicileri)

Çizelge 189. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQIASI_FIRST	-1	X'FFFFFFFF'
MQIASY_CODED_CHAR_SET_ID	-1	X'FFFFFFFF'

Çizelge 189. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQIASY_TIP	-2	X'FFFFFFFFE'
MQIASY_XX_ENCODE_CASE_ONE komutu	-3	X'FFFFFFFFD'
MQIASY_MSG_SEQ_NUMBER	-4	X'FFFFFFFFC'
MQIASY_CONTROL	-5	X'FFFFFFFFB'
MQIASY_COMP_CODE	-6	X'FFFFFFFFA'
MQIASI_NEDENI	-7	X'FFFFFFFF9'
MQIASI_BAG_OPTIONS	-8	X'FFFFFFFF8'
MQIASY_VERSION	-9	X'FFFFFFFF7'
MQIASI_LAST_KULLANILDI	-9	X'FFFFFFFF7'
MQIASY_LAST	-2000	X'FFFFFF830'

MQIAUT_* (IMS bilgi üstbilgisi Kimlik Doğrulayıcı)

Çizelge 190. Sabit adlar ve değerler

Ad	Değer
MQIAUT_NONE	" "
MQIAUT_NONE_ARRAY	' ',' ',' ',' ',' ',' ',' ',' ',' ',' '

Not: ~ simgesi, tek bir boş karakteri temsil eder.

MQIAV_* (Tamsayı Öznitelik Değerleri)

Çizelge 191. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQIAV_NOT_UYGULANAMAZ	-1	X'FFFFFFFF'
MQIAV_UNDEFINED	-2	X'FFFFFFFFE'

MQICM_* (IMS bilgi üstbilgisi Kesinleştirme Kipleri)

Çizelge 192. Sabit adlar ve değerler

Ad	Değer
MQICM_COMMIT_THEN_SEND	'0'
MQICM_SEND_THEN_COMMIT	'1'

MQIDO_* (Komut biçimi belirsiz seçenekleri)

Çizelge 193. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQIDO_COMMIT	1	X'00000001'
MQIDO_BACKOUT	2	X'00000002'

MQIEP_ * (Arabirim giriş noktaları)

Bağlantı güvenliği parametreleri yapısı

Çizelge 194. Sabitlerin yapıları	
Ad	Yapı
MQIEP_STRUC_ID	"IEP↵"
MQIEP_STRUC_ID_ARRAY	'I', 'E', 'P', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 195. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIEP_VERSION_1	1	X'00000001'
MQDXP_CURRENT_VERSION	1	X'00000001'

MQIGQ_ * (Grup içi kuyruğa alma)

Çizelge 196. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIGQ_DISABLED	0	X'00000000'
MQIGQ_ENABLED	1	X'00000001'

MQIGQPA_ * (Grup içi kuyruğa alma koyma yetkisi)

Çizelge 197. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIGQPA_DEFAULT	1	X'00000001'
MQIGQPA_CONTEXT	2	X'00000002'
MQIGQPA_ONLY_IGQ	3	X'00000003'
MQIGQPA_ALTERNATE_OR_IGQ	4	X'00000004'

MQIIH_ * (IMS bilgi üstbilgisi yapısı ve işaretleri)

IMS bilgi üstbilgisi yapısı

Çizelge 198. Sabitlerin yapıları	
Ad	Yapı
MQIIH_STRUC_ID	"IIH↵"
MQIIH_STRUC_ID_ARRAY	'I', 'I', 'H', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 199. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIIH_VERSION_1	1	X'00000001'
MQIIH_CURRENT_VERSION	1	X'00000001'
MQIIH_LENGTH_1	84	X'00000054'

IMS bilgi üstbilgisi İşaretleri

Çizelge 200. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIIH_NONE	0	X'00000000'
MQIIH_PASS_SÜRE SONU	1	X'00000001'
MQIIH_UNLIMITED_SON kullanma tarihi	0	X'00000000'
MQIIH_REPLY_FORMAT_NONE	8	X'00000008'
MQIIH_IGNORE_PURG	16	X'00000010'
MQIIH_CM0_REQUEST_RESPONSE	32	X'00000020'

MQIMPO_* (İleti özelliği seçeneklerini ve yapısını sorgularken)

İleti sorma özelliği seçenekleri yapısı

Çizelge 201. Sabitlerin yapıları	
Ad	Yapı
MQIMPO_STRUC_ID	"IMPO"
MQIMPO_STRUC_ID_ARRAY	'I', 'M', 'P', 'O'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 202. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIMPO_VERSION_1	1	X'00000001'
MQIMPO_CURRENT_VERSION	1	X'00000001'

Sorma İletisi Özellik Seçenekleri

Çizelge 203. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIMPO_CONVERT_TYPE	2	X'00000002'
MQIMPO_SORGU_UZUNLUĞU	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_* (Komut biçimi Gelen Düzenleme)

Çizelge 204. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQINBD_Q_MGR	0	X'00000000'
MQINBD_GROUP	3	X'00000003'

MQIND_* (Özel Dizin Değerleri)

Çizelge 205. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIND_NONE	-1	X'FFFFFFFF'
MQIND_Tümü	-2	X'FFFFFFFE'

MQIPADDR_* (IP Adresi Sürümleri)

Çizelge 206. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIPADDR_IPV4	0	X'00000000'
MQIPADDR_IPV6	1	X'00000001'

MQISS_* (IMS bilgi üstbilgisi Güvenlik Kapsamları)

Çizelge 207. Sabit adlar ve değerler	
Ad	Değer
MQISS_CHECK	'C'
MQISS_FULL	'F'

MQIT_* (Dizin Tipleri)

Çizelge 208. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQIT_NONE	0	X'00000000'
MQIT_MSG_ID	1	X'00000001'
MQIT_CORREL_ID	2	X'00000002'
MQIT_MSG_TOKEN	4	X'00000004'
MQIT_GRUP_Tnt	5	X'00000005'

MQITEM_* (mqInquireItemInfo için Öğe Tipi)

Çizelge 209. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_DIZGI_IFADESI süzgeci	6	X'00000006'
MQITEM_INTEGER64	7	X'00000007'
MQITEM_BYTE_STRING_FILTER	8	X'00000008'

MQITII_* (IMS bilgi üstbilgisi İşlem Eşgörünümü Tanıtıcısı)

Çizelge 210. Sabit adlar ve değerler	
Ad	Değer
MQITII_NONE	X'00...00' (16 boş değer)
MQITII_NONE_ARRAY	'\0', '\0', ... (16 boş değer)

MQITS_* (IMS bilgi üstbilgisi İşlem Durumları)

Çizelge 211. Sabit adlar ve değerler	
Ad	Değer
MQITS_IN_ETKİLEŞİM	'C'
MQITS_NOT_IN_ETKİLEŞİM	'-'
MQITS_MİMARILMIŞ	'A'

Not: - simgesi, tek bir boş karakteri temsil eder.

MQKAI_* (KeepAlive Aralığı)

Çizelge 212. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQKAI_AUTO	-1	X'FFFFFFFF'

MQMASTER_* (Ana denetim)

Çizelge 213. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMASTER_NO	0	X'00000000'
MQMASTER_YES	1	X'00000001'

MQMCAS_* (Komut biçimi İleti Kanal Aracısı Durumu)

Çizelge 214. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMCAS_DURDURULDU	0	X'00000000'
MQMCAS_RUNNING	3	X'00000003'

MQMCAT_* (MCA Tipleri)

Çizelge 215. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMCAT_PROCESS	1	X'00000001'
MQMCAT_THREAD	2	X'00000002'

MQMCD_* (Yayınlama/Abone Olma Seçenekleri Etiket Bilgileri)

Yayınlama/Abone Olma Seçenekleri Etiket İleti İçerik Açıklayıcı (mcd) Etiketler

Çizelge 216. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMCD_FOLDER_VERSION	1	X'00000001'

Yayınlama/Abone Olma Seçenekleri Etiket Adları

Çizelge 217. Sabit adlar ve değerler	
Ad	Değer
MQMCD_MSG_DOMAIN	"Msd"
MQMCD_MSG_SET	"Set"
MQMCD_MSG_TYPE	"Type"
MQMCD_MSG_FORMAT	"Fmt"

Yayınlama/Abone Olma Seçenekleri Etiket XML etiketi adları

Çizelge 218. Sabit adlar ve değerler	
Ad	Değer
MQMCD_MSG_DOMAIN_B	"<Msd>"
MQMCD_MSG_DOMAIN_E	"</Msd>"
MQMCD_MSG_SET_B	"<Set>"
MQMCD_MSG_SET_E	"</Set>"
MQMCD_MSG_TYPE_B	"<Type>"
MQMCD_MSG_TYPE_E	"</Type>"
MQMCD_MSG_FORMAT_B	"<Fmt>"
MQMCD_MSG_FORMAT_E	"</Fmt>"

Yayınlama/Abone Olma Seçenekleri Etiket Etiket değeri

Çizelge 219. Sabit adlar ve değerler	
Ad	Değer
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_* (İleti tanımlayıcı yapısı)

Çizelge 220. Sabitlerin yapıları	
Ad	Yapı
MQMD_STRUC_ID	"MD↵"
MQMD_STRUC_ID_ARRAY	'M', 'D', '↵', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 221. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMD_VERSION_1	1	X'00000001'
MQMD_VERSION_2	2	X'00000002'
MQMD_CURRENT_VERSION	2	X'00000002'

MQMDE_* (İleti tanımlayıcı uzantısı yapısı)

Çizelge 222. Sabitlerin yapıları	
Ad	Yapı
MQMDE_STRUC_ID	"MDE↵"
MQMDE_STRUC_ID_ARRAY	'M', 'D', 'E', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 223. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMDE_VERSION_2	2	X'00000002'
MQMDE_CURRENT_VERSION	2	X'00000002'
MQMDE_LENGTH_2	72	X'00000048'

MQMDEF_* (İleti tanımlayıcı uzantısı işaretleri)

Çizelge 224. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMDEF_NONE	0	X'00000000'

MQMDS_* (İleti Teslim Sırası)

Çizelge 225. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMDS_PRIORITY	0	X'00000000'
MQMDS_FIFO	1	X'00000001'

MQMF_* (İleti İşaretleri)

Çizelge 226. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMF_SEGMENTATION_INHIBE	0	X'00000000'
MQMF_SEGMENTATION_ALLOWED	1	X'00000001'

Çizelge 226. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
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_* (Arabellek seçenekleri ve yapısı için ileti tanıtıcısı)

Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısı

Çizelge 227. Sabitlerin yapıları	
Ad	Yapı
MQMHBO_STRUC_ID	"MHBO"
MQMHBO_STRUC_ID_ARRAY	'M', 'H', 'B', 'O'

Not: ↯ simgesi, tek bir boş karakteri temsil eder.

Çizelge 228. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMHBO_VERSION_1	1	X'00000001'
MQMHBO_CURRENT_VERSION	1	X'00000001'

İleti Tanıtıcısı-Arabellek Seçenekleri

Çizelge 229. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMHBO_PROPERTIES_IN_MQRFH2	1	X'00000001'
MQMHBO_DELETE_PROPERTIES	2	X'00000002'
MQMHBO_NONE	0	X'00000000'

MQMI_* (İleti Tanıtıcısı)

Çizelge 230. Sabit adlar ve değerler	
Ad	Değer
MQMI_NONE	X'00...00' (24 boş değer)
MQMI_NONE_ARRAY	'\0', '\0', ... (24 boş değer)

MQMMBI_* (İleti İşaret-Göz At Aralığı)

Çizelge 231. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMMBI_SINIRSIZ	-1	X'FFFFFFFF'

MQMO_* (Eşleştirme Seçenekleri)

Çizelge 232. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi kip seçenekleri)

Çizelge 233. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMODE_FORCE	0	X'00000000'
MQMODE_QUIESCE	1	X'00000001'
MQMODE_TERMINATE	2	X'00000002'

MQMON_* (İzleme Değerleri)

Çizelge 234. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQMON_KULLANILAMAZ	-1	X'FFFFFFFF'
MQMON_YOK	-1	X'FFFFFFFF'
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_ORTAMI	33	X'00000021'
MQMON_YükSEKLIK	65	X'00000041'

MQMT_* (İleti Tipleri)

Çizelge 235. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

Çizelge 235. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_* (İleti Simgesi)

Çizelge 236. Sabit adlar ve değerler

Ad	Değer
MQMTOK_NONE	X'00...00' (16 boş değer)
MQMTOK_NONE_ARRAY	'\0', '\0', ... (16 boş değer)

MQNC_* (Ad Sayısı)

Çizelge 237. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQNC_MAX_NAMELIST_NAME_COUNT	256	X'00000100'

MQNPM_* (Kalıcı Olmayan İleti Sınıfı)

Çizelge 238. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQNPM_CLASS_NORMAL	0	X'00000000'
MQNPM_CLASS_HIGH	10	X'0000000A'

MQNPMS_* (NonPersistent-İleti Hızları)

Çizelge 239. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQNPMS_NORMAL	1	X'00000001'
MQNPMS_FAST	2	X'00000002'

MQNT_* (Ad Listesi Tipleri)

Çizelge 240. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQNT_NONE	0	X'00000000'
MQNT_Q	1	X'00000001'
MQNT_CLUSTER	2	X'00000002'
MQNT_AUTH_INFO	4	X'00000004'
MQNT_Tümü	1001	X'000003E9'

MQNV\$_* (Ad/Deęer Dizesi Adları)

Çizelge 241. Sabit adlar ve deęerler	
Ad	Deęer
MQNV\$_APPL_TYPE	"OPT_APP_GRP~"
MQNV\$_MSG_TYPE	"OPT_MSG_TYPE~"

Not: ~ simgesi, tek bir boş karakteri temsil eder.

MQOA_* (Nesne Özniteliklerine İlişkin Seçicilere İlişkin Sınırlar)

Çizelge 242. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQOA_FIRST	1	X'00000001'
MQOA_SON	9000	X'00002328'

MQOD_* (Nesne tanımlayıcı yapısı)

Çizelge 243. Sabitlerin yapıları	
Ad	Yapı
MQOD_STRUC_ID	"OD~"
MQOD_STRUC_ID_ARRAY	'0', 'D', '~', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 244. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQOD_VERSION_1	1	X'00000001'
MQOD_VERSION_2	2	X'00000002'
MQOD_VERSION_3	3	X'00000003'
MQOD_VERSION_4	4	X'00000004'
MQOD_CURRENT_VERSION	4	X'00000004'
MQOD_CURRENT_LENGTH	(value differs by platform or version)	(value differs by platform or version)

MQOII_* (Nesne Eşgörünümü Tanıtıcısı)

Çizelge 245. Sabit adlar ve deęerler	
Ad	Deęer
MQII_NONE	X'00...00' (24 boş deęer)
MQII_NONE_ARRAY	'\0', '\0', ... (24 boş deęer)

MQOL_* (Özgün Uzunluk)

Çizelge 246. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQOL_UNDEFINED	-1	X'FFFFFFF'

MQOM_* (Sorma Grubunda Eski Db2 İleti seçenekleri)

Çizelge 247. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQOM_NO	0	X'00000000'
MQOM_YES	1	X'00000001'

MQOO_* (Açma Seçenekleri)

Çizelge 248. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_GÖZ AT	8	X'00000008'
MQOO_ÇıKışı	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'
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_ÖNDEN	524288	X'00080000'
MQOO_READ_ÖNDEN	1048576	X'00100000'
MQOO_BIND_ON_GROUP	4194304	X'00400000'

MQOO_* (Yalnızca C++ dilinde kullanılan aşağıdakiler)

Çizelge 249. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQOO_RESOLVE_NAMES	65536	X'00010000'
MQOO_RESOLVE_LOCAL_Q	262144	X'00040000'

MQOP_* (MQCTL ve MQCB için işlem kodları)

MQCTL için işlem kodları

Çizelge 250. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQOP_START	1	X'00000001'
MQOP_START_WAIT	2	X'00000002'
MQOP_DUR	4	X'00000004'

MQCB için işlem kodları

Çizelge 251. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQOP_REGISTER	256	X'00000100'
MQOP_DEREGISTER	512	X'00000200'

MQCTL ve MQCB için işlem kodları

Çizelge 252. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQOP_SUSPEND	65536	X'00010000'
MQOP_RESUME	131072	X'00020000'

MQOPEN_* (MQOPEN_PRIV yapısıyla ilgili değerler)

Çizelge 253. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQOPEN_PRIV_VERSION_1	1	X'00000001'
MQOPEN_PRIV_CURRENT_VERSION	1	X'00000001'

MQOPER_* (Etkinlik İşlemleri)

Çizelge 254. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQOPER_SYSTEM_FIRST	0	X'00000000'
MQOPER_BILINMIYOR	0	X'00000000'
MQOPER_GÖZ At	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'

Çizelge 254. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQOPER_PUBLISH	10	X'0000000A'
MQOPER_EXCLUDED_PUBLISH	11	X'0000000B'
MQOPER_DISCARDED_PUBLISH	12	X'0000000C'
MQOPER_SISTEM_SON	65535	X'0000FFFF'
MQOPER_APPL_FIRST	65536	X'00010000'
MQOPER_APPL_SON	99999999	X'3B9AC9FF'

MQOT_* (Nesne Tipleri ve Genişletilmiş Nesne Tipleri)

Nesne Tipleri

Çizelge 255. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_KANAL	6	X'00000006'
MQOT_AUTH_INFO	7	X'00000007'
MQOT_KONU	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'

Genişletilmiş Nesne Tipleri

Çizelge 256. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQOT_TÜMÜ	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_SUNUCU_KANAL	1008	X'000003F0'
MQOT_ISTEK_KANAL	1009	X'000003F1'
MQOT_RECEIVER_CHANNEL	1010	X'000003F2'
MQOT_CURRENT_CHANNEL	1011	X'000003F3'
MQOT_SAVED_CHANNEL	1012	X'000003F4'

Çizelge 256. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQOT_SVRCONN_CHANNEL	1013	X'000003F5'
MQOT_CLNTCONN_CHANNEL	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_CHANNEL	1021	X'000003FD'
MQOT_AUTH_REC	1022	X'000003FE'

MQPA_* (Put Yetkisi)

Çizelge 257. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPA_DEFAULT	1	X'00000001'
MQPA_CONTEXT	2	X'00000002'
MQPA_ONLY_MCA	3	X'00000003'
MQPA_ALTERNATE_OR_MCA	4	X'00000004'

MQPD_* (Özellik tanımlayıcısı, destek ve bağlam)

Özellik tanımlayıcı yapısı

Çizelge 258. Sabitlerin yapıları	
Ad	Yapı
MQPD_STRUC_ID	"PD↵↵"
MQPD_STRUC_ID_ARRAY	'P', 'D', '↵', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 259. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPD_VERSION_1	1	X'00000001'
MQPD_CURRENT_VERSION	1	X'00000001'

Özellik Tanımlayıcı Seçenekleri

Çizelge 260. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPD_NONE	0	X'00000000'

Özellik Desteği Seçenekleri

Çizelge 261. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

Özellik Bağlamı

Çizelge 262. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPD_NO_CONTEXT	0	X'00000000'
MQPD_USER_CONTEXT	1	X'00000001'

MQPER_* (Kalıcılık Değerleri)

Çizelge 263. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPER_PERSISTENCE_AS_PARENT	-1	X'FFFFFFFF'
MQPER_NOT_KALICI	0	X'00000000'
MQPER_KALICI	1	X'00000001'
MQPER_PERSISTENCE_AS_Q_DEF	2	X'00000002'
MQPER_PERSISTENCE_AS_TOPIC_DEF	2	X'00000002'

MQPL_* (Platformlar)

Çizelge 264. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPL_MVS	1	X'00000001'
MQPL_OS390	1	X'00000001'
MQPL_ZOS	1	X'00000001'
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'

Çizelge 264. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQPL_TPF	23	X'00000017'
MQPL_VSE	27	X'0000001B'
MQPL_APPLIANCE	28	X'0000001C'
MQPL_NAT	1	X'00000001'

MQPMO_* (Yayınlama maskesi için ileti seçeneklerini ve yapısını koyun)

Koyma iletisi seçenekleri yapısı

Çizelge 265. Sabitlerin yapıları	
Ad	Yapı
MQPMO_STRUC_ID	"PMO↯"
MQPMO_STRUC_ID_ARRAY	'P', 'M', 'O', '↯'

Not: ↯ simgesi, tek bir boş karakteri temsil eder.

Çizelge 266. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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)

Koyma İletisi Seçenekleri

Çizelge 267. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPMO_SYNCPOINT	2	X'00000002'
MQPMO_NO_SYNCPOINT	4	X'00000004'
MQPMO_DEFAULT_CONTEXT	32	X'00000020'
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 (mqpmo_async_yaniti)	65536	X'00010000'

Çizelge 267. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQPMO_SYNC_RESPONSE	131072	X'00020000'
MQPMO_RESOLVE_LOCAL_Q	262144	X'00040000'
MQPMO_RETAIN	2097152	X'00200000'
YALNIZCA MQPMO_MD_FOR_OUTPUT_YALNIZCA	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'

Yayınlama maskesi için İleti Seçeneklerini Belirle

Çizelge 268. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPMO_PUB_OPTIONS_MASK	2097152	X'00200000'

MQPMRF_* (İleti Kaydı Alanlarını Koy)

Çizelge 269. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi Temizleme Seçenekleri)

Çizelge 270. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPO_YES	1	X'00000001'
MQPO_NO	0	X'00000000'

MQPRI_* (Öncelik)

Çizelge 271. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Kuyruk ve Kanal Özellik Denetimi Değerleri ve Özellik Uzunluğu Üst Sınırı)

Kuyruk ve Kanal Özellik Denetimi Değerleri

Çizelge 272. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPROP_COMPATIBILITY	0	X'00000000'
MQPROP_NONE	1	X'00000001'
MQPROP_Tümü	2	X'00000002'
MQPROP_FORCE_MQRFH2	3	X'00000003'

Özellik Uzunluğu Üst Sınırı

Çizelge 273. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPROP_UNRESTRICTED_LENGTH	-1	X'FFFFFFFF'

MQPRT_* (Put Response Values)

Çizelge 274. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPRT_RESPONSE_AS_PARENT	0	X'00000000'
MQPRT_SYNC_RESPONSE	1	X'00000001'
MQPRT_ASYNC_RESPONSE (mqprt_async_yanıtı)	2	X'00000002'

MQPS_* (Yayınla/Abone)

Komut Biçimi Yayınla/Abone Olma Durumu

Çizelge 275. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPS_STATUS_INACTIVE	0	X'00000000'
MQPS_STATUS_STARTING	1	X'00000001'
MQPS_STATUS_DURDURULUYOR	2	X'00000002'
MQPS_STATUS_ETKINLEŞTİRİCİ	3	X'00000003'
MQPS_STATUS_COMPAT	4	X'00000004'
MQPS_STATUS_HATA	5	X'00000005'
MQPS_STATUS_REDEDİLDİ	6	X'00000006'

Etiketleri Dize Olarak Yayınla/Abone Ol

MQPS_COMMAND	"MQPSCommand"
MQPS_COMP_CODE	"MQPSCompCode"
MQPS_CORREL_ID	"MQPSCorrelId"
MQPS_DELETE_OPTIONS	"MQPSDe10pts"

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_ADİ	"MQPSQMgrName"
MQPS_Q_NAME	"MQPSQName"
MQPS_REASON	"MQPSReason"
MQPS_NEDEN_METNI	"MQPSReasonText"
MQPS_REGISTRATION_OPTIONS	"MQPSRegOpts"
MQPS_SEQUENCE_NUMBER	"MQPSSeqNum"
MQPS_STREAM_NAME	"MQPSStreamName"
MQPS_STRING_VERİ	"MQPSStringData"
MQPS_SUBSCRIPTION_IDENTITY	"MQPSSubIdentity"
MQPS_SUBSCRIPTION_NAME	"MQPSSubName"
MQPS_SUBSCRIPTION_USER_DATA	"MQPSSubUserData"
MQPS_TOPIC	"MQPSTopic"
MQPS_USER_ID	"MQPSUserId"

Etiketleri boş-ekli dizgiler olarak yayınla/abone ol

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	"-MQSParmId-"
MQPS_YAYINLAMA_SEÇENEKLERİ_B	"-MQSPubOpts-"
MQPS_PUBLISH_TIMESTAMP_B	"-MQSPubTime-"
MQPS_Q_MGR_NAME_B	"-MQPSQMgrName-"
MQPS_Q_NAME_B	"-MQPSQName-"
MQPS_NEDENİ_B	"-MQPSReason-"
MQPS_NEDEN_METİN_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~"

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Komut Etiket Değerlerini Dizgi Olarak Yayınla/Abone Ol

MQPS_DELETE_YAYINI	"DeletePub"
MQPS_DEREGISTER_PUBLISHER	"DeregPub"
MQPS_DEREGISTER_SUBSCRIBER	"DeregSub"
MQPS_PUBLISH	"Publish"
MQPS_REGISTER_PUBLISHER	"RegPub"
MQPS_REGISTER_SUBSCRIBER	"RegSub"
MQPS_REQUEST_UPDATE	"ReqUpdate"

Yayınlama/Abone Olma Komutu Etiket Değerleri boş-kapalı dizgiler olarak

MQPS_DELETE_PUBLICATION_B	"~DeletePub~"
MQPS_DEREGISTER_PUBLISHER_B	"~DeregPub~"
MQPS_DEREGISTER_SUBSCRIBER_B	"~DeregSub~"
MQPS_YAYIN_B	"~Publish~"
MQPS_REGISTER_PUBLISHER_B	"~RegPub~"
MQPS_REGISTER_SUBSCRIBER_B	"~RegSub~"
MQPS_REQUEST_UPDATE_B	"~ReqUpdate~"

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Yayınlama/Abone Olma Seçenekleri Etiket Değerleri

MQPS_ADD_NAME	"AddName"
MQPS_ANONIM	"Anon"
MQPS_CORREL_ID_AS_IDENTITY	"CorrelAsId"
MQPS_DEREGISTER_ALL	"DeregAll"
MQPS_DIRECT_REQUESTS	"DirectReq"
MQPS_DUPLICATES_OK	"DupsOK"

MQPS_FULL_RESPONSE (mqps_tam_yanit)	"FullResp"
MQPS_INCLUDE_STREAM_NAME	"InclStreamName"
MQPS_INFORM_IF_KORUNDU	"InformIfRet"
MQPS_IS_RETAINED_YAYINI	"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_KAYDI	"NoReg"
MQPS_NON_PERSISTENT	"NonPers"
MQPS_NONE	"None"
MQPS_OTHER_SUBSCRIBERS_ONLY	"OtherSubsOnly"
MQPS_KALICI	"Pers"
MQPS_PERSISTENT_AS_PUBLISH	"PersAsPub"
MQPS_PERSISTENT_AS_Q	"PersAsQueue"
MQPS_PUBLISH_ON_REQUEST_ONLY	"PubOnReqOnly"
MQPS_RETAIN_YAYINI	"RetainPub"
MQPS_VARIABLE_USER_ID	"VariableUserId"

Yayınlama/Abone Olma Seçenekleri Etiket Değerleri boş-kapalı dizgiler olarak

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_INFORM_IF_RETAINED_B	"¬InformIfRet¬"
MQPS_IS_RETAINED_PUBLICATION_B	"¬IsRetainedPub¬"
MQPS_JOIN_HARİÇ tutma_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¬"

Not: ¬ simgesi, tek bir boş karakteri temsil eder.

MQPSC_* (Yayınlama/Abone Olma Seçenekleri Etiketleri Yayınla/Abone Ol Komut Klasörü (Psc) Etiketleri)

<i>Çizelge 276. Değişmezlerin değerleri</i>		
Ad	Ondalık değer	Onaltılı değer
MQPSC_FOLDER_VERSION	1	X'00000001'

MQPSC_* (Yayınlama/Abone Olma Seçenekleri Etiket Adları)

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_ * (Yayınlama/Abone Olma Seçenekleri Etiket XML etiketi adları)

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_ * (Yayınlama/Abone Olma Seçenekleri Etiket Yayınlayıcı Değerleri (dizgi olarak))

MQPSC_DELETE_PUBLICATION	"DeletePub"
--------------------------	-------------

MQPSC_DEREGISTER_SUBSCRIBER	"DeregSub"
MQPSC_PUBLISH	"Publish"
MQPSC_REGISTER_SUBSCRIBER	"RegSub"
MQPSC_REQUEST_UPDATE	"ReqUpdate"

MQPSC_* (Yayınlama/Abone Olma Seçenekleri Etiket Adı Değerleri dizgi olarak)

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 KORUNDU	"InformIfRet"
MQPSC_IS_RETAINED_PUB	"IsRetainedPub"
MQPSC_JOIN_SHARED	"JoinShared"
MQPSC_JOIN_EXCLUSIVE	"JoinExcl"
YALNIZCA MQPSC_LEAVE_YALNIZCA	"LeaveOnly"
MQPSC_LOCAL	"Local"
MQPSC_LOCKED	"Locked"
MQPSC_NEW_PUBS_YALNIZCA	"NewPubsOnly"
MQPSC_NO_ALTERATION	"NoAlter"
MQPSC_NON_PERSISTENT	"NonPers"
YALNIZCA MQPSC_OTHER_SUBS_	"OtherSubsOnly"
MQPSC_KALICI	"Pers"
MQPSC_PERSISTENT_AS_PUBLISH	"PersAsPub"
MQPSC_PERSISTENT_AS_Q	"PersAsQueue"
MQPSC_NONE	"None"
YALNIZCA MQPSC_PUB_ON_REQUEST_ONLY	"PubOnReqOnly"
MQPSC_RETAIN_PUB	"RetainPub"
MQPSC_VARIABLE_USER_ID	"VariableUserId"

MQPSCR_* (Yayınlama/Abone Olma Seçenekleri)

Yayınlama/Abone Olma Seçenekleri Etiket Yayınlama/Abone Olma Yanıt Klasörü (Pscr) Etiketleri

Çizelge 277. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPSCR_FOLDER_VERSION	1	X'00000001'

Yayınlama/Abone Olma Seçenekleri Etiket Adları

MQPSCR_COMPLETION	"Completion"
MQPSCR_RESPONSE	"Response"
MQPSCR_REASON	"Reason"

Yayınlama/Abone Olma Seçenekleri Etiket XML etiketi adları

MQPSCR_COMPLETION_B	"<Completion>"
MQPSCR_COMPLETION_E	"</Completion>"
MQPSCR_RESPONSE_B	"<Response>"
MQPSCR_RESPONSE_E	"</Response>"
MQPSCR_REASON_B	"<Reason>"
MQPSCR_REASON_E	"</Reason>"

Yayınlama/Abone Olma Seçenekleri Etiket Etiket değeri

MQPSCR_OK	"ok"
MQPSCR_UYARI	"warning"
MQPSCR_HATASI	"error"

MQPSM_* (Pub/Alt Kipi)

Çizelge 278. Değişmezlerin değeri		
Ad	Ondalık değeri	Onaltılı değeri
MQPSM_DISABLED	0	X'00000000'
MQPSM_COMPAT	1	X'00000001'
MQPSM_ENABLED	2	X'00000002'

MQPSPROP_* (Pub/Alt İleti Özellikleri)

Çizelge 279. Değişmezlerin değeri		
Ad	Ondalık değeri	Onaltılı değeri
MQPSPROP_NONE	0	X'00000000'
MQPSPROP_COMPAT	1	X'00000001'
MQPSPROP_RFH2	2	X'00000002'
MQPSPROP_MSGPROP	3	X'00000003'

MQPSST_* (Komut biçimi Pub/Alt Durum Tipi)

Çizelge 280. Değişmezlerin değeri		
Ad	Ondalık değeri	Onaltılı değeri
MQPSST_Tümü	0	X'00000000'
MQPSST_LOCAL	1	X'00000001'
MQPSST_üst ÖGESİ	2	X'00000002'

Çizelge 280. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQPSST_CHILD	3	X'00000003'

MQPUBO_* (Yayınlama/Abone Olma Yayın Seçenekleri)

Çizelge 281. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPUBO_NONE	0	X'00000000'
MQPUBO_CORREL_ID_AS_IDENTITY	1	X'00000001'
MQPUBO_RETAIN_YAYINI	2	X'00000002'
MQPUBO_OTHER_SUBSCRIBERS_ONLY	4	X'00000004'
MQPUBO_NO_KAYDI	8	X'00000008'
MQPUBO_IS_RETAINED_YAYINI	16	X'00000010'

MQPXP_* (Yayınlama/abone olma yöneltme çıkışı değiştirgesi yapısı)

Çizelge 282. Sabitlerin yapıları	
Ad	Yapı
MQPXP_STRUC_ID	"PXP¬"
MQPXP_STRUC_ID_ARRAY	'P', 'X', 'P', '¬'

Not: ¬ simgesi, tek bir boş karakteri temsil eder.

Çizelge 283. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQPXP_VERSION_1	1	X'00000001'
MQPXP_CURRENT_VERSION	1	X'00000001'

MQQA_* (Kuyruk öznitelikleri)

Değerleri Almayı Engelle

Çizelge 284. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQA_GET_INENGELLEME	1	X'00000001'
MQQA_GET_ALLOWED	0	X'00000000'

Koyma Değerlerini Engelle

Çizelge 285. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQA_PUT_INENGELLEME	1	X'00000001'
MQQA_PUT_ALLOWED	0	X'00000000'

Kuyruk Paylaşılabilir

Çizelge 286. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQA_SHAREABLE	1	X'00000001'
MQQA_NOT_SHAREABLE	0	X'00000000'

Geri Çekilme Hardening

Çizelge 287. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQA_BACKOUT_HARDENED	1	X'00000001'
MQQA_BACKOUT_NOT_SERTLEŞTİRİLMİŞ	0	X'00000000'

MQQDT_* (Kuyruk Tanımlaması Tipleri)

Çizelge 288. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQDT_ÖNCEDEDEN TANIMLI	1	X'00000001'
MQQDT_PERMANENT_DYNAMIC	2	X'00000002'
MQQDT_TEMPORARY_DYNAMIC	3	X'00000003'
MQQDT_SHARED_DYNAMIC	4	X'00000004'

MQQF_* (Kuyruk İşaretleri)

Çizelge 289. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQF_LOCAL_Q	1	X'00000001'
MQQF_CLWL_USEQ_ANY	64	X'00000040'
MQQF_CLWL_USEQ_LOCAL	128	X'00000080'

MQQMDT_* (Komut biçimi Kuyruk Yöneticisi Tanımlama Tipleri)

Çizelge 290. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQMDT_BELIRTIC_CLUSTER_SENDER	1	X'00000001'
MQQMDT_AUTO_CLUSTER_SENDER	2	X'00000002'
MQQMDT_AUTO_EXP_CLUSTER_SENDER	4	X'00000004'
MQQMDT_CLUSTER_ALICI	3	X'00000003'

MQQMF_* (Kuyruk Yöneticisi İşaretleri)

Çizelge 291. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQMF_REPOSITORY_Q_MGR	2	X'00000002'
MQQMF_CLUSSDR_USER_DEFINED	8	X'00000008'
MQQMF_CLUSSDR_AUTO_DEFINED	16	X'00000010'

Çizelge 291. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQQMF_KULLANILABILIR	32	X'00000020'

MQQMFACT_* (Komut biçimi Kuyruk Yöneticisi Olanığı)

Çizelge 292. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQMFACT_IMS_BRIDGE	1	X'00000001'
MQQMFACT_DB2	2	X'00000002'

MQQMSTA_* (Komut biçimi Kuyruk Yöneticisi Durumu)

Çizelge 293. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQMSTA_BASLATILYOR	1	X'00000001'
MQQMSTA_RUNNING	2	X'00000002'
MQQMSTA_QUIESCING	3	X'00000003'

MQQMT_* (Komut biçimi Kuyruk Yöneticisi Tipleri)

Çizelge 294. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQMT_NORMAL	0	X'00000000'
MQQMT_REPOSITORY	1	X'00000001'

MQQO_* (Komut biçimi Susturma Seçenekleri)

Çizelge 295. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQO_YES	1	X'00000001'
MQQO_NO	0	X'00000000'

MQQSGD_* (Kuyruk paylaşım grubu yok etme)

Çizelge 296. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi kuyruk paylaşım grubu durumu)

Çizelge 297. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQSGS_UNKNOWN	0	X'00000000'
MQQSGS_CREATED	1	X'00000001'
MQQSGS_ETKİNLEŞTİRİLDİ	2	X'00000002'
MQQSGS_INACTIVE	3	X'00000003'
MQQSGS_FAILED	4	X'00000004'
MQQSGS_PENDING	5	X'00000005'

MQQSIE_* (Komut biçimi Kuyruk Hizmeti-Aralık Olayları)

Çizelge 298. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQSIE_NONE	0	X'00000000'
MQQSIE_HIGH	1	X'00000001'
MQQSIE_OK	2	X'00000002'

MQQSO_* (SET, BROWSE, INPUT için Komut biçimi Kuyruk Durumu Açma Seçenekleri)

Çizelge 299. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQSO_NO	0	X'00000000'
MQQSO_YES	1	X'00000001'
MQQSO_SHARED	1	X'00000001'
MQQSO_EXCLUSIVE	2	X'00000002'

MQQSOT_* (Komut biçimi Kuyruk Durumu Açık Tipleri)

Çizelge 300. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQSOT_ALL	1	X'00000001'
MQQSOT_INPUT	2	X'00000002'
MQQSOT_OUTPUT	3	X'00000003'

MQQSUM_* (Komut biçimi Kuyruk Durumu Kesinleştirilmemiş İletiler)

Çizelge 301. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQQSUM_YES	1	X'00000001'
MQQSUM_NO	0	X'00000000'

MQQT_* (Kuyruk Tipleri ve Geniřletilmiř Kuyruk Tipleri)

Kuyruk Tipleri

Çizelge 302. Deęiřmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
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'

Geniřletilmiř Kuyruk Tipleri

Çizelge 303. Deęiřmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQQT_ALL	1001	X'000003E9'

MQRC_* (neden kodları)

Çizelge 304. Deęiřmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQRC_NONE	0	X'00000000'
MQRC_APPL_FIRST	900	X'00000384'
MQRC_APPL_LAST	999	X'000003E7'
MQRC_ALIAS_BASE_Q_TYPE_HATA	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_HATA	2006	X'000007D6'
MQRC_CHAR_ATTRS_HATA	2007	X'000007D7'
MQRC_CHAR_ATTRS_TOO_SHORT	2008	X'000007D8'
MQRC_CONNECTION_BROKEN	2009	X'000007D9'
MQRC_DATA_LENGTH_HATA	2010	X'000007DA'
MQRC_DYNAMIC_Q_AD_HATASI	2011	X'000007DB'
MQRC_ENVIRONMENT_ERROR	2012	X'000007DC'
MQRC_EXPIRY_HATA	2013	X'000007DD'
MQRC_FEEDBACK_HATA	2014	X'000007DE'
MQRC_GET_INHIBE	2016	X'000007E0'
MQRC_HANDLE_NOT_YOK	2017	X'000007E1'
MQRC_HCONN_HATA	2018	X'000007E2'
MQRC_HOBJ_HATA	2019	X'000007E3'
MQRC_INHIBIT_VALUE_ERROR	2020	X'000007E4'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_INT_ATTR_COUNT_HATA	2021	X'000007E5'
MQRC_INT_ATTR_COUNT_TOO_SMALL	2022	X'000007E6'
MQRC_INT_ATTRS_ARRAY_HATA	2023	X'000007E7'
MQRC_SYNCPOINT_LIMIT_ULAŞILDI	2024	X'000007E8'
MQRC_MAX_CONNS_LIMIT_SINIRINA ulaşıldı	2025	X'000007E9'
MQRC_MD_ERROR	2026	X'000007EA'
MQRC_MISSING_REPLY_TO_Q	2027	X'000007EB'
MQRC_MSG_TYPE_HATA	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_YETKILI	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'
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_HATA	2044	X'000007FC'
MQRC_OPTION_NOT_VALID_FOR_TYPE	2045	X'000007FD'
MQRC_OPTIONS_HATA	2046	X'000007FE'
MQRC_PERSISTENCE_HATA	2047	X'000007FF'
MQRC_PERSISTENT_NOT_ALLOWED	2048	X'00000800'
MQRC_PRIORITY_MAXIMUM değerini aşıyor	2049	X'00000801'
MQRC_PRIORITY_HATA	2050	X'00000802'
MQRC_PUT_INENGELLEME	2051	X'00000803'
MQRC_Q_SILINDI	2052	X'00000804'
MQRC_Q_FULL	2053	X'00000805'
MQRC_Q_NOT_EMPTY	2055	X'00000807'
MQRC_Q_SPACE_NOT_KULLANILAMIYOR	2056	X'00000808'
MQRC_Q_TYPE_HATA	2057	X'00000809'
MQRC_Q_MGR_AD_HATASI	2058	X'0000080A'
MQRC_Q_MGR_YOK	2059	X'0000080B'
MQRC_REPORT_OPTIONS_HATA	2061	X'0000080D'
MQRC_SECOND_MARK_NOT_ALLOWED	2062	X'0000080E'
MQRC_SECURITY_HATA	2063	X'0000080F'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_SELECTOR_COUNT_HATA	2065	X'00000811'
MQRC_SELECTOR_LIMIT_AŞILDI	2066	X'00000812'
MQRC_SELECTOR_ERROR	2067	X'00000813'
MQRC_SELECTOR_NOT_FOR_TYPE	2068	X'00000814'
MQRC_SIGNAL_HENÜZ tamamlanmamış	2069	X'00000815'
MQRC_SIGNAL_REQUEST_ACCEPTED	2070	X'00000816'
MQRC_STORAGE_KULLANILAMIYOR	2071	X'00000817'
MQRC_SYNCPOINT_NOT_YOK	2072	X'00000818'
MQRC_TRIGGER_CONTROL_HATA	2075	X'0000081B'
MQRC_TRIGGER_DEPTH_HATA	2076	X'0000081C'
MQRC_TRIGGER_MSG_PRIORITY_ERR	2077	X'0000081D'
MQRC_TRIGGER_TYPE_ERROR (QRC_TRIGGER_TYPE_HATA)	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'
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_HATA	2091	X'0000082B'
MQRC_XMIT_Q_USAGE_HATASI	2092	X'0000082C'
MQRC_NOT_OPEN_FOR_PASS_Tümü	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_HATA	2097	X'00000831'
MQRC_CONTEXT_NOT_VAR	2098	X'00000832'
MQRC_SIGNAL1_ERROR	2099	X'00000833'
MQRC_OBJECT_ALREADY_EXISTS	2100	X'00000834'
MQRC_OBJECT_HASARLI	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_HATA	2105	X'00000839'
MQRC_COD_NOT_VALID_FOR_XCF_Q	2106	X'0000083A'
MQRC_XWAIT_İPTAL edildi	2107	X'0000083B'
MQRC_XWAIT_HATASI	2108	X'0000083C'
MQRC_SUPPRESSED_BY_EXIT	2109	X'0000083D'
MQRC_FORMAT_HATASI	2110	X'0000083E'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_SOURCE_CCSDID_HATA	2111	X'0000083F'
MQRC_SOURCE_INTEGER_ENC_HATA	2112	X'00000840'
MQRC_SOURCE_DECIMAL_ENC_HATA	2113	X'00000841'
MQRC_SOURCE_FLOAT_ENC_HATA	2114	X'00000842'
MQRC_TARGET_CCSDID_HATA	2115	X'00000843'
MQRC_TARGET_INTEGER_ENC_HATA	2116	X'00000844'
MQRC_TARGET_DECIMAL_ENC_HATA	2117	X'00000845'
MQRC_TARGET_FLOAT_ENC_HATA	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_KATILIMCILAR	2121	X'00000849'
MQRC_PARTICIPANT_NOT_VAR	2122	X'0000084A'
MQRC_OUTCOME_MIXED	2123	X'0000084B'
MQRC_OUTCOME_PENDING	2124	X'0000084C'
MQRC_BRIDGE_BASLATILDI	2125	X'0000084D'
MQRC_BRIDGE_DURDURULDU	2126	X'0000084E'
MQRC_ADAPTER_STORAGE_SHORTAGE	2127	X'0000084F'
MQRC_UOW_IN_PROGRESS	2128	X'00000850'
MQRC_ADAPTER_CONN_LOAD_ERROR (UYGULAMA_HATASI)	2129	X'00000851'
MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)	2130	X'00000852'
MQRC_ADAPTER_DEFS_HATA	2131	X'00000853'
MQRC_ADAPTER_DEFS_LOAD_ERROR	2132	X'00000854'
MQRC_ADAPTER_CONV_LOAD_ERROR (UYGULAMA_HATASI)	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_HATA	2138	X'0000085A'
MQRC_CNO_HATA	2139	X'0000085B'
MQRC_CICS_WAIT_FAILED	2140	X'0000085C'
MQRC_DLH_HATA	2141	X'0000085D'
MQRC_HEADER_HATA	2142	X'0000085E'
MQRC_SOURCE_LENGTH_ERROR	2143	X'0000085F'
MQRC_TARGET_LENGTH_HATA	2144	X'00000860'
MQRC_SOURCE_BUFFER_ERROR (QRC_SOURCE_BUFFER_ERROR)	2145	X'00000861'
MQRC_TARGET_BUFFER_HATA	2146	X'00000862'
MQRC_IIH_HATA	2148	X'00000864'
MQRC_PCF_HATA	2149	X'00000865'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_DBCS_HATA	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 (RECORDS_HATASI)	2155	X'0000086B'
MQRC_RESPONSE_RECORDS_ERROR	2156	X'0000086C'
MQRC_ASID_MISMATCH	2157	X'0000086D'
MQRC_PMO_RECORD_FLAGS_HATA	2158	X'0000086E'
MQRC_PUT_MSG_RECORDS_HATA	2159	X'0000086F'
MQRC_CONN_ID_IN_USE	2160	X'00000870'
MQRC_Q_MGR QUIESCING	2161	X'00000871'
MQRC_Q_MGR DURDURULUYOR	2162	X'00000872'
MQRC_DUPLICATE_RECOV_COORD	2163	X'00000873'
MQRC_PMO_HATA	2173	X'0000087D'
MQRC_API_EXIT_NOT_BULUNAMADI	2182	X'00000886'
MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)	2183	X'00000887'
MQRC_REMOTE_Q_NAME_ERROR	2184	X'00000888'
MQRC_INPERSIST_PERSISTENCE	2185	X'00000889'
MQRC_GMO_HATA	2186	X'0000088A'
MQRC_CICS_BRIDGE_RESTRICTION	2187	X'0000088B'
MQRC_STOPPED_BY_CLUSTER_EXIT	2188	X'0000088C'
MQRC_CLUSTER_RESOLUTION_HATA	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_HATA	2193	X'00000891'
MQRC_NAME_NOT_VALID_FOR_TYPE	2194	X'00000892'
MQRC_UNEXPECTED_HATASI	2195	X'00000893'
MQRC_UNKNOWN_XMIT_Q	2196	X'00000894'
MQRC_UNKNOWN_DEF_XMIT_Q	2197	X'00000895'
MQRC_DEF_XMIT_Q_TYPE_HATA	2198	X'00000896'
MQRC_DEF_XMIT_Q_USAGE_HATA	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 DURDURULUYOR	2203	X'0000089B'
MQRC_ADAPTER_NOT_KULLANILAMIYOR	2204	X'0000089C'
MQRC_MSG_ID_HATA	2206	X'0000089E'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_CORREL_ID_HATA	2207	X'0000089F'
MQRC_FILE_SİSTEM_HATASI	2208	X'000008A0'
MQRC_NO_MSG_LOCKED	2209	X'000008A1'
MQRC_SOAP_DOTNET_HATA	2210	X'000008A2'
MQRC_SOAP_AXIS_HATA	2211	X'000008A3'
MQRC_SOAP_URL_HATA	2212	X'000008A4'
MQRC_FILE_NOT_AUDITED	2216	X'000008A8'
MQRC_CONNECTION_NOT_YETKILI	2217	X'000008A9'
MQRC_MSG_TOO_BIG_FOR_CHANNEL	2218	X'000008AA'
MQRC_CALL_IN_PROGRESS	2219	X'000008AB'
MQRC_RMH_ERROR (QRC_RMH_HATA)	2220	X'000008AC'
MQRC_Q_MGR_ETKİNLİŞTİRİCİ	2222	X'000008AE'
MQRC_Q_MGR_NOT_ETKİN	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'
MQRC_Q_SERVICE_INTERVAL_OK	2227	X'000008B3'
MQRC_RFH_HEADER_FIELD_HATASI	2228	X'000008B4'
MQRC_RAS_197_HATASI	2229	X'000008B5'
MQRC_UNIT_OF_WORK_NOT_BAŞLATILDI	2232	X'000008B8'
MQRC_CHANNEL_AUTO_DEF_OK	2233	X'000008B9'
MQRC_CHANNEL_AUTO_DEF_HATA	2234	X'000008BA'
MQRC_CFH_HATA	2235	X'000008BB'
MQRC_CFIL_HATA	2236	X'000008BC'
MQRC_CFIN_HATA	2237	X'000008BD'
MQRC_CFSL_HATA	2238	X'000008BE'
MQRC_CFST_HATA	2239	X'000008BF'
MQRC_INCOMPLETE_GROUP	2241	X'000008C1'
MQRC_INCOMPLETE_MSG	2242	X'000008C2'
MQRC_INCONSISTENT_CCSDID ' LER	2243	X'000008C3'
MQRC_INENCOT_ENCODINGS	2244	X'000008C4'
MQRC_TUTARSIZ_UOW	2245	X'000008C5'
MQRC_INVALID_MSG_UNDER_CURSOR	2246	X'000008C6'
MQRC_MATCH_OPTIONS_HATA	2247	X'000008C7'
MQRC_MDE_ERROR	2248	X'000008C8'
MQRC_MSG_FLAGS_HATA	2249	X'000008C9'
MQRC_MSG_SEQ_NUMBER_ERROR (QRC_MSG_SEQ_NUMBER_ERROR)	2250	X'000008CA'
MQRC_OFFSET_HATA	2251	X'000008CB'
MQRC_ORIGINAL_LENGTH_ERROR	2252	X'000008CC'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_SEGMENT_LENGTH_ZERO	2253	X'000008CD'
MQRC_UOW_YOK	2255	X'000008CF'
MQRC_ERROR_gmo_sürümü	2256	X'000008D0'
MQRC_ERROR_md_version	2257	X'000008D1'
MQRC_GROUP_ID_HATA	2258	X'000008D2'
MQRC_TUTARSIZ_GÖZAT	2259	X'000008D3'
MQRC_XQH_HATA	2260	X'000008D4'
MQRC_SRC_ENV_HATA	2261	X'000008D5'
MQRC_SRC_NAME_HATA	2262	X'000008D6'
MQRC_DEST_ENV_HATA	2263	X'000008D7'
MQRC_DEST_NAME_ERROR	2264	X'000008D8'
MQRC_TM_ERROR	2265	X'000008D9'
MQRC_CLUSTER_EXIT_ERROR (Çıkış)	2266	X'000008DA'
MQRC_CLUSTER_EXIT_LOAD_ERROR (Çıkış)	2267	X'000008DB'
MQRC_CLUSTER_PUT_INENGELLEME	2268	X'000008DC'
MQRC_CLUSTER_RESOURCE_ERROR	2269	X'000008DD'
MQRC_NO_DESTINATIONS_VAR	2270	X'000008DE'
MQRC_CONN_TAG_IN_USE	2271	X'000008DF'
MQRC_PARTIALLY_DÖNÜŞTÜRÜLDÜ	2272	X'000008E0'
MQRC_CONNECTION_HATA	2273	X'000008E1'
MQRC_OPTION_ENVIRONMENT_ERROR	2274	X'000008E2'
MQRC_CD_ERROR	2277	X'000008E5'
MQRC_CLIENT_CONN_ERROR (MQRC_CLIENT_HATASI)	2278	X'000008E6'
MQRC_CHANNEL_STOPD_BY_USER	2279	X'000008E7'
MQRC_HCONFIG_HATA	2280	X'000008E8'
MQRC_FUNCTION_HATA	2281	X'000008E9'
MQRC_CHANNEL_STARDI	2282	X'000008EA'
MQRC_CHANNEL_STOPPED	2283	X'000008EB'
MQRC_CHANNEL_CONV_HATA	2284	X'000008EC'
MQRC_SERVICE_KULLANILAMIYOR	2285	X'000008ED'
MQRC_INITIALIZATION_FAILED	2286	X'000008EE'
MQRC_TERMINATION_FAILED	2287	X'000008EF'
MQRC_UNKNOWN_Q_NAME	2288	X'000008F0'
MQRC_SERVICE_HATA	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'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_CHANNEL_ACTIVATED	2295	X'000008F7'
MQRC_CHANNEL_NOT_ACTIVATED	2296	X'000008F8'
MQRC_UOW_IPTAL edildi	2297	X'000008F9'
MQRC_FUNCTION_NOT_SUPPORTED	2298	X'000008FA'
MQRC_SELECTOR_TYPE_ERROR	2299	X'000008FB'
MQRC_COMMAND_TYPE_HATA	2300	X'000008FC'
MQRC_MULTIPLE_INSTANCE_ERROR	2301	X'000008FD'
MQRC_SYSTEM_ITEM_NOT_ALTERABLE	2302	X'000008FE'
MQRC_BAG_CONVERSION_HATA	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_HATA	2307	X'00000903'
MQRC_ENCODING_NOT_DESTEKLENMIYOR	2308	X'00000904'
MQRC_SELECTOR_NOT_PRESENT	2309	X'00000905'
MQRC_OUT_SELECTOR_HATA	2310	X'00000906'
MQRC_DIZGI_IFADESI kesildi	2311	X'00000907'
MQRC_SELECTOR_ERROR_type	2312	X'00000908'
MQRC_TUTARSIZ_ÖĞE_TIPI	2313	X'00000909'
MQRC_INDEX_HATA	2314	X'0000090A'
MQRC_SYSTEM_BAG_NOT_ALTERABLE	2315	X'0000090B'
MQRC_ITEM_COUNT_HATA	2316	X'0000090C'
MQRC_FORMAT_NOT_DESTEKLENMIYOR	2317	X'0000090D'
MQRC_SELECTOR_NOT_DESTEKLENMIYOR	2318	X'0000090E'
MQRC_ITEM_VALUE_ERROR	2319	X'0000090F'
MQRC_HBAG_HATA	2320	X'00000910'
MQRC_PARAMETER_MISSING	2321	X'00000911'
MQRC_CMD_SUNUCU_YOK	2322	X'00000912'
MQRC_STRING_LENGTH_HATA	2323	X'00000913'
MQRC_INQUIRY_COMMAND_ERROR	2324	X'00000914'
MQRC_NESTED_BAG_NOT_DESTEKLENMIYOR	2325	X'00000915'
MQRC_BAG_ERROR_type	2326	X'00000916'
MQRC_ITEM_TYPE_HATA	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 (QRC_CODED_CHAR_SET_ID_ERROR)	2330	X'0000091A'
MQRC_MSG_TOKEN_HATA	2331	X'0000091B'
MQRC_MISSING_WIH	2332	X'0000091C'
MQRC_WIH_ERROR	2333	X'0000091D'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_RFH_ERROR	2334	X'0000091E'
MQRC_RFH_STRING_HATASI	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_HATA	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_SERBEST BIRAKILMAMIŞ	2344	X'00000928'
MQRC_CF_NOT_YOK	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_HATA	2349	X'0000092D'
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_HATA	2354	X'00000932'
MQRC_UOW_MIX_NOT_DESTEKLENMIYOR	2355	X'00000933'
MQRC_WXP_HATA	2356	X'00000934'
MQRC_CURRENT_RECORD_ERROR	2357	X'00000935'
MQRC_NEXT_OFFSET_ERROR	2358	X'00000936'
MQRC_NO_RECORD_VAR	2359	X'00000937'
MQRC_OBJECT_LEVEL_UYUMSUZ	2360	X'00000938'
MQRC_NEXT_RECORD_ERROR	2361	X'00000939'
MQRC_BACKOUT_THRESHOLD_ULAŞILDI	2362	X'0000093A'
MQRC_MSG_NOT_MATCHED	2363	X'0000093B'
MQRC_JMS_FORMAT_HATA	2364	X'0000093C'
MQRC_SEGMENTS_NOT_DESTEKLENMIYOR	2365	X'0000093D'
MQRC_YANLIŞ_CF_DÜZEYI	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_HATA	2371	X'00000943'
MQRC_PARTICIPANT_NOT_DEFINED	2372	X'00000944'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_CF_STRUC_FAILED	2373	X'00000945'
MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)	2374	X'00000946'
MQRC_API_EXIT_INIT_ERROR (UYGULAMA_HATASI)	2375	X'00000947'
MQRC_API_EXIT_TERM_HATA	2376	X'00000948'
MQRC_EXIT_REASON_HATA	2377	X'00000949'
MQRC_RESERVED_VALUE_ERROR	2378	X'0000094A'
MQRC_NO_DATA_VAR	2379	X'0000094B'
MQRC_SCO_ERROR	2380	X'0000094C'
MQRC_KEY_REPOSITORY_HATA	2381	X'0000094D'
MQRC_CRYPTTO_HARDWARE_HATA	2382	X'0000094E'
MQRC_AUTH_INFO_REC_COUNT_HATA	2383	X'0000094F'
MQRC_AUTH_INFO_REC_HATA	2384	X'00000950'
MQRC_AIR_HATA	2385	X'00000951'
MQRC_AUTH_INFO_TYPE_ERROR	2386	X'00000952'
MQRC_AUTH_INFO_CONN_NAME_ERROR (KUYRUK_AD_HATASI)	2387	X'00000953'
MQRC_LDAP_USER_NAME_ERROR (QRC_LDAP_USER_NAME_ERROR)	2388	X'00000954'
MQRC_LDAP_USER_NAME_LENGTH_ERR	2389	X'00000955'
MQRC_LDAP_PASSWORD_HATASI	2390	X'00000956'
MQRC_SSL_ALREADY_KULLANIMA_HAZIRLANDI	2391	X'00000957'
MQRC_SSL_CONFIG_HATA	2392	X'00000958'
MQRC_SSL_INITIALIZION_HATA	2393	X'00000959'
MQRC_Q_INDEX_TYPE_HATA	2394	X'0000095A'
MQRC_CFBS_HATA	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 (QRC_SSL_PEER_NAME_ERROR)	2399	X'0000095F'
MQRC_UNSUPPORTED_CIPHER_SUITE	2400	X'00000960'
MQRC_SSL_CERTIFICATE_IPTAL edildi	2401	X'00000961'
MQRC_SSL_CERT_STORE_HATA	2402	X'00000962'
MQRC_CLIENT_EXIT_LOAD_ERROR (HATA_HATASI)	2406	X'00000966'
MQRC_CLIENT_EXIT_HATA	2407	X'00000967'
MQRC_UOW_KESINLEŞTİRİLDİ	2408	X'00000968'
MQRC_SSL_KEY_RESET_HATA	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_HATA	2414	X'0000096E'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_CFSF_HATA	2415	X'0000096F'
MQRC_CFGR_HATA	2416	X'00000970'
MQRC_MSG_NOT_ALLOWED_IN_GROUP	2417	X'00000971'
MQRC_FILTER_OPERATOR_ERROR	2418	X'00000972'
MQRC_NESTED_SELECTOR_HATA	2419	X'00000973'
MQRC_EPH_HATA	2420	X'00000974'
MQRC_RFH_FORMAT_HATA	2421	X'00000975'
MQRC_CFBF_HATA	2422	X'00000976'
MQRC_CLIENT_CHANNEL_CONFLICT	2423	X'00000977'
MQRC_SD_ERROR	2424	X'00000978'
MQRC_TOPIC_STRING_HATASI	2425	X'00000979'
MQRC_STS_HATA	2426	X'0000097A'
MQRC_NO_ABONELIK	2428	X'0000097C'
MQRC_SUBSCRIPTION_IN_USE	2429	X'0000097D'
MQRC_STAT_TYPE_ERROR	2430	X'0000097E'
MQRC_SUB_USER_DATA_ERROR (QRC_SUB_USER_DATA_HATA)	2431	X'0000097F'
MQRC_SUB_ALREADY_EXISTS	2432	X'00000980'
MQRC_IDENTITY_MISMATCH	2434	X'00000982'
MQRC_ALTER_SUB_HATA	2435	X'00000983'
MQRC_DURABILITY_NOT_ALLOWED	2436	X'00000984'
MQRC_NO_RETAINED_MSG	2437	X'00000985'
MQRC_SRO_HATA	2438	X'00000986'
MQRC_SUB_NAME_ERROR (MQRC_AD_HATASI)	2440	X'00000988'
MQRC_OBJECT_STRING_HATA	2441	X'00000989'
MQRC_NAMETY_NAME_ERROR	2442	X'0000098A'
MQRC_SEGMENTATION_ALLOWED	2443	X'0000098B'
MQRC_CBD_ERROR	2444	X'0000098C'
MQRC_CTLO_ERROR	2445	X'0000098D'
MQRC_NO_CALLBACKS_ETKINLEŞTIRICI	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_HATA	2460	X'0000099C'
MQRC_CMHO_HATA	2461	X'0000099D'
MQRC_DMHO_ERROR	2462	X'0000099E'
MQRC_SMPO_HATA	2463	X'0000099F'
MQRC_IMPO_HATA	2464	X'000009A0'
MQRC_COUNTY_NAME_TOO_BIG	2465	X'000009A1'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_PROP_VALUE_NOT_CONVERTED	2466	X'000009A2'
MQRC_PROP_TYPE_NOT_SUPPORTED	2467	X'000009A3'
MQRC_COUNTY_VALUE_TOO_BIG	2469	X'000009A5'
MQRC_PROP_CONV_NOT_SUPPORTED	2470	X'000009A6'
MQRC_PROPERTY_NOT_VAR	2471	X'000009A7'
MQRC_PROP_NUMBER_FORMAT_HATA	2472	X'000009A8'
MQRC_19TY_TYPE_HATA	2473	X'000009A9'
MQRC_COUNTIES_TOO_BIG	2478	X'000009AE'
MQRC_PUT_NOT_KORUNDU	2479	X'000009AF'
MQRC_ALIAS_TARGTYPE_CHANGED	2480	X'000009B0'
MQRC_DMPO_ERROR	2481	X'000009B1'
MQRC_PD_ERROR (QRC_PD_HATA)	2482	X'000009B2'
MQRC_CALLBACK_TYPE_HATA	2483	X'000009B3'
MQRC_CBD_OPTIONS_HATA	2484	X'000009B4'
MQRC_MAX_MSG_LENGTH_ERROR	2485	X'000009B5'
MQRC_CALLBACK_ROUTINE_HATA	2486	X'000009B6'
MQRC_CALLBACK_LINK_HATA	2487	X'000009B7'
MQRC_OPERATION_HATA	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_HATA	2501	X'000009C5'
MQRC_PUBLICATION_FAILURE	2502	X'000009C6'
MQRC_SUB_INENGELLEME	2503	X'000009C7'
MQRC_SELECTOR_ALWAYS_FALSE	2504	X'000009C8'
MQRC_XEPO_HATA	2507	X'000009CB'
MQRC_DURABILITY_NOT_ALTERABLE	2509	X'000009CD'
MQRC_TOPIC_NOT_DEĞİŞTİRİLEBİLİR	2510	X'000009CE'
MQRC_SUBLEVEL_NOT_ALTERABLE	2512	X'000009D0'
MQRC_NAME_LENGTH_ERR	2513	X'000009D1'
MQRC_DUPLICATE_GROUP_SUB	2514	X'000009D2'
MQRC_GROUPING_NOT_DEĞİŞTİRİLEBİLİR	2515	X'000009D3'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_HATA	2519	X'000009D7'
MQRC_RES_OBJECT_STRING_HATA	2520	X'000009D8'
MQRC_CONNECTION_ASKIYA ALINDI	2521	X'000009D9'
MQRC_INVALID_DESTINATION	2522	X'000009DA'
MQRC_GEÇERSİZ_ABONELİK	2523	X'000009DB'
MQRC_SELECTOR_NOT_DEĞİŞTİRİLEBİLİR	2524	X'000009DC'
MQRC_RETAINED_MSG_Q_ERROR	2525	X'000009DD'
MQRC_RETAINED_NOT_TESLİM edilemedi	2526	X'000009DE'
MQRC_RFH_RESTRICTED_FORMAT_ERR	2527	X'000009DF'
MQRC_CONNECTION_DURDURULDU	2528	X'000009E0'
MQRC_ASYNC_UOW_CONFLICT	2529	X'000009E1'
MQRC_ASYNC_XA_CONFLICT	2530	X'000009E2'
MQRC_PUBSUB_INHIBE	2531	X'000009E3'
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_HATA	2535	X'000009E7'
MQRC_CHANNEL_NOT_AVAILABLE	2537	X'000009E9'
MQRC_ANASİSTEM_YOK	2538	X'000009EA'
MQRC_CHANNEL_CONFIG_HATA	2539	X'000009EB'
MQRC_UNKNOWN_CHANNEL_NAME	2540	X'000009EC'
MQRC_LOOPING_YAYIN	2541	X'000009ED'
MQRC_ALREADY_JOINED	2542	X'000009EE'
MQRC_CHANNEL_SSL_WARNING	2552	X'000009F8'
MQRC_OCSP_URL_HATA	2553	X'000009F9'
MQRC_CIPHER_SPEC_NOT_SUITE_B	2591	X'00000A1F'
MQRC_SUITE_B_HATA	2592	X'00000A20'
MQRC_PASSWORD_PROTECTION_ERROR	2594	X'00000A22'
MQRC_REOPEN_EXCL_INPUT_HATA	6100	X'000017D4'
MQRC_REOPEN_INQUIRE_HATA	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_HATA	6106	X'000017DA'
MQRC_STRUC_HATA	6107	X'000017DB'

Çizelge 304. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRC_NULL_POINTER	6108	X'000017DC'
MQRC_NO_CONNECTION_REFERENCE	6109	X'000017DD'
MQRC_NO_BUFFER	6110	X'000017DE'
MQRC_BINARY_DATA_LENGTH_HATA	6111	X'000017DF'
MQRC_BUFFER_NOT_AUTOMATIC	6112	X'000017E0'
MQRC_INSUYETERINCE arabellek	6113	X'000017E1'
MQRC_VERIMSIZ_VERI	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_TUTARSIZ_BIÇIMI	6119	X'000017E7'
MQRC_TUTARSIZ_NESNE_DURUMU	6120	X'000017E8'
MQRC_CONTEXT_OBJECT_NOT_VALID	6121	X'000017E9'
MQRC_CONTEXT_OPEN_HATA	6122	X'000017EA'
MQRC_STRUC_LENGTH_HATA	6123	X'000017EB'
MQRC_NOT_CONNECTED	6124	X'000017EC'
MQRC_NOT_OPEN	6125	X'000017ED'
MQRC_DISTRIBUTION_LIST_EMPTY	6126	X'000017EE'
MQRC_INTUTARSIZ_OPEN_OPTIONS	6127	X'000017EF'
MQRC_YANLIŞ_SÜRÜMÜ	6128	X'000017F0'
MQRC_REFERENCE_HATA	6129	X'000017F1'

MQRCCF_* (Komut biçimi üstbilgisi neden kodları)

Programcının yanıtına ilişkin ek bilgi için PCF neden kodları başlıklı konuya bakın.

Çizelge 305. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQRCCF_CFH_TYPE_HATA	3001	X'00000BB9'
MQRCCF_CFH_LENGTH_HATA	3002	X'00000BBA'
MQRCCF_CFH_VERSION_HATA	3003	X'00000BBB'
MQRCCF_CFH_MSG_SEQ_NUMBER_ERR	3004	X'00000BBC'
MQRCCF_CFH_CONTROL_HATASI	3005	X'00000BBD'
MQRCCF_CFH_PARM_COUNT_HATA	3006	X'00000BBE'
MQRCCF_CFH_COMMAND_HATASI	3007	X'00000BBF'
MQRCCF_COMMAND_FAILED	3008	X'00000BC0'
MQRCCF_CFIN_LENGTH_HATA	3009	X'00000BC1'
MQRCCF_CFST_LENGTH_HATA	3010	X'00000BC2'
MQRCCF_CFST_DIZGI_UZUNLUĞU_ERR	3011	X'00000BC3'
MQRCCF_FORCE_VALUE_HATA	3012	X'00000BC4'

Çizelge 305. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRCCF_STRUCTURE_TYPE_ERROR	3013	X'00000BC5'
MQRCCF_CFIN_PARM_ID_HATA	3014	X'00000BC6'
MQRCCF_CFST_PARM_ID_HATA	3015	X'00000BC7'
MQRCCF_MSG_LENGTH_HATA	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_HATA	3022	X'00000BCE'
MQRCCF_MD_FORMAT_HATA	3023	X'00000BCF'
MQRCCF_CFSL_LENGTH_HATA	3024	X'00000BD0'
MQRCCF_REPLACE_VALUE_ERROR	3025	X'00000BD1'
MQRCCF_CFIL_DUPLICATE_VALUE	3026	X'00000BD2'
MQRCCF_CFIL_COUNT_HATA	3027	X'00000BD3'
MQRCCF_CFIL_LENGTH_HATA	3028	X'00000BD4'
MQRCCF QUIESCE_VALUE_ERROR	3029	X'00000BD5'
MQRCCF_MODE_VALUE_ERROR	3029	X'00000BD5'
MQRCCF_MSG_SEQ_NUMBER_ERROR (QRCCF_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_HATA	3033	X'00000BD9'
MQRCCF_CHANNEL_TYPE_HATA	3034	X'00000BDA'
MQRCCF_PARM_SEQUENCE_ERROR	3035	X'00000BDB'
MQRCCF_XMIT_PROTOCOL_TYPE_ERR	3036	X'00000BDC'
MQRCCF_BATCH_SIZE_HATA	3037	X'00000BDD'
MQRCCF_DISC_INT_HATA	3038	X'00000BDE'
MQRCCF_SHORT_RETRY_HATA	3039	X'00000BDF'
MQRCCF_SHORT_TIMER_HATA	3040	X'00000BE0'
MQRCCF_LONG_RETRY_HATA	3041	X'00000BE1'
MQRCCF_LONG_TIMER_HATA	3042	X'00000BE2'
MQRCCF_SEQ_NUMBER_WRAP_ERROR (QRCCF_SEQ_NUMBER_WRAP_HATA)	3043	X'00000BE3'
MQRCCF_MAX_MSG_LENGTH_HATA	3044	X'00000BE4'
MQRCCF_PUT_AUTH_ERROR	3045	X'00000BE5'
MQRCCF_PURGE_VALUE_ERROR	3046	X'00000BE6'
MQRCCF_CFIL_PARM_ID_HATA	3047	X'00000BE7'
MQRCCF_MSG_TRUNCATED	3048	X'00000BE8'
MQRCCF_CC SID_HATA	3049	X'00000BE9'

Çizelge 305. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRCCF_ENCODING_HATASI	3050	X'00000BEA'
MQRCCF_QUEUES_VALUE_HATA	3051	X'00000BEB'
MQRCCF_DATA_CONV_VALUE_ERROR	3052	X'00000BEC'
MQRCCF_INDOUBT_VALUE_HATA	3053	X'00000BED'
MQRCCF_ESCAPE_TYPE_HATA	3054	X'00000BEE'
MQRCCF_REPOS_VALUE_HATA	3055	X'00000BEF'
MQRCCF_CHANNEL_TABLE_HATA	3062	X'00000BF6'
MQRCCF_MCA_TYPE_HATA	3063	X'00000BF7'
MQRCCF_CHL_INST_TYPE_HATA	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_HATA	3068	X'00000BFC'
MQRCCF_CFSL_STRING_LENGTH_ERR	3069	X'00000BFD'
MQRCCF_BROKER_SILINDI	3070	X'00000BFE'
MQRCCF_STREAM_ERROR	3071	X'00000BFF'
MQRCCF_TOPIC_HATA	3072	X'00000C00'
MQRCCF_NOT_REGISTERED	3073	X'00000C01'
MQRCCF_Q_MGR_AD_HATASI	3074	X'00000C02'
MQRCCF_HATA_AKIS	3075	X'00000C03'
MQRCCF_Q_AD_HATASI	3076	X'00000C04'
MQRCCF_NO_RETAINED_MSG	3077	X'00000C05'
MQRCCF_DUPLICATE_IDENTITY	3078	X'00000C06'
MQRCCF_HATA_Q	3079	X'00000C07'
MQRCCF_CORREL_ID_HATA	3080	X'00000C08'
MQRCCF_NOT_YETKILI	3081	X'00000C09'
MQRCCF_UNKNOWN_STREAM	3082	X'00000C0A'
MQRCCF_REG_OPTIONS_ERROR	3083	X'00000C0B'
MQRCCF_PUB_OPTIONS_HATA	3084	X'00000C0C'
MQRCCF_UNKNOWN_BROKER	3085	X'00000C0D'
MQRCCF_Q_MGR_CCSID_HATA	3086	X'00000C0E'
MQRCCF_DEL_OPTIONS_HATA	3087	X'00000C0F'
MQRCCF_CLUSTER_NAME_CONFLICT	3088	X'00000C10'
MQRCCF_REPOS_NAME_CONFLICT	3089	X'00000C11'
MQRCCF_CLUSTER_Q_USAGE_HATA	3090	X'00000C12'
MQRCCF_ACTION_VALUE_HATA	3091	X'00000C13'
MQRCCF_COMMS_KITAPLIĞI_HATASI	3092	X'00000C14'
MQRCCF_NETBIOS_AD_HATASI	3093	X'00000C15'
MQRCCF_BROKER_COMMAND_FAILED	3094	X'00000C16'

Çizelge 305. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRCCF_CFST_ÇAKIŞAN_PARM	3095	X'00000C17'
MQRCCF_PATH_NOT_VALID	3096	X'00000C18'
MQRCCF_PARM_SYNTAX_ERROR	3097	X'00000C19'
MQRCCF_PWD_LENGTH_HATA	3098	X'00000C1A'
MQRCCF_FILTER_HATA	3150	X'00000C4E'
MQRCCF_YANLIŞ_KULLANICI	3151	X'00000C4F'
MQRCCF_DUPLICATE_SUBSCRIPTION	3152	X'00000C50'
MQRCCF_SUB_AD_HATASI	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_JOINED	3157	X'00000C55'
MQRCCF_OBJECT_IN_USE	3160	X'00000C58'
MQRCCF_UNKNOWN_DOSYASI_ADI	3161	X'00000C59'
MQRCCF_DOSYA_YOK	3162	X'00000C5A'
MQRCCF_DISC_RETRY_HATA	3163	X'00000C5B'
MQRCCF_ALLOC_RETRY_HATA	3164	X'00000C5C'
MQRCCF_ALLOC_SLOW_TIMER_HATA	3165	X'00000C5D'
MQRCCF_ALLOC_FAST_TIMER_HATA	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_ADI_HATASI	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_HATA	3174	X'00000C66'
MQRCCF_LOG_TYPE_HATA	3175	X'00000C67'
MQRCCF_PROGRAM_YOK	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_KOMUT_ENGELLEME	3204	X'00000C84'
MQRCCF_OBJECT_BEING_SILINDI	3205	X'00000C85'
MQRCCF_STORAGE_CLASS_IN_USE	3207	X'00000C87'
MQRCCF_OBJECT_NAME_RESTRICTED	3208	X'00000C88'
MQRCCF_OBJECT_LIMIT_AŞILDI	3209	X'00000C89'

Çizelge 305. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_ADİ HATASI	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_HATA	3225	X'00000C99'
MQRCCF_COMMAND_REPLY_HATA	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_HATA	3230	X'00000C9E'
MQRCCF_COMMAND_ORIGIN_ERROR	3231	X'00000C9F'
MQRCCF_LISTENER_CONFLICT	3232	X'00000CA0'
MQRCCF_LISTENER_BAŞLATILDI	3233	X'00000CA1'
MQRCCF_LISTENER_DURDURULDU	3234	X'00000CA2'
MQRCCF_CHANNEL_HATA	3235	X'00000CA3'
MQRCCF_CF_STRUC_HATA	3236	X'00000CA4'
MQRCCF_UNKNOWN_USER_ID	3237	X'00000CA5'
MQRCCF_UNEXPECTED_HATASI	3238	X'00000CA6'
MQRCCF_NO_XCF_PARTNER	3239	X'00000CA7'
MQRCCF_CFGR_PARM_ID_HATA	3240	X'00000CA8'
MQRCCF_C50_LENGTH_ERROR	3241	X'00000CA9'
MQRCCF_C50_OPERATOR_HATASI	3242	X'00000CAA'
MQRCCF_CFIF_PARM_ID_HATA	3243	X'00000CAB'
MQRCCF_CFSF_FILTER_VAL_LEN_ERR	3244	X'00000CAC'
MQRCCF_CFSF_LENGTH_HATA	3245	X'00000CAD'
MQRCCF_CFSF_OPERATOR_HATA	3246	X'00000CAE'
MQRCCF_CFSF_PARM_ID_HATA	3247	X'00000CAF'
MQRCCF_TOO_MANY_SÜZGEÇLERİ	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_DURDURULDU	3253	X'00000CB5'

Çizelge 305. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRCCF_CFBS_DUPLICATE_PARM	3254	X'00000CB6'
MQRCCF_CFBS_LENGTH_HATA	3255	X'00000CB7'
MQRCCF_CFBS_PARM_ID_HATA	3256	X'00000CB8'
MQRCCF_CFBS_DIZGI_UZUNLUĞU_ERR	3257	X'00000CB9'
MQRCCF_CFGR_LENGTH_HATA	3258	X'00000CBA'
MQRCCF_CFGR_PARM_COUNT_HATA	3259	X'00000CBB'
MQRCCF_CONN_NOT_DURDURULDU	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_HATA	3264	X'00000CC0'
MQRCCF_CFBF_PARM_ID_HATA	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'
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_HATA	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_AD_HATASI	3313	X'00000CF1'
MQRCCF_DURABILITY_IN_ALLOWED	3314	X'00000CF2'
MQRCCF_HOBJ_HATA	3315	X'00000CF3'
MQRCCF_DEST_NAME_ERROR	3316	X'00000CF4'
MQRCCF_GEÇERSİZ_HEDEF	3317	X'00000CF5'
MQRCCF_PUBSUB_INENGELLEME	3318	X'00000CF6'
MQRCCF_CHLAUTH_TYPE_ERROR	3326	X'00000CFE'
MQRCCF_CHLAUTH_ACTION_HATA	3327	X'00000CFF'
MQRCCF_CHLAUTH_USERSRC_HATA	3335	X'00000D07'
MQRCCF_ERROR_chlauth_type	3336	X'00000D08'
MQRCCF_CHLAUTH_ALREADY_EXISTS	3337	X'00000D09'
MQRCCF_CHLAUTH_BULUNAMADI	3338	X'00000D0A'

Çizelge 305. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRCCF_ERROR_chlauth_action	3339	X'00000D0B'
MQRCCF_HAKSIZ_CHLAUTH_USERSRC	3340	X'00000D0C'
MQRCCF_CHLAUTH_WARN_HATA	3341	X'00000D0D'
MQRCCF_ERROR_chlauth_match	3342	X'00000D0E'
MQRCCF_IPADDR_RANGE_CONFLICT	3343	X'00000D0F'
MQRCCF_CHLAUTH_MAX_AŞILDI	3344	X'00000D10'
MQRCCF_IPADDR_HATA	3345	X'00000D11'
MQRCCF_IPADDR_RANGE_HATA	3346	X'00000D12'
MQRCCF_PROFIL_ADI_EKSIK	3347	X'00000D13'
MQRCCF_CHLAUTH_CLNTUSER_HATA	3348	X'00000D14'
MQRCCF_CHLAUTH_AD_HATASI	3349	X'00000D15'
MQRCCF_SUITE_B_HATA	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_ENGELLENDI	3382	X'00000D36'
MQRCCF_OBJECT_ALREADY_EXISTS	4001	X'00000FA1'
MQRCCF_OBJECT_ERROR_type	4002	X'00000FA2'
MQRCCF_LIKE_OBJECT_ERROR_type	4003	X'00000FA3'
MQRCCF_OBJECT_OPEN	4004	X'00000FA4'
MQRCCF_ATTR_VALUE_HATA	4005	X'00000FA5'
MQRCCF_UNKNOWN_Q_MGR	4006	X'00000FA6'
MQRCCF_Q_HAKSIZ_TIPI	4007	X'00000FA7'
MQRCCF_OBJECT_NAME_ERROR (NESNE ADI HATASI)	4008	X'00000FA8'
MQRCCF_ALLOCATE_FAILED	4009	X'00000FA9'
MQRCCF_HOST_NOT_KULLANILAMAZ	4010	X'00000FAA'
MQRCCF_CONFIGURATION_ERROR	4011	X'00000FAB'
MQRCCF_CONNECTION_REDDDEDILDI	4012	X'00000FAC'
MQRCCF_ENTRY_HATA	4013	X'00000FAD'
MQRCCF_SEND_FAILED	4014	X'00000FAE'
MQRCCF_RECEIVED_DATA_ERROR	4015	X'00000FAF'
MQRCCF_RECEIVE_FAILED	4016	X'00000FB0'
MQRCCF_CONNECTION_CLOSED	4017	X'00000FB1'
MQRCCF_NO_STORAGE	4018	X'00000FB2'
MQRCCF_NO_COMMS_MANAGER	4019	X'00000FB3'
MQRCCF_LISTENER_NOT_BAŞLATILAMADI	4020	X'00000FB4'
MQRCCF_BIND_FAILED	4024	X'00000FB8'
MQRCCF_CHANNEL_INDOUBT	4025	X'00000FB9'
MQRCCF_MQCONN_FAILED	4026	X'00000FBA'

Çizelge 305. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRCCF_MQOPEN_FAILED	4027	X'00000FBB'
MQRCCF_MQGET_FAILED	4028	X'00000FBC'
MQRCCF_MQPUT_FAILED	4029	X'00000FBD'
MQRCCF_PING_HATASI	4030	X'00000FBE'
MQRCCF_CHANNEL_IN_USE	4031	X'00000FBF'
MQRCCF_CHANNEL_NOT_FOUND	4032	X'00000FC0'
MQRCCF_UNKNOWN_REMOTE_CHANNEL	4033	X'00000FC1'
MQRCCF_REMOTE_QM_UNAVAILABLE	4034	X'00000FC2'
MQRCCF_REMOTE_QM_TERMINATING	4035	X'00000FC3'
MQRCCF_MQINQ_FAILED	4036	X'00000FC4'
MQRCCF_NOT_XMIT_Q	4037	X'00000FC5'
MQRCCF_CHANNEL_DISABLED	4038	X'00000FC6'
MQRCCF_USER_EXIT_NOT_KULLANILAMIYOR	4039	X'00000FC7'
MQRCCF_COMMIT_FAILED	4040	X'00000FC8'
MQRCCF_ERROR_channel_type	4041	X'00000FC9'
MQRCCF_CHANNEL_ALREADY_EXISTS	4042	X'00000FCA'
MQRCCF_DATA_TOO_LARGE	4043	X'00000FCB'
MQRCCF_CHANNEL_NAME_HATA	4044	X'00000FCC'
MQRCCF_XMIT_Q_NAME_HATASI	4045	X'00000FCD'
MQRCCF_MCA_AD_HATASI	4047	X'00000FCF'
MQRCCF_SEND_EXIT_AD_HATASI	4048	X'00000FD0'
MQRCCF_SEC_EXIT_NAME_ERROR (MQRCCF_SEC_AD_HATASI)	4049	X'00000FD1'
MQRCCF_MSG_EXIT_AD_HATA	4050	X'00000FD2'
MQRCCF_RCV_EXIT_AD_HATASI	4051	X'00000FD3'
MQRCCF_XMIT_Q_NAME_ERROR_type	4052	X'00000FD4'
MQRCCF_MCA_NAME_ERROR_type	4053	X'00000FD5'
MQRCCF_DISC_INT_ERROR_type	4054	X'00000FD6'
MQRCCF_SHORT_RETRY_ERROR_type	4055	X'00000FD7'
MQRCCF_SHORT_TIMER_ERROR_type	4056	X'00000FD8'
MQRCCF_LONG_RETRY_ERROR_type	4057	X'00000FD9'
MQRCCF_LONG_TIMER_ERROR_type	4058	X'00000FDA'
MQRCCF_PUT_AUTH_ERROR_type	4059	X'00000FDB'
MQRCCF_KEEP_ALIVE_INT_HATA	4060	X'00000FDC'
MQRCCF_MISSING_CONN_NAME	4061	X'00000FDD'
MQRCCF_CONN_AD_HATASI	4062	X'00000FDE'
MQRCCF_MQSET_FAILED	4063	X'00000FDF'
MQRCCF_CHANNEL_NOT_ACTIVE	4064	X'00000FE0'
MQRCCF_TERMINATED_BY_SEC_EXIT	4065	X'00000FE1'
MQRCCF_DYNAMIC_Q_SCOPE_ERROR	4067	X'00000FE3'

Çizelge 305. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRCCF_CELL_DIR_NOT_KULLANILAMAZ	4068	X'00000FE4'
MQRCCF_MR_COUNT_ERROR (MQRCCF_COUNT_ERROR)	4069	X'00000FE5'
MQRCCF_MR_COUNT_ERROR tipi	4070	X'00000FE6'
MQRCCF_MR_EXIT_AD_HATASI	4071	X'00000FE7'
MQRCCF_MR_EXIT_NAME_ERROR_type	4072	X'00000FE8'
MQRCCF_MR_INTERVAL_HATA	4073	X'00000FE9'
MQRCCF_MR_INTERVAL_ERROR_type	4074	X'00000FEA'
MQRCCF_NPM_SPEED_HATASI	4075	X'00000FEB'
MQRCCF_NPM_SPEED_ERROR_type	4076	X'00000FEC'
MQRCCF_HB_INTERVAL_HATA	4077	X'00000FED'
MQRCCF_HB_INTERVAL_ERROR_type	4078	X'00000FEE'
MQRCCF_CHAD_HATA	4079	X'00000FEF'
MQRCCF_CHAD_ERROR_type	4080	X'00000FF0'
MQRCCF_CHAD_EVENT_HATA	4081	X'00000FF1'
MQRCCF_CHAD_EVENT_ERROR_type	4082	X'00000FF2'
MQRCCF_CHAD_EXIT_HATA	4083	X'00000FF3'
MQRCCF_CHAD_EXIT_ERROR_type	4084	X'00000FF4'
MQRCCF_SUPPRESSED_BY_EXIT	4085	X'00000FF5'
MQRCCF_BATCH_INT_HATA	4086	X'00000FF6'
MQRCCF_BATCH_INT_ERROR_type	4087	X'00000FF7'
MQRCCF_NET_PRIORITY_ERROR	4088	X'00000FF8'
MQRCCF_NET_PRIORITY_ERROR_type	4089	X'00000FF9'
MQRCCF_CHANNEL_CLOSED	4090	X'00000FFA'
MQRCCF_Q_STATUS_NOT_FOUND	4091	X'00000FFB'
MQRCCF_SSL_CIPHER_SPEC_ERROR (QRCCF_SSL_CIPHER_SPEC_HATA)	4092	X'00000FFC'
MQRCCF_SSL_PEER_NAME_ERROR (QRCCF_SSL_PEER_NAME_HATA)	4093	X'00000FFD'
MQRCCF_SSL_CLIENT_AUTH_ERROR (KUYRUK_HATASI)	4094	X'00000FFE'
MQRCCF_RETAINED_NOT_DESTEKLENMIYOR	4095	X'00000FFF'
 MQRCCF_KWD_VALUE_ERROR_type	4096	X'00001000'

MQRCN_* (İstemci yeniden bağlanma değişmezleri)

Çizelge 306. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQRCN_NO	0	X'00000000'
MQRCN_YES	1	X'00000001'
MQRCN_Q_MGR	2	X'00000002'
MQRCN_DISABLED	3	X'00000003'

MQRCVTIME_* (Alma Zamanasını Tipleri)

Çizelge 307. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRCVTIME_MULTIPLY	0	X'00000000'
MQRCVTIME_ADD	1	X'00000001'
MQRCVTIME_EQUAL	2	X'00000002'

MQREADA_* (Önünü Okuma Değerleri)

Çizelge 308. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQREADA_NO	0	X'00000000'
MQREADA_YES	1	X'00000001'
MQREADA_DISABLED	2	X'00000002'
MQREADA_ENGELLEME	3	X'00000003'
MQREADA_BACKLOG	4	X'00000004'

MQRECORDING_* (Kayıt Seçenekleri)

Çizelge 309. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRECORDING_DISABLED	0	X'00000000'
MQRECORDING_Q	1	X'00000001'
MQRECORDING_MSG	2	X'00000002'

MQREGO_* (Yayınlama/Abone Olma Kayıt Seçenekleri)

Çizelge 310. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_KORUNDU	256	X'00000100'
MQREGO_DUPLICATES_OK	512	X'00000200'
MQREGO_NON_PERSISTENT	1024	X'00000400'
MQREGO_KALICI	2048	X'00000800'
MQREGO_PERSISTENT_AS_PUBLISH	4096	X'00001000'
MQREGO_PERSISTENT_AS_Q	8192	X'00002000'

Çizelge 310. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQREGO_ADD_NAME	16384	X'00004000'
MQREGO_NO_ALTERATION	32768	X'00008000'
MQREGO_FULL_RESPONSE (mqrego_tam_yanıt)	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_* (Kurallar ve biçimlendirme üstbilgi yapısı ve İşaretleri)

Kurallar ve biçimlendirme üstbilgi yapısı

Çizelge 311. Sabitlerin yapıları

Ad	Yapı
MQRFH_STRUC_ID	"RFH↵"
MQRFH_STRUC_ID_ARRAY	'R','F','H','↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 312. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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'

Kurallar ve biçimlendirme üstbilgisi İşaretleri

Çizelge 313. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQRFH_NONE	0	X'00000000'
MQRFH_NO_FLAGS	0	X'00000000'

MQRFH2_* (Yayınla/Abone Ol Seçenekleri Etiketleri RFH2 Üst Düzey Klasör Etiketleri)

Çizelge 314. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQRFH2_NAME_VALUE_VERSION	1	X'00000001'

MQRFH2_* (Yayınla/Abone Ol Seçenekleri Etiket Adları)

MQRFH2_PUBSUB_CMD_FOLDER	"psc"
MQRFH2_PUBSUB_RESP_FOLDER	"pscr"

MQRFH2_MSG_CONTENT_FOLDER	"mcd"
MQRFH2_USER_FOLDER	"usr"

MQRFH2_* (Yayınla/Abone Ol Seçenekleri Etiketleri XML etiketleri adları)

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_* (Döndürülen Uzunluk)

Çizelge 315. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRL_UNDEFINED	-1	X'FFFFFFFF'

MQRMH_* (Başvuru iletisi üstbilgi yapısı)

Çizelge 316. Sabitlerin yapıları	
Ad	Yapı
MQRMH_STRUC_ID	"RMH↵"
MQRMH_STRUC_ID_ARRAY	'R', 'M', 'H', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 317. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRMH_VERSION_1	1	X'00000001'
MQRMH_CURRENT_VERSION	1	X'00000001'

MQRMHF_* (Başvuru iletisi üstbilgisi İşaretleri)

Çizelge 318. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRMHF_SON	1	X'00000001'
MQRMHF_NOT_LAST	0	X'00000000'

MQRO_* (Rapor Seçenekleri)

Çizelge 319. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRO_EXCEPTION	16777216	X'01000000'
MQRO_EXCEPTION_WITH_DATA	50331648	X'03000000'

Çizelge 319. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRO_EXCEPTION_WITH_FULL_DATA	117440512	X'07000000'
MQRO_SÜRE SONU	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'
MQRO_PASS_DISCARD_AND_EXPIRY	16384	X'00004000'
MQRO_NONE	0	X'00000000'

MQRO_* (Rapor Seçenekleri Maskeleri)

Çizelge 320. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQRO_REJECT_UNSUP_MASK	270270464	X'101C0000'
MQRO_ACCEPT_UNSUP_MASK	-270532353	X'EFE000FF'
MQRO_ACCEPT_UNSUP_IF_XMIT_MASK	261888	X'0003FF00'

MQROUTE_* (İzleme-yolu)**İzleme-Etkinlik Sayısı Üst Sınırı (MQIACF_MAX_ACTIVITIES)**

Çizelge 321. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQROUTE_UNLIMITED_ACTIVITIES	0	X'00000000'

İzleme Rotası Ayrıntısı (MQIACF_ROUTE_DETAIL)

Çizelge 322. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQROUTE_DETAIL_LOW	2	X'00000002'

Çizelge 322. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQRROUTE_DETAIL_MEDIUM	8	X'00000008'
MQRROUTE_DETAIL_HIGH	32	X'00000020'

İzleme rotası iletme (MQIACF_ROUTE_FORWARDING)

Çizelge 323. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRROUTE_FORWARD_ALL	256	X'00000100'
MQRROUTE_FORWARD_IF_DESTEKLI	512	X'00000200'
MQRROUTE_FORWARD_REJ_UNSUP_MASK	-65536	X'FFFF0000'

İzleme-rota Teslimi (MQIACF_ROUTE_DELIVERY)

Çizelge 324. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRROUTE_DELIVER_YES	4096	X'00001000'
MQRROUTE_DELIVER_NO	8192	X'00002000'
MQRROUTE_DELIVER_REJ_UNSUP_MASK	-65536	X'FFFF0000'

İzleme Rotası Toplama (MQIACF_ROUTE_ACCUMULATION)

Çizelge 325. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRROUTE_ACCUMULATE_NONE	65539	X'00010003'
MQRROUTE_ACCUMULATE_IN_MSG	65540	X'00010004'
MQRROUTE_ACCUMULATE_AND_REPLY	65541	X'00010005'

MQRP_* (Komut biçimi Değiştirme Seçenekleri)

Çizelge 326. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRP_YES	1	X'00000001'
MQRP_NO	0	X'00000000'

MQRQ_* (Komut biçimi Neden Niteleyicileri)

Çizelge 327. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRQ_CONN_NOT_YETKILI	1	X'00000001'
MQRQ_OPEN_NOT_AUTHORIZED	2	X'00000002'
MQRQ_CLOSE_NOT_YETKILI	3	X'00000003'
MQRQ_CMD_NOT_YETKILI	4	X'00000004'
MQRQ_Q_MGR_DURDURULUYOR	5	X'00000005'
MQRQ_Q_MGR_QUIESCING	6	X'00000006'

Çizelge 327. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQRQ_CHANNEL_STOPPED_OK	7	X'00000007'
MQRQ_CHANNEL_STOPPED_ERROR	8	X'00000008'
MQRQ_CHANNEL_STOPPED_RETRY	9	X'00000009'
MQRQ_CHANNEL_DURDURED_DISABLED	10	X'0000000A'
MQRQ_BRIDGE_STOPPED_OK	11	X'0000000B'
MQRQ_BRID_STOPPED_HATASI	12	X'0000000C'
MQRQ_SSL_HANDSHAKE_HATA	13	X'0000000D'
MQRQ_SSL_CIPHER_SPEC_ERROR (QRQ_SSL_CIPHER_SPEC_HATA)	14	X'0000000E'
MQRQ_SSL_CLIENT_AUTH_ERROR (QRQ_SSL_CLIENT_AUTH_ERROR)	15	X'0000000F'
MQRQ_SSL_PEER_NAME_ERROR (QRQ_SSL_PEER_NAME_HATA)	16	X'00000010'
MQRQ_SUB_NOT_YETKILI	17	X'00000011'
MQRQ_SUB_DEST_NOT_YETKILI	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_KANAL SAYISI	24	X'00000018'
MQRQ_MAX_KANAL SAYISI	25	X'00000019'
MQRQ_SVRCONN_INST_LIMIT	26	X'0000001A'
MQRQ_CLIENT_INST_LIMIT!	27	X'0000001B'
MQRQ_CAF_NOT_KURULU	28	X'0000001C'
MQRQ_CSP_NOT_YETKILI	29	X'0000001D'
MQRQ_FAILOVER_İZIN VERILDI	30	X'0000001E'
MQRQ_FAILOVER_NOT_ALLOWED	31	X'0000001F'
MQRQ_STANDBY_ETKINLEŞTİRİLDİ	32	X'00000020'
MQRQ_REPLICA_ETKINLEŞTİRİLDİ	33	X'00000021'

MQRT_* (Komut biçimi Yenileme Tipleri)

Çizelge 328. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQRT_CONFIGURATION	1	X'00000001'
MQRT_EXPIRY	2	X'00000002'
MQRT_NSPROC	3	X'00000003'
MQRT_PROXYSUB	4	X'00000004'

MQRU_* (Yalnızca İstek)

Çizelge 329. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQRU_PUBLISH_ON_REQUEST	1	X'00000001'
MQRU_PUBLISH_ALL	2	X'00000002'

MQSCA_* (TLS İstemci Kimlik Doğrulaması)

Çizelge 330. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSCA_REQUIRED	0	X'00000000'
MQSCA_OPTIONAL	1	X'00000001'

MQSCO_* (TLS yapılandırma seçenekleri)

TLS yapılandırma seçenekleri yapısı

Çizelge 331. Sabitlerin yapıları	
Ad	Yapı
MQSCO_STRUC_ID	"SCO~"
MQSCO_STRUC_ID_ARRAY	'S', 'C', 'O', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 332. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

TLS yapılandırma seçenekleri Anahtar Sıfırlama Sayısı

Çizelge 333. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSCO_RESET_COUNT_DEFAULT	0	X'00000000'

Komut Biçimi Kuyruk Tanımlaması Kapsamı

Çizelge 334. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSCO_Q_MGR	1	X'00000001'
MQSCO_CELL	2	X'00000002'

MQSCOPE_* (Yayınlama kapsamı)

Çizelge 335. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSCOPE_ALL	0	X'00000000'
MQSCOPE_AS_PARENT	1	X'00000001'
MQSCOPE_QMGR	4	X'00000004'

MQSCYC_* (Güvenlik Vakası)

Çizelge 336. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSCYC_UPPER	0	X'00000000'
MQSCYC_MIXED	1	X'00000001'

MQSD_* (Nesne tanımlayıcı yapısı)

Çizelge 337. Sabit adlar ve yapılar	
Ad	Yapı
MQSD_STRUC_ID	"SD--"
MQSD_STRUC_ID_ARRAY	'S','D','-', '-'

Not: - simgesi, tek bir boş karakteri temsil eder.

Çizelge 338. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSD_VERSION_1	1	X'00000001'
MQSD_CURRENT_VERSION	1	X'00000001'

MQSECITEM_* (Komut biçimi Güvenlik Öğeleri)

Çizelge 339. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSECITEM_Tümü	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_MQCMLS	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_* (Güvenlik İletişim Kuralı Tipleri)

Çizelge 340. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSECPROT_NONE	0	X'00000000'
MQSECPROT_SSLV30	1	X'00000001'
MQSECPROT_TL SV10	2	X'00000002'
MQSECPROT_TL SV12	4	X'00000004'

MQSECSW_* (Komut biçimi Güvenlik Anahtarları ve Anahtar Durumları)

Komut biçimi Güvenlik Anahtarları

Çizelge 341. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'
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'

Komut biçimi Güvenlik Anahtarı Durumları

Çizelge 342. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_HATA	25	X'00000019'
MQSECSW_ON_OVERRIDDEN	26	X'0000001A'

MQSECTYPE_* (Komut biçimi Güvenlik Tipleri)

Çizelge 343. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSECTYPE_AUTHSERV	1	X'00000001'
MQSECTYPE_SSL	2	X'00000002'

Çizelge 343. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQSECTYPE_CLASSES	3	X'00000003'

MQSEG_* (Bölümlleme)

Çizelge 344. Sabit adlar ve değerler	
Ad	Değer
MQSEG_ENGELLEME	'↵'
MQSEG_ALLOWED	'A'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

MQSEL_* (Özel Seçici Değerleri)

Çizelge 345. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Seçici Tipleri)

Çizelge 346. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSELTYPE_NONE	0	X'00000000'
MQSELTYPE_STANDARD	1	X'00000001'
MQSELTYPE_EXTENDED	2	X'00000002'

MQSID_* (Güvenlik Tanıtıcısı)

Çizelge 347. Sabit adlar ve değerler	
Ad	Değer
MQSID_YOK	X'00...00' (40 boş değer)
MQSID_NONE_ARRAY	'\0', '\0', ... (40 boş değer)

MQSIDT_* (Güvenlik Tanıtıcısı Tipleri)

Çizelge 348. Sabit adlar ve değerler	
Ad	Onaltılı değer
MQSIDT_NONE	X'00'
MQSIDT_NT_SECURITY_ID	X'01'
MQSIDT_WAS_SECURITY_ID	X'02'

MQSMPO_* (İleti özelliği seçeneklerini ve yapısını ayarla)

İleti özelliği seçenekleri yapısını ayarla

Çizelge 349. Sabitlerin yapıları	
Ad	Yapı
MQSMPO_STRUC_ID	"SMPO"
MQSMPO_STRUC_ID_ARRAY	'S', 'M', 'P', 'O'

Not: – simgesi, tek bir boş karakteri temsil eder.

Çizelge 350. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSMPO_VERSION_1	1	X'00000001'
MQSMPO_CURRENT_VERSION	1	X'00000001'

İleti Özelliği Seçeneklerini Belirle

Çizelge 351. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSMPO_SET_FIRST	0	X'00000000'
MQSMPO_SET_PROP_UNDER_CURSOR	1	X'00000001'
MQSMPO_SET_PROP_AFTER_CURSOR	2	X'00000002'
MQSMPO_APPEND_ÖZELLİĞİ	4	X'00000004'
MQSMPO_SET_PROP_BEFORE_CURSOR	8	X'00000008'
MQSMPO_NONE	0	X'00000000'

MQSO_* (Abone Olma Seçenekleri)

Çizelge 352. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSO_YOK	0	X'00000000'
MQSO_NON_DURABLE	0	X'00000000'
MQSO_READ_AHEAD_AS_Q_DEF	0	X'00000000'
MQSO_ALTER	1	X'00000001'
MQSO_OLUŞTURMA	2	X'00000002'
MQSO_RESUME	4	X'00000004'
MQSO_DURABLE	8	X'00000008'
MQSO_GROUP_SUB	16	X'00000010'
MQSO_YÖNETİLDİR	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'

Çizelge 352. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
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_ÖNDEN	134217728	X'08000000'
MQSO_READ_ÖNDEN	268435456	X'10000000'

MQSP_* (Eşitleme noktası Kullanılabilirliği)

Çizelge 353. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSP_KULLANILABILIR	1	X'00000001'
MQSP_NOT_VAR	0	X'00000000'

MQSPL_* (Güvenlik İlkesi Koruma Seçenekleri)

Çizelge 354. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSPL_PASSTHRU	0	X'00000000'
MQSPL_REMOVE	1	X'00000001'
MQSPL_AS_POLICY	2	X'00000002'

MQSQQM_* (Paylaşılan Kuyruk Yöneticisi Adı)

Çizelge 355. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSQQM_USE	0	X'00000000'
MQSQQM_IGNORE	1	X'00000001'

MQSR_* (İşlem)

Çizelge 356. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSR_ACTION_YAYIN	1	X'00000001'

MQSRO_* (Abonelik isteği seçenekleri yapısı)

Çizelge 357. Sabitlerin yapıları	
Ad	Yapı
MQSRO_STRUC_ID	"SRO~"
MQSRO_STRUC_ID_ARRAY	'S', 'R', 'O', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 358. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Bölüm Durumu)

Çizelge 359. Sabit adlar ve yapılar	
Ad	Yapı
MQSS_NOT_A_SEGMENT	'-'
MQSS_SEGMENT	'S'
MQSS_LAST_SEGMENT	'L'

Not: - simgesi, tek bir boş karakteri temsil eder.

MQSSL_* (TLS FIPS Gereksinimleri)

Not: AIX, Linux, and Windows işletim sistemlerinde IBM MQ, IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu [İşlem listesindeki NIST CMVP modüllerinde](#) aranarak görüntülenebilir.

Çizelge 360. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSSL_FIPS_NO	0	X'00000000'
MQSSL_FIPS_YES	1	X'00000001'

MQSTAT_* (Stat Seçenekleri)

Çizelge 361. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSTAT_TYPE_ASYNC_HATASI	0	X'00000000'
MQSTAT_TYPE_RECONNECTION	0	X'00000000'
MQSTAT_TYPE_RECONNECTION_HATA	0	X'00000000'

MQSTS_* (Durum raporlama yapısı yapısı)

Çizelge 362. Sabitlerin yapıları	
Ad	Yapı
MQSTS_STRUC_ID	"STAT"
MQSTS_STRUC_ID_ARRAY	'S', 'T', 'A', 'T'

Not: - simgesi, tek bir boş karakteri temsil eder.

Çizelge 363. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSTS_VERSION_1	1	X'00000001'
MQSTS_CURRENT_VERSION	1	X'00000001'

MQSUB_* (Durabilir abonelikler)

Sürekli izin verilen abonelikler

Çizelge 364. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSUB_DURABLE_AS_PARENT	0	X'00000000'
MQSUB_DURABLE_ALLOWED	1	X'00000001'
MQSUB_DURABLE_INENGELLEME	2	X'00000002'

Sürekli abonelik aralığı

Çizelge 365. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSUB_DURABLE_ALL	-1	X'FFFFFFFF'
MQSUB_DURABLE_YES	1	X'00000001'
MQSUB_DURABLE_NO	2	X'00000002'

MQSUBTYPE_* (Komut biçimi Abonelik Tipleri)

Çizelge 366. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Komut biçimi Askıya Alma Durumu)

Çizelge 367. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSUS_YES	1	X'00000001'
MQSUS_NO	0	X'00000000'

MQSVC_* (Hizmet)

Hizmet tipleri

Çizelge 368. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQSVC_TYPE_COMMAND	0	X'00000000'

Çizelge 368. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQSVC_TYPE_SERVER	1	X'00000001'

Hizmet Denetimleri

Çizelge 369. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQSVC_CONTROL_Q_MGR	0	X'00000000'
MQSVC_CONTROL_Q_MGR_START	1	X'00000001'
MQSVC_CONTROL_MANUAL	2	X'00000002'

Hizmet Durumu

Çizelge 370. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQSVC_STATUS_DURDURULDU	0	X'00000000'
MQSVC_STATUS_STARTING	1	X'00000001'
MQSVC_STATUS_RUNNING	2	X'00000002'
MQSVC_STATUS_DURDURULUYOR	3	X'00000003'
MQSVC_STATUS_RETRYING	4	X'00000004'

MQSYNCPPOINT_* (Pub/Sub geçişi için Komut biçimi Syncpoint değerleri)

Çizelge 371. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQSYNCPPOINT_YES	0	X'00000000'
MQSYNCPPOINT_IFPER	1	X'00000001'

MQSYSP_* (Komut biçimi Sistem Değiştirgesi Değerleri)

Çizelge 372. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQSYSP_NO	0	X'00000000'
MQSYSP_YES	1	X'00000001'
MQSYSP_GENİŞLETİLMİŞ	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'

Çizelge 372. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQSYSP_STATUS_PREMOUNT	31	X'0000001F'
MQSYSP_STATUS_VAR	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_* (Konu öznitelikleri)

jokerler

Çizelge 373. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQTA_BLOCK	1	X'00000001'
MQTA_PASSTHRU	2	X'00000002'

Aboneliklere İzin Verilir

Çizelge 374. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQTA_SUB_AS_PARENT	0	X'00000000'
MQTA_SUB_INENGELLEME	1	X'00000001'
MQTA_SUB_ALLOWED	2	X'00000002'

Yetkili Sunucu Alt Yayma

Çizelge 375. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQTA_PROXY_SUB_FORCE	1	X'00000001'
MQTA_PROXY_SUB_FIRSTUSE	2	X'00000002'

İzin Verilen Yayınlar

Çizelge 376. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQTA_PUB_AS_PARENT	0	X'00000000'
MQTA_PUB_INHIBE	1	X'00000001'
MQTA_PUB_ALLOWED	2	X'00000002'

MQTC_* (Tetikleyici Denetimleri)

Çizelge 377. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQTC_OFF	0	X'00000000'
MQTC_ON	1	X'00000001'

MQTCPKEEP_* (TCP canlı tutma)

Çizelge 378. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTCPKEEP_NO	0	X'00000000'
MQTCPKEEP_YES	1	X'00000001'

MQTCPSTACK_* (TCP Yiğini Tipleri)

Çizelge 379. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTCPSTACK_SINGLE	0	X'00000000'
MQTCPSTACK_MULTIPLE	1	X'00000001'

MQTIME_* (Komut biçimi Saat birimleri)

Çizelge 380. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTIME_UNIT_MINS	0	X'00000000'
MQTIME_UNIT_SECS	1	X'00000001'

MQTM_* (Tetikleyici ileti yapısı)

Çizelge 381. Sabitlerin yapıları	
Ad	Yapı
MQTM_STRUC_ID	"TM↯"
MQTM_STRUC_ID_ARRAY	'T', 'M', '↯', '↯'

Not: ↯ simgesi, tek bir boş karakteri temsil eder.

Çizelge 382. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTM_VERSION_1	1	X'00000001'
MQTM_CURRENT_VERSION	1	X'00000001'

MQTM_C_* (Tetikleyici ileti karakter biçimi yapısı)

Çizelge 383. Sabitlerin yapıları	
Ad	Yapı
MQTM_C_STRUC_ID	"TMC↯"
MQTM_C_STRUC_ID_ARRAY	'T', 'M', 'C', '↯'
MQTM_C_VERSION_1	"↯↯1"
MQTM_C_VERSION_2	"↯↯2"
MQTM_C_CURRENT_VERSION	"↯↯2"
MQTM_C_VERSION_1_ARRAY	'↯', '↯', '↯', '1'
MQTM_C_VERSION_2_ARRAY	'↯', '↯', '↯', '2'
MQTM_C_CURRENT_VERSION_ARRAY	'↯', '↯', '↯', '2'

MQTOPT_* (Konu Tipi)

Çizelge 384. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTOPT_LOCAL	0	X'00000000'
MQTOPT_CLUSTER	1	X'00000001'
MQTOPT_ALL	2	X'00000002'

MQTRAXSTR_* (Kanal Başlatıcı İzlemesi Otomatik Başlatma)

Çizelge 385. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTRAXSTR_NO	0	X'00000000'
MQTRAXSTR_YES	1	X'00000001'

MQTSOPE_* (Abonelik Kapsamı)

Çizelge 386. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTSOPE_QMGR	1	X'00000001'
MQTSOPE_ALL	2	X'00000002'

MQTT_* (Tetikleyici Tipleri)

Çizelge 387. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTT_NONE	0	X'00000000'
MQTT_FIRST	1	X'00000001'
MQTT_HER	2	X'00000002'
MQTT_DERİNLİĞİ	3	X'00000003'

MQTYPE_* (Özellik veri tipleri)

Çizelge 388. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQTYPE_AS_SET	0	X'00000000'
MQTYPE_NULL (boş değerli)	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'

Çizelge 388. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQTYPE_STRING	1024	X'00000400'

MQUA_* (Yayınlama/Abone Olma Kullanıcı Özniteliği Seçicileri)

Çizelge 389. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQUA_FIRST	65536	X'00010000'
SON	999999999	X'3B9AC9FF'

MQUIDSUPP_* (Komut biçimi Kullanıcı Kimliği Desteği)

Çizelge 390. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQUIDSUPP_NO	0	X'00000000'
MQUIDSUPP_YES	1	X'00000001'

MQUNDELIVERED_* (Pub/Sub geçişi için Komut biçimi Teslim Edilmemiş değerler)

Çizelge 391. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQUNDELIVERED_NORMAL	0	X'00000000'
MQUNDELIVERED_SAFE	1	X'00000001'
MQUNDELIVERED_DISCARD	2	X'00000002'
MQUNDELIVERED_TUT	3	X'00000003'

MQUOWST_* (Komut biçimi UOW Durumları)

Çizelge 392. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQUOWST_NONE	0	X'00000000'
MQUOWST_ACTIVE	1	X'00000001'
MQUOWST_HAZIRLANDI	2	X'00000002'
MQUOWST_ÇÖZÜMLENMEDI	3	X'00000003'

MQUOWT_* (Komut biçimi UOW Tipleri)

Çizelge 393. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_* (Kuyruk Usajları)

Çizelge 394. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQUS_NORMAL	0	X'00000000'
MQUS_ILETIMI	1	X'00000001'

MQUSAGE_* (Komut biçimi Sayfa Kümesi Kullanım Değerleri ve Veri Kümesi Kullanım Değerleri)

Komut biçimi Sayfa Kümesi Kullanım Değerleri

Çizelge 395. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQUSAGE_PS_, KULLANILABILIR	0	X'00000000'
MQUSAGE_PS_DEFINED	1	X'00000001'
MQUSAGE_PS_ÇEVIRIMDIŞI	2	X'00000002'
MQUSAGE_PS_NOT_DEFINED	3	X'00000003'
MQUSAGE_PS_ASKIYA_ALINDI	4	X'00000004'
MQUSAGE_EXPAND_USER	1	X'00000001'
MQUSAGE_EXPAND_SYSTEM	2	X'00000002'
MQUSAGE_EXPAND_NONE	3	X'00000003'

Komut biçimi Veri Kümesi Kullanım Değerleri

Çizelge 396. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Değer Uzunluğu)

Çizelge 397. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQVL_NULL_TERMINATED	-1	X'FFFFFFFF'
MQVL_EMPTY_STRING	0	X'00000000'

MQVU_* (Değişken Kullanıcı Kimliği)

Çizelge 398. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQVU_FIXED_USER	1	X'00000001'
MQVU_ANY_USER	2	X'00000002'

MQWDR_* (Küme iş yükü çıkış hedefi kayıt yapısı)

Çizelge 399. Sabitlerin yapıları	
Ad	Yapı
MQWDR_STRUC_ID	"WDR~"
MQWDR_STRUC_ID_ARRAY	'W', 'D', 'R', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 400. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Bekleme Aralığı)

Çizelge 401. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQWI_SINIRSIZ	-1	X'FFFFFFFF'

MQWIH_* (İş yükü bilgileri üstbilgi yapısı ve işaretleri)

İş yükü bilgileri üstbilgi yapısı

Çizelge 402. Sabitlerin yapıları	
Ad	Yapı
MQWIH_STRUC_ID	"WIH~"
MQWIH_STRUC_ID_ARRAY	'W', 'I', 'H', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 403. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQWIH_VERSION_1	1	X'00000001'
MQWIH_CURRENT_VERSION	1	X'00000001'
MQWIH_LENGTH_1	120	X'00000078'
MQWIH_CURRENT_LENGTH	120	X'00000078'

İş yükü bilgileri üstbilgisi İşaretleri

Çizelge 404. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQWIH_NONE	0	X'00000000'

MQWQR_* (Küme iş yükü çıkış kuyruğu kayıt yapısı)

Çizelge 405. Sabitlerin yapıları	
Ad	Yapı
MQWQR_STRUC_ID	"WQR↵"
MQWQR_STRUC_ID_ARRAY	'W', 'Q', 'R', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 406. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'
MQWQR_LENGTH_2	208	X'000000D0'
MQWQR_LENGTH_3	212	X'000000D4'
MQWQR_CURRENT_LENGTH	212	X'000000D4'

MQWS_* (Genel Arama Kartı Şeması)

Çizelge 407. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQWS_DEFAULT	0	X'00000000'
MQWS_CHAR	1	X'00000001'
MQWS_TOPIC	2	X'00000002'

MQWXP_* (Küme iş yükü çıkış parametresi yapısı)

MQWXP_* (Küme iş yükü çıkış parametresi yapısı)

Çizelge 408. Sabitlerin yapıları	
Ad	Yapı
MQWXP_STRUC_ID	"WXP↵"
MQWXP_STRUC_ID_ARRAY	'W', 'X', 'P', '↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 409. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Küme İş Yüğü İşaretleri)

Çizelge 410. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQWXP_PUT_BY_CLUSTER_CHL	2	X'00000002'

İlgili başvurular

“MQWXP ' deki Alanlar-Küme iş yüğü çıkış parametresi yapısı” sayfa 1516

MQWXP -Küme iş yüğü çıkış parametresi yapısındaki alanların açıklaması

MQXACT_* (API çağırın Tipleri)

Çizelge 411. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQXACT_EXTERNAL	1	X'00000001'
MQXACT_INTERNAL	2	X'00000002'

MQXC_* (Çıkış Komutları)

Çizelge 412. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Çıkış Yanıtları)

Çizelge 413. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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_* (Çıkış Yanıtı)

Çizelge 414. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQXDR_OK	0	X'00000000'
MQXDR_CONVERSION_FAILED	1	X'00000001'

MQXE_* (Ortamlar)

Çizelge 415. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQXE_DİĞER	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'

MQXEPO_* (Kayıt Giriş Noktası Seçenekleri yapısı ve Çıkış Seçenekleri)

Kayıt Giriş Noktası Seçenekleri yapısı

Çizelge 416. Sabitlerin yapıları	
Ad	Yapı
MQXEPO_STRUC_ID	"XEPO"
MQXEPO_STRUC_ID_ARRAY	'X', 'E', 'P', 'O'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 417. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQXEPO_VERSION_1	1	X'00000001'
MQXEPO_CURRENT_VERSION	1	X'00000001'

Çıkış Seçenekleri

Çizelge 418. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQXEPO_NONE	0	X'00000000'

MQXF_* (API İşlev Tanıtıcıları)

Çizelge 419. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

Çizelge 419. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
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_GERI	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'

MQXP_* (API, çıkış değiştirgesi yapısını geçiyor)

Çizelge 420. Sabitlerin yapıları

Ad	Yapı
MQXP_STRUC_ID	"XP~"
MQXP_STRUC_ID_ARRAY	'X', 'P', '~', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 421. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQXP_VERSION_1	1	X'00000001'

MQXPDA_* (Sorun Belirleme Alanı)

Çizelge 422. Sabit adlar ve değerler	
Ad	Değer
MQXPDA_NONE	X'00...00' (48 boş değer)
MQXPDA_NONE_ARRAY	'\0', '\0', ... (48 boş değer)

MQXPT_* (İletim Tipleri)

Çizelge 423. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQXPT_Tümü	-1	X'FFFFFFF'
MQXPT_LOCAL	0	X'0000000'
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_* (İletim kuyruğu üstbilgi yapısı)

Çizelge 424. Sabitlerin yapıları	
Ad	Yapı
MQXQH_STRUC_ID	"XQH↯"
MQXQH_STRUC_ID_ARRAY	'X', 'Q', 'H', '↯'

Not: ↯ simgesi, tek bir boş karakteri temsil eder.

Çizelge 425. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQXQH_VERSION_1	1	X'00000001'
MQXQH_CURRENT_VERSION	1	X'00000001'

MQXR_* (Çıkış Nedenleri)

Çizelge 426. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
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'

Çizelge 426. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQXR_RETRY	17	X'00000011'
MQXR_AUTO_CLUSSDR	18	X'00000012'
MQXR_AUTO_ALICI	19	X'00000013'
MQXR_CLWL_OPEN	20	X'00000014'
MQXR_CLWL_PUT	21	X'00000015'
MQXR_CLWL_MOVE	22	X'00000016'
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_* (Çıkış Yanıtı 2)

Çizelge 427. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_* (Çıkış Tanıtıcıları)

Çizelge 428. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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'

Çizelge 428. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQXT_PUBSUB_ROUTING_EXIT	21	X'00000015'

MQXUA_* (Kullanıcı Alanı Değerinden Çık)

Çizelge 429. Sabit adlar ve değerler	
Ad	Değer
MQXUA_NONE	X'00...00' (16 boş değer)
MQXUA_NONE_ARRAY	'\0', '\0', ... (16 boş değer)

MQXWD_* (Bekleme tanımlayıcısı yapısından çık)

Çizelge 430. Sabitlerin yapıları	
Ad	Yapı
MQXWD_STRUC_ID	"XWD~"
MQXWD_STRUC_ID_ARRAY	'X', 'W', 'D', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 431. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQXWD_VERSION_1	1	X'00000001'

MQZAC_* (Uygulama bağlamı yapısı)

Çizelge 432. Sabitlerin yapıları	
Ad	Yapı
MQZAC_STRUC_ID	"ZAC~"
MQZAC_STRUC_ID_ARRAY	'Z', 'A', 'C', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 433. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQZAC_VERSION_1	1	X'00000001'
MQZAC_CURRENT_VERSION	1	X'00000001'

MQZAD_* (Yetki veri yapısı)

Çizelge 434. Sabitlerin yapıları	
Ad	Yapı
MQZAD_STRUC_ID	"ZAD~"
MQZAD_STRUC_ID_ARRAY	'Z', 'A', 'D', '~'

Not: ~ simgesi, tek bir boş karakteri temsil eder.

Çizelge 435. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQZAD_VERSION_1	1	X'00000001'
MQZAD_VERSION_2	2	X'00000002'
MQZAD_CURRENT_VERSION	2	X'00000002'

MQZAET_* (Kurulabilir Hizmetler Varlık Tipleri)

Çizelge 436. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQZAET_NONE	0	X'00000000'
MQZAET_BIRINCIL_KULLANICISI	1	X'00000001'
MQZAET_GRUP	2	X'00000002'
MQZAET_BILINMIYOR	3	X'00000003'

MQZAO_* (Kurulabilir Hizmet Yetkileri)

Çizelge 437. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQZAO_CONNECT	1	X'00000001'
MQZAO_GÖZ AT	2	X'00000002'
MQZAO_INPUT	4	X'00000004'
MQZAO_OUTPUT	8	X'00000008'
MQZAO_INQUIRE	16	X'00000010'
MQZAO_KÜMESİ	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_ABONE OLUŞTU	4096	X'00001000'
MQZAO_RESUME	8192	X'00002000'
MQZAO_ALL_MQI	16383	X'00003FFF'
MQZAO_OLUŞTURMA	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_AUTHORIZE	8388608	X'00800000'
MQZAO_ALL_ADMIN	16646144	X'00FE0000'
MQZAO_Tümü	16662527	X'00FE3FFF'

Çizelge 437. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQZAO_REMOVE	16777216	X'01000000'
MQZAO_NONE	0	X'00000000'

MQZAS_* (Kurulabilir Hizmetler Hizmet Arabirimi Sürümü)

Çizelge 438. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
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_* (Kimlik Doğrulama Tipleri)

Çizelge 439. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQZAT_INITIAL_CONTEXT	0	X'00000000'
MQZAT_CHANGE_CONTEXT	1	X'00000001'

MQZCI_* (Kurulabilir Hizmetler Devam Göstergesi)

Çizelge 440. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQZCI_DEFAULT	0	X'00000000'
MQZCI_CONTINUE	0	X'00000000'
MQZCI_DURUŞU	1	X'00000001'

MQZED_* (Varlık veri yapısı)

Çizelge 441. Sabitlerin yapıları

Ad	Yapı
MQZED_STRUC_ID	"ZED↵"
MQZED_STRUC_ID_ARRAY	'Z','E','D','↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 442. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQZED_VERSION_1	1	X'00000001'
MQZED_VERSION_2	2	X'00000002'
MQZED_CURRENT_VERSION	2	X'00000002'

MQZFP_* (Serbest deęiřtirge yapısı)

Çizelge 443. Sabitlerin yapıları	
Ad	Yapı
MQZFP_STRUC_ID	"ZFP↵"
MQZFP_STRUC_ID_ARRAY	'Z','F','P','↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 444. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQZFP_VERSION_1	1	X'00000001'
MQZFP_CURRENT_VERSION	1	X'00000001'

MQZIC_* (Kimlik baęlamı yapısı)

Çizelge 445. Sabitlerin yapıları	
Ad	Yapı
MQZIC_STRUC_ID	"ZIC↵"
MQZIC_STRUC_ID_ARRAY	'Z','I','C','↵'

Not: ↵ simgesi, tek bir boş karakteri temsil eder.

Çizelge 446. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQZIC_VERSION_1	1	X'00000001'
MQZIC_CURRENT_VERSION	1	X'00000001'

MQZID_* (Hizmetler için işlev tanıtıcıları)

Tüm hizmetler için ortak işlev tanıtıcıları

Çizelge 447. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
MQZID_INIT	0	X'00000000'
MQZID_TERM	1	X'00000001'

Yetki hizmeti için işlev tanıtıcıları

Çizelge 448. Deęişmezlerin deęerleri		
Ad	Ondalık deęer	Onaltılı deęer
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'

Çizelge 448. Değişmezlerin değerleri (devamı var)

Ad	Ondalık değer	Onaltılı değer
MQZID_GET_BELIRIT_YETKISI	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'
MQZID_CHECK_PRIVILEGED	13	X'0000000D'

Ad hizmeti için işlev tanıtıcıları

Çizelge 449. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQZID_INIT_ADı	0	X'00000000'
MQZID_TERM_ADı	1	X'00000001'
MQZID_LOOKUP_ADı	2	X'00000002'
MQZID_INSERT_NAME	3	X'00000003'
MQZID_DELETE_NAME	4	X'00000004'

Kullanıcı kimliği hizmeti için işlev tanıtıcıları

Çizelge 450. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQZID_INIT_USERID	0	X'00000000'
MQZID_TERM_USERID	1	X'00000001'
MQZID_FIND_USERID	2	X'00000002'

MQZIO_* (Kurulabilir Hizmetler Başlatma Seçenekleri)

Çizelge 451. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQZIO_PRIMARY	0	X'00000000'
MQZIO_IKINCIL	1	X'00000001'

MQZNS_* (Ad Hizmeti Arabirimi Sürümü)

Çizelge 452. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQZNS_VERSION_1	1	X'00000001'

MQZSE_* (Kurulabilir Hizmetler Başlatma-Sıralama Göstergesi)

Çizelge 453. Değişmezlerin değerleri

Ad	Ondalık değer	Onaltılı değer
MQZSE_START	1	X'00000001'

Çizelge 453. Değişmezlerin değerleri (devamı var)		
Ad	Ondalık değer	Onaltılı değer
MQZSE_CONTINUE	0	X'00000000'

MQZSL_* (Kurulabilir Hizmetler Seçici Göstergesi)

Çizelge 454. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQZSL_NOT_DÖNDÜRÜLMEDI	0	X'00000000'
MQZSL_DÖNDÜRÜLDÜ	1	X'00000001'

MQZTO_* (Kurulabilir Hizmetler Sonlandırma Seçenekleri)

Çizelge 455. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQZTO_BİRİNCİL	0	X'00000000'
MQZTO_IKİNCİL	1	X'00000001'

MQZUS_* (Kullanıcı Kimliği Hizmet Arabirimi Sürümü)

Çizelge 456. Değişmezlerin değerleri		
Ad	Ondalık değer	Onaltılı değer
MQZUS_VERSION_1	1	X'00000001'

MQI 'da kullanılan veri tipleri

İleti Kuyruğu Arabirimi 'nde (MQI) kullanılacak veri tiplerine ilişkin bilgiler. Her veri tipi ile ilgili diller için tanımlar, alanlar ve dil bildirimleri.

MQI için veri tipleri ve programlama

Temel ve Yapı veri tiplerinin tanıtılması ve C programlama, COBOL programlama ya da High Level Assembler programlaması aracılığıyla MQI 'ın nasıl kullanılacağı.

Temel veri tipleri

MQI 'da ya da çıkış işlevlerinde kullanılan veri tipleriyle ilgili bilgi. Bunlar, desteklenen programlama dillerinde temel veri türlerinin nasıl bildirileceğini gösteren örneklerle ayrıntılı olarak açıklanır.

MQI 'da ya da çıkış işlevlerinde kullanılan veri tipleri şunlardır:

- Temel veri tipleri ya da
- Temel veri türlerinin (diziler veya yapılar) toplamları

MQI 'da ya da çıkış işlevlerinde kullanılan temel veri tipleri şunlardır:

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları		
Temel veri tipi adı	Veri tipi	Açıklama
MQBOOL	Boole	MQBOOL veri tipi bir Boole değerini gösterir. 0 değeri false değerini gösterir. Diğer herhangi bir değer true değerini temsil eder. Bir MQBOOL, MQLONG veri tipine göre hizalanmalıdır.

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
MQBYTE	Byte	<p>MQBYTE veri tipi, tek byte veri tipini gösterir. Bayta belirli bir yorum konmaz; ikili sayı ya da karakter olarak değil, bir bit dizgisi olarak işlenir. Özel hizalama gerekmez.</p> <p>Farklı karakter kümeleri ya da kodlamalar kullanan kuyruk yöneticileri arasında MQBYTE verileri gönderildiğinde, MQBYTE verileri hiçbir şekilde dönüştürülmez. MQMD yapısındaki <i>MsgId</i> ve <i>CorrelId</i> alanları şöyle.</p> <p>Kuyruk yöneticisinin bilmediği bir ana saklama alanını göstermek için bazen MQBYTE dizisi kullanılır. Örneğin, alan uygulama iletili verilerini ya da bir yapıyı içerebilir. Bu alanın sınır hizalaması, içindeki verilerin niteliğiyle uyumlu olmalıdır.</p> <p>C programlama dilinde, herhangi bir veri tipi, MQBYTE dizileri olarak gösterilen işlev parametreleri için kullanılabilir. Bunun nedeni, bu tür parametrelerin her zaman adres tarafından iletilmesi ve C ' de işlev parametresinin işaretçi-boş olarak bildirilmesi olabilir.</p>
MQBYTEn	n bayt dizisi	<p>Her MQBYTEn veri tipi n byte 'lık bir dizgiyi temsil eder; burada n şu değerlerden herhangi birini alabilir: 8, 16, 24, 32, 40 ya da 128. Her byte, MQBYTE veri tipi tarafından tanımlanır. Özel hizalama gerekmez.</p> <p>Bayt dizgisindeki veriler, dizginin tanımlı uzunluğundan kısaysa, dizgiyi doldurmak için veriler boş değer ile doldurulmalıdır.</p> <p>Kuyruk yöneticisi uygulamaya byte dizgileri döndürdüğünde (örneğin, MQGET çağrısında), kuyruk yöneticisi dizginin tanımlı uzunluğuna boş değer ile doldurur.</p> <p>Adlandırılmış değişmezler, bayt dizilimi alanlarının uzunluklarını tanımlamak için kullanılabilir. Bunlar “Değişmezler” sayfa 61 içinde listelenmiştir.</p>

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
MQCHAR	Karakter	<p>MQCHAR veri tipi, tek byte 'lık bir karakteri ya da çift byte 'lık ya da çok byte 'lık bir karakteri gösterir. Özel hizalama gerekmez.</p> <p>Farklı karakter kümeleri ya da kodlamaları kullanan kuyruk yöneticileri arasında MQCHAR verileri gönderildiğinde, MQCHAR verileri genellikle verilerin doğru yorumlanabilmesi için dönüştürme gerektirir. Kuyruk yöneticisi, MQMD yapısındaki MQCHAR verileri için bunu otomatik olarak yapar. Uygulama iletisi verilerindeki MQCHAR verilerinin dönüştürülmesi, MQGET çağrısında belirtilen MQGMO_CONVERT seçeneğiyle denetlenir; ek ayrıntılar için “MQGMO-Get-message seçenekleri” sayfa 368 içindeki bu seçeneğin tanımına bakın.</p>

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
MQCHARN	n karakterlik dizgi	<p>Her MQCHARn veri tipi n karakterli bir dizgiyi temsil eder; burada n şu değerlerden herhangi birini alabilir: 4, 8, 12, 20, 28, 32, 48, 64, 128 ya da 256. Her karakter MQCHAR veri tipi tarafından tanımlanır. Özel hizalama gerekmez.</p> <p>Dizilimdeki veriler, dizilimin tanımlı uzunluğundan kısaysa, dizilimi doldurmak için veriler boşluklarla doldurulmalıdır. Bazı durumlarda, dizgiyi boşluklarla doldurmak yerine, zamanından önce sonlandırmak için boş karakter kullanılabilir; boş karakter ve onu izleyen karakterler, dizginin tanımlı uzunluğuna kadar boşluk olarak değerlendirilir. Boş değer kullanılabilmesi için yerler, çağrı ve veri tipi açıklamalarında tanımlanır.</p> <p>Kuyruk yöneticisi uygulamaya karakter dizgileri döndürdüğünde (örneğin, MQGET çağrısında), kuyruk yöneticisi her zaman dizginin tanımlı uzunluğuna kadar boşluklarla doldurur; kuyruk yöneticisi dizgiyi sınırlamak için boş karakteri kullanmaz.</p> <p>Karakter dizilimi alanlarının uzunluklarını tanımlayan ve "Değişmezler" sayfa 61 içinde listelenen, adlandırılmış değişmezler vardır.</p>
MQFLOAT32	32 bitlik kayan noktalı sayı	<p>MQFLOAT32 veri tipi, standart IEEE kayan nokta biçimi kullanılarak gösterilen 32 bitlik bir kayan noktalı sayıdır. MQFLOAT32, 4 baytlık bir sınırdaki hizalanmalıdır.</p> <p>C ' de z/OS üzerinde MQFLOAT32 kullanımı, FLOAT (IEEE) derleyici işaretinin kullanılmasını gerektirir.</p> <p>COBOL dilinde MQFLOAT32 kullanımı, IEEE biçiminde kayan noktalı sayıları destekleyen derleyicilerle sınırlıdır. Bu, FLOAT (NATIVE) derleyici işaretinin kullanılmasını gerektirebilir.</p>

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
MQFLOAT64	64 bitlik kayan noktalı sayı	<p>MQFLOAT64 veri tipi, standart IEEE kayan nokta biçimi kullanılarak gösterilen 64 bitlik bir kayan noktalı sayıdır. MQFLOAT64 8 baytlık bir sınırdaki hizalanmalıdır.</p> <p>C ' de z/OS üzerinde MQFLOAT64 kullanımı, FLOAT (IEEE) derleyici işaretinin kullanılmasını gerektirir.</p> <p>COBOL dilinde MQFLOAT64 kullanımı, IEEE biçiminde kayan noktalı sayıları destekleyen derleyicilerle sınırlıdır. Bu, FLOAT (NATIVE) derleyici işaretinin kullanılmasını gerektirebilir.</p>
MQHCONFIG	Yapılandırma tanıtıcısı	<p>MQHCONFIG veri tipi, belirli bir kurulabilir hizmet için yapılandırılmakta olan bir yapılandırma tanıtıcısını gösterir. Bir yapılandırma tanıtıcısı, doğal sınırında hizalanmalıdır.</p> <p>Uygulamalar, bu tanıtıcı içinde saklanan verilerin biçimine güvenmemelidir. Geçerliyse, değeri sonraki MQI çağrılarında kullanılabilir olması amaçlanmıştır, ancak bu amacın dışında herhangi bir anlamı olması amaçlanmamıştır.</p>
MQHCONN	Bağlantı tanıtıcısı	<p>MQHCONN veri tipi, belirli bir kuyruk yöneticisine yönelik bağlantı tanıtıcısını gösterir. Bağlantı tanıtıcısı 4 baytlık bir sınırdaki hizalanmalıdır.</p> <p>Uygulamalar, bu tanıtıcı içinde saklanan verilerin biçimine güvenmemelidir. Geçerliyse, değeri sonraki MQI çağrılarında kullanılabilir olması amaçlanmıştır, ancak bu amacın dışında herhangi bir anlamı olması amaçlanmamıştır.</p>

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
MQHMSG	İleti tanıtıcısı	<p>MQHMSG veri tipi, bir iletiye erişim veren ileti tanıtıcısını gösterir. İleti tanıtıcısı 8 baytlık bir sınırdaki hizalanmalıdır.</p> <p>Uygulamalar, bu tanıtıcı içinde saklanan verilerin biçimine güvenmemelidir. Geçerliyse, değeri sonraki MQI çağrılarında kullanılabilir olması amaçlanmıştır, ancak bu amacın dışında herhangi bir anlamı olması amaçlanmamıştır.</p>
MQHOBJ	Nesne tanıtıcısı	<p>MQHOBJ veri tipi, bir nesneye erişim veren bir nesne tanıtıcısını gösterir. Nesne tanıtıcısı, 4 baytlık bir sınırdaki hizalanmalıdır.</p> <p>Uygulamalar, bu tanıtıcı içinde saklanan verilerin biçimine güvenmemelidir. Geçerliyse, değeri sonraki MQI çağrılarında kullanılabilir olması amaçlanmıştır, ancak bu amacın dışında herhangi bir anlamı olması amaçlanmamıştır.</p>
MQINT8	8 bitlik işaretli tamsayı	<p>MQINT8 veri tipi, bağlam tarafından tersi belirtilmedikçe, -128-+127 aralığındaki herhangi bir değeri alabilen 8 bitlik imzalı bir tamsayıdır.</p>
MQINT16	16 bitlik işaretli tamsayı	<p>MQINT16 veri tipi, bağlam tarafından tersi belirtilmedikçe, -32 768-+32 767 aralığındaki herhangi bir değeri alabilen 16 bitlik imzalı bir tamsayıdır. MQINT16 , 2 baytlık bir sınırdaki hizalanmalıdır.</p>
MQINT32	32 bit işaretli tamsayı	<p>MQINT32 veri tipi, bağlam tarafından tersi belirtilmedikçe, -2 147 483 648-+ 2 147 483 647 aralığındaki herhangi bir değeri alabilen 32 bitlik imzalı bir ikili tamsayıdır.</p> <p>MQLONGtanımlamasına bakın.</p>

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
MQINT64	64 bit işaretli tamsayı	<p>MQINT64 veri tipi, -9 223 372 036 854 775 808-+ 9 223 372 036 854 775 807 aralığındaki herhangi bir değeri alabilen 64 bitlik imzalı bir tamsayıdır.</p> <p>COBOL için geçerli değer aralığı, -999 999 999 999 999 999 ile 999 999 999 arasında bir değerle sınırlıdır. 64 bitlik bir tamsayı, 8 baytlık bir sınırdan hizalanmalıdır.</p>
MQLONG	32 bit işaretli tamsayı	<p>MQLONG veri tipi, bağlam tarafından tersi belirtilmediği sürece, -2 147 483 648-+ 2 147 483 647 aralığındaki herhangi bir değeri alabilen 32 bitlik imzalı bir ikili tamsayıdır.</p> <p>COBOL için geçerli aralık, -999 999 ile +999 999 999 999 arasında bir değerle sınırlıdır. MQLONG, 4 byte 'lık bir sınırdan hizalanmalıdır.</p>
MQPID	Süreç tanıtıcısı	<p>IBM MQ işlem tanıtıcısı.</p> <p>Bu, MQ izleme ve FFST™ dökümlerinde kullanılan tanıtıcıyla aynıdır, ancak işletim sistemi işlem tanıtıcısından farklı olabilir.</p>
MQPTR	Pointer	<p>MQPTR veri tipi, herhangi bir tipteki verilerin adresidir. Bir işaretçinin doğal sınırıyla hizalanması gerekir; bu, IBM üzerinde 16 baytlık bir sınır ve diğer platformlarda 8 baytlık bir sınırdır.</p> <p>Bazı programlama dilleri tip atanmış göstergeleri destekler; MQI bunları birkaç durumda da kullanır (örneğin, C programlama dilinde PMQCHAR ve PMQLONG).</p>
MQTID	İş parçacığı tanıtıcısı	<p>IBM MQ iş parçacığı tanıtıcısı.</p> <p>Bu, MQ izlemesinde ve FFST™ dökümlerinde kullanılan tanıtıcıyla aynıdır, ancak işletim sistemi iş parçacığı tanıtıcısından farklı olabilir.</p>

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)		
Temel veri tipi adı	Veri tipi	Açıklama
MQUINT8	8 bitlik işaretli tamsayı	MQUINT8 veri tipi, bağlam tarafından tersi belirtilmedikçe, 0 ile +255 aralığındaki herhangi bir değeri alabilen, 8 bitlik işaretli bir tamsayıdır.
MQUINT16	16 bit işaretli tamsayı	MQUINT16 veri tipi, bağlam tarafından tersi belirtilmedikçe, 0-+65 535 aralığındaki herhangi bir değeri alabilen, 16 bitlik işaretli bir tamsayıdır. MQUINT16, 2 baytlık bir sınırdır hizalanmalıdır.
MQUINT32	32 bit işaretli tamsayı	MQUINT32 veri tipi, 32 bitlik imzalanmamış bir ikili tamsayıdır. MQULONG tanımlamasına bakın.
MQUINT64	64 bit işaretli tamsayı	MQUINT64 veri tipi, bağlam tarafından tersi belirtilmedikçe, 0 ile +18 446 744 073 709 551 615 aralığındaki herhangi bir değeri alabilen 64 bitlik bir işaretli tamsayıdır. COBOL için, geçerli aralık 0 ile +999 999 999 999 999 999 arasında bir değerle sınırlıdır. 64 bitlik bir tamsayı, 8 baytlık bir sınırdır hizalanmalıdır.
MQULONG	32 bit işaretli tamsayı	MQULONG veri tipi, bağlamla kısıtlanmadıkça, 0-+ 4 294 967 294 aralığındaki herhangi bir değeri alabilen 32 bitlik imzalanmamış bir ikili tamsayıdır. COBOL için geçerli aralık, 0 ile +999 999 999 999 arasında bir değerle sınırlıdır. MQULONG, 4 baytlık bir sınırdır hizalanmalıdır.
PMQACH	Pointer	MQACH tipindeki bir veri yapısına ilişkin gösterge
PMQAIR	Pointer	MQAIR tipindeki bir veri yapısına ilişkin gösterge
PMQAXC	Pointer	MQAXC tipindeki bir veri yapısına ilişkin gösterge
PMQAXP	Pointer	MQAXP tipindeki bir veri yapısına ilişkin gösterge
PMQBMHO	Pointer	MQBMHO tipindeki bir veri yapısına ilişkin gösterge

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
PMQBO	Pointer	MQBO tipindeki bir veri yapısına ilişkin gösterge
PMQBOOL	Pointer	MQBOOL tipindeki verilere ilişkin gösterge
PMQBYTE	Pointer	MQBYTE tipindeki verilere ilişkin gösterge
PMQBYTE _n	Pointer	MQBYTE _n tipindeki verilere ilişkin gösterge; burada n, 8, 16, 24, 32, 40, 128 olabilir.
PMQCBC	Pointer	MQCBC tipindeki bir veri yapısına ilişkin gösterge
PMQCBD	Pointer	MQCBD tipindeki bir veri yapısına ilişkin gösterge
PMQCHAR	Pointer	MQCHAR tipindeki verilere ilişkin gösterge
PMQCHARN	Pointer	MQCHARN veri tipine ilişkin gösterge; burada n, 4, 8, 12, 20, 28, 32, 48, 64, 128, 256, 264 olabilir.
PMQCHARV	Pointer	MQCHARV tipindeki bir veri yapısına ilişkin gösterge
PMQCIH	Pointer	MQCIH tipindeki bir veri yapısına ilişkin gösterge
PMQCMHO	Pointer	MQCMHO tipindeki bir veri yapısına ilişkin gösterge
PMQCNO	Pointer	MQCNO tipindeki bir veri yapısına ilişkin gösterge
PMQCSP	Pointer	MQCSP tipindeki bir veri yapısına ilişkin gösterge
PMQCTLO	Pointer	MQCTLO tipindeki bir veri yapısına ilişkin gösterge
PMQDH	Pointer	MQDH tipindeki bir veri yapısına ilişkin gösterge
PMQDHO	Pointer	MQDHO tipindeki bir veri yapısına ilişkin gösterge
PMQDLH	Pointer	MQDLH tipindeki bir veri yapısına ilişkin gösterge
PMQDMHO	Pointer	MQDMHO tipindeki bir veri yapısına ilişkin gösterge
PMQDMPO	Pointer	MQDMPO tipindeki bir veri yapısına ilişkin gösterge
PMQEPH	Pointer	MQEPH tipindeki bir veri yapısına ilişkin gösterge

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
PMQFLOAT32	Pointer	MQFLOAT32 tipindeki bir veri yapısına ilişkin gösterge
PMQFLOAT64	Pointer	MQFLOAT64 tipindeki bir veri yapısına ilişkin gösterge
PMQFUNC	Pointer	Bir işleve ilişkin gösterge
PMQGMO	Pointer	MQGMO tipindeki bir veri yapısına ilişkin gösterge
PMQHCONFIG	Pointer	MQHCONFIG tipindeki verilere ilişkin gösterge
PMQHCONN	Pointer	MQHCONN tipindeki verilere ilişkin gösterge
PMQHMSG	Pointer	MQHMSG tipindeki verileri gösteren gösterge
PMQHOBJ	Pointer	MQHOBJ tipindeki verilere ilişkin gösterge
PMQIIH	Pointer	MQIIH tipindeki bir veri yapısına ilişkin gösterge
PMQIMPO	Pointer	MQIMPO tipindeki bir veri yapısına ilişkin gösterge
PMQINT8	Pointer	MQINT8 tipindeki verileri gösteren gösterge
PMQINT16	Pointer	MQINT16 tipindeki verileri gösteren gösterge
PMQINT32	Pointer	MQINT32 tipindeki verilere ilişkin gösterge
PMQINT64	Pointer	MQINT64 veri tipine ilişkin gösterge
PMQLONG	Pointer	MQLONG tipindeki verilere ilişkin gösterge
PMQMD	Pointer	MQMD tipindeki yapıya ilişkin gösterge
PMQMDE	Pointer	MQMDE tipindeki bir veri yapısına ilişkin gösterge
PMQMD1	Pointer	MQMD1 tipindeki bir veri yapısına ilişkin gösterge
PMQMD2	Pointer	MQMD2 tipindeki bir veri yapısına ilişkin gösterge
PMQMHBO	Pointer	MQMHBO tipindeki bir veri yapısına ilişkin gösterge
PMQOD	Pointer	MQOD tipindeki bir veri yapısına ilişkin gösterge
PMQOR	Pointer	MQOR tipindeki bir veri yapısına ilişkin gösterge

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
PMQPD	Pointer	MQPD tipindeki bir veri yapısına ilişkin gösterge
PMQPID	Pointer	Bir süreç tanıtıcısına ilişkin işaretçi
PMQMD	Pointer	MQMD tipindeki bir veri yapısına ilişkin gösterge
PMQPMO	Pointer	MQPMO tipindeki bir veri yapısına ilişkin gösterge
PMQPTR	Pointer	MQPTR tipindeki verilere ilişkin gösterge
PMQRFH	Pointer	MQRFH tipindeki bir veri yapısına ilişkin gösterge
PMQRFH2	Pointer	MQRFH2 tipindeki bir veri yapısına ilişkin gösterge
PMQRMH	Pointer	MQRMH tipindeki bir veri yapısına ilişkin gösterge
PMQRR	Pointer	MQRR tipindeki bir veri yapısına ilişkin gösterge
PMQSCO	Pointer	MQSCO tipindeki bir veri yapısına ilişkin gösterge
PMQSD	Pointer	MQSD tipindeki bir veri yapısına ilişkin gösterge
PMQSMPO	Pointer	MQSMPO tipindeki bir veri yapısına ilişkin gösterge
PMQSRO	Pointer	MQSRO tipindeki bir veri yapısına ilişkin gösterge
PMSSTS	Pointer	MQSTS tipindeki bir veri yapısına ilişkin gösterge
PMQTID	Pointer	Bir iş parçacığı tanıtıcısına ilişkin gösterge
PMQTM	Pointer	MQTM tipindeki bir veri yapısına ilişkin gösterge
PMQTM2	Pointer	MQTM2 tipindeki bir veri yapısına ilişkin gösterge
PMQUINT8	Pointer	MQUINT8 veri tipine ilişkin gösterge
PMQUINT16	Pointer	MQUINT16 veri tipine ilişkin gösterge
PMQUINT32	Pointer	MQUINT32 veri tipine ilişkin gösterge
PMQUINT64	Pointer	MQUINT64 veri tipine ilişkin gösterge
PMQULONG	Pointer	MQULONG veri tipine ilişkin gösterge

Çizelge 457. Temel veri tipi adları, tipleri ve açıklamaları (devamı var)

Temel veri tipi adı	Veri tipi	Açıklama
PMQVOID	Pointer	
PMQWIIH	Pointer	MQWIIH tipindeki bir veri yapısına ilişkin gösterge
PMQXQH	Pointer	MQXQH tipindeki bir veri yapısına ilişkin gösterge

C veri tipi bildirimleri

Çizelge 458. C veri tipi adları ve gösterimleri

Veri tipi	Gösterim
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>

Çizelge 458. C veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
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>
MQCHAR128	<code>typedef MQCHAR MQCHAR128[128];</code>
MQCHAR256	<code>typedef MQCHAR MQCHAR256[256];</code>
MQFLOAT32	<code>typedef float MQFLOAT32;</code>
MQFLOAT64	<code>typedef double MQFLOAT64;</code>
MQHCONFIG	<code>typedef void MQPOINTER MQHCONFIG;</code>
MQHCONN	<code>typedef MQLONG MQHCONN;</code>
MQHOBJ	<code>typedef MQLONG MQHOBJ;</code>
MQINT8	<code>typedef signed char MQINT8;</code>
MQINT16	<code>typedef short MQINT16;</code>

Çizelge 458. C veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
MQINT64	<p>UNIX 64 bit UNIX üzerinde:</p> <pre>typedef long;</pre> <p>AIX 32 bit AIX üzerinde:</p> <pre>typedef int64_t;</pre> <p>IBM i Linux z/OS Linux, IBM i ve z/OS sistemlerinde:</p> <pre>typedef long long;</pre> <p>Windows Windows'ta:</p> <pre>typedef _int64;</pre>
MQLONG	<p>IBM i IBM i'ta:</p> <pre>typedef long MQLONG;</pre> <p>z/OS ALW Diğer platformlarda:</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>

Çizelge 458. C veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
MQUINT64	<p>UNIX 64 bit UNIX üzerinde:</p> <pre>typedef unsigned long;</pre> <p>AIX 32 bit AIX üzerinde:</p> <pre>typedef uint64_t;</pre> <p>IBM i Linux z/OS Linux, IBM ve z/OS sistemlerinde:</p> <pre>typedef unsigned long long;</pre> <p>Windows Windows'ta:</p> <pre>typedef unsigned _int64;</pre>
MQULONG	<p>IBM i IBM i'ta:</p> <pre>typedef unsigned long MQULONG;</pre> <p>z/OS ALW Diğer platformlarda:</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>

Çizelge 458. C veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
PMQBYTE32	<code>typedef MQBYTE32[32] MQPOINTER PMQBYTE32[32];</code>
PMQBYTE40	<code>typedef MQBYTE40[40] MQPOINTER PMQBYTE40[40];</code>
PMQBYTE128	<code>typedef MQBYTE128[128] MQPOINTER PMQBYTE128[128];</code>
PMQCHAR	<code>typedef MQCHAR MQPOINTER PMQCHAR;</code>
PMQCHAR4	<code>typedef MQCHAR4[4] MQPOINTER PMQCHAR4[4];</code>
PMQCHAR8	<code>typedef MQCHAR8[8] MQPOINTER PMQCHAR8[8];</code>
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>

Çizelge 458. C veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
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>
PMQGM0	<code>typedef MQGM0 MQPOINTER PMQGM0;</code>
PMQHCONFIG	<code>typedef MQHCONFIG MQPOINTER PMQHCONFIG;</code>
PMQHCONN	<code>typedef MQHCONN MQPOINTER PMQHCONN;</code>
PMQH0BJ	<code>typedef MQH0BJ MQPOINTER PMQH0BJ;</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>

Çizelge 458. C veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
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>
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>

<i>Çizelge 458. C veri tipi adları ve gösterimleri (devamı var)</i>	
Veri tipi	Gösterim
PPMQCHAR	typedef PMQCHAR MQPOINTER PPMQCHAR;
PPMQCNO	typedef PMQCNO MQPOINTER PPMQCNO;
PPMQGMO	typedef PMQGMO MQPOINTER PPMQGMO;
PPMQHCONN	typedef PMQHCONN MQPOINTER PPMQHCONN;
PPMQHOBJ	typedef PMQHOBJ MQPOINTER PPMQHOBJ;
PPMQLONG	typedef PMQLONG MQPOINTER PPMQLONG;
PPMQMD	typedef PMQMD MQPOINTER PPMQMD;
PPMQOD	typedef PMQOD MQPOINTER PPMQOD;
PPMQPMO	typedef PMQPMO MQPOINTER PPMQPMO;
PPMQULONG	typedef PMQULONG MQPOINTER PPMQULONG;
PPMQVOID	typedef PMQVOID MQPOINTER PPMQVOID;
Burada defined (MQ_64_BIT) , 64 bitlik bir platform anlamına gelir.	

MQPOINTER makro değişkeninin tanımı için bkz. [“Veri türleri” sayfa 263](#) .

COBOL veri tipi bildirimleri

<i>Çizelge 459. COBOL veri tipi adları ve gösterimleri</i>	
Veri tipi	Gösterim
MQBOOL	PIC S9(9) BINARY
MQBYTE	PIC X

Çizelge 459. COBOL veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
MQBYTE8	PIC X(8)
MQBYTE16	PIC X(16)
MQBYTE24	PIC X(24)
MQBYTE32	PIC X(32)
MQBYTE40	PIC X(40)
MQCHAR	PIC X
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)

Çizelge 459. COBOL veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
MQFLOAT32	USAGE COMP-1
MQFLOAT64	USAGE COMP-2
MQHCONN	Açıkz/OS PIC S9(9) COMP-5 Diğer platformlarda PIC S9(9) BINARY
MQHOBj	PIC S9(9) BINARY
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

PL/I veri tipi bildirimleri

Çizelge 460. PL/I veri tipi adları ve gösterimleri

Veri tipi	Gösterim
MQBOOL	fixed bin(31)
MQBYTE	char(1)
MQBYTE8	char(8)
MQBYTE16	char(16)
MQBYTE24	char(24)
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)

<i>Çizelge 460. PL/I veri tipi adları ve gösterimleri (devamı var)</i>	
Veri tipi	Gösterim
MQCHAR128	char(128)
MQCHAR256	char(256)
MQFLOAT32	binary float(21) ieee
MQFLOAT64	binary float(52) ieee
MQHCONN	fixed bin(31)
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)

High Level Assembler veri tipi bildirimleri

Çizelge 461. System/390 çevirici veri tipi adları ve gösterimleri

Veri tipi	Gösterim
MQBOOL	DS F
MQBYTE	DS XL1
MQBYTE8	DS XL8
MQBYTE16	DS XL16
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

Çizelge 461. System/390 çevirici veri tipi adları ve gösterimleri (devamı var)

Veri tipi	Gösterim
MQCHAR128	DS CL128
MQCHAR256	DS CL256
MQFLOAT32	DS EB
MQFLOAT64	DS DB
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

Yapı veri tipleri

Yapı veri tiplerinin özeti, MQI yapılarının tutarlı bir şekilde eşlenmesine ilişkin kurallar ve her yapı veri tipi tanımında kullanılan kurallar.

- “MQI çağrılarında ya da çıkış işlevlerinde kullanılan yapı veri tiplerinin özeti” sayfa 259
- “İleti verilerinde kullanılan yapı veri tiplerinin özeti” sayfa 260
- “MQI yapılarının tutarlı bir şekilde eşlenmesine ilişkin kurallar” sayfa 260
- “Her yapı veri tipi tanımında kullanılan kurallar” sayfa 261

MQI çağrılarında ya da çıkış işlevlerinde kullanılan yapı veri tiplerinin özeti

<i>Çizelge 462. MQI çağrılarında ya da çıkış işlevlerinde kullanılan yapı veri tipleri</i>		
Yapı	Açıklama	Kullanıldığı yerdeki çağrılar
MQACH	API çıkış zinciri üstbilgisi	
MQAIR	Kimlik doğrulama bilgileri kaydı	MQCONN
MQAXC	API çıkış bağlamı	
MQAXP	API çıkış parametresi	
MQBMHO	İleti işleme seçenekleri için arabellek	MQBUFMH
MQBO	Başlangıç seçenekleri	MQBEGIN
MQCBD	Geri çağırma tanımlayıcısı	MQCB
MQCBO	Create-bag seçenekleri	mqCreateBag
MQCHARV	Değişken uzunluklu dizgi	MQINQMP
MQCNO	Bağlantı seçenekleri	MQCONN
MQCSP	Güvenlik parametreleri	MQCONN
MQCTLO	Geri Arama Seçenekleri	MQCTL
MQDMPO	İleti özelliği seçeneklerini sil	MQDLTMP
MQGMO	İleti alma seçenekleri	MQGet
MQIMPO	İleti özelliği sorma seçenekleri	MQINQMP
Mqmd	İleti tanımlayıcı	MQBUFMH , MQMHBUF , MQCB , MQGET , MQPUT , MQPUT1
MQMHBO	İleti tanıtıcısı-arabellek seçenekleri	MQMHBUF
MQOD	Nesne tanımlayıcı	MQOPEN , MQPUT1
MQOR	Nesne kaydı	MQOPEN , MQPUT1
MQPD	Özellik tanımlayıcı	MQSETMP
MQPMO	Koyma iletisi seçenekleri	MQPUT , MQPUT1
MQPMR	Koyma iletisi kaydı	MQPUT , MQPUT1
MQRR	Yanıt kaydı	MQOPEN , MQPUT , MQPUT1
MQSCO	TLS yapılandırma seçenekleri	MQCONN
MQSD	Abonelik tanımlayıcısı	MQSUB
MQSMPO	İleti özelliği seçeneğini ayarla	MQSETMP
MQSRO	Abonelik isteği seçenekleri	MQSUBRQ

Çizelge 462. MQI çağrılarında ya da çıkış işlevlerinde kullanılan yapı veri tipleri (devamı var)

Yapı	Açıklama	Kullanıldığı yerdeki çağrılar
<u>MQSTS</u>	Durum raporlama yapısı	<u>MQSTAT</u>

İleti verilerinde kullanılan yapı veri tiplerinin özeti

Çizelge 463. İleti verilerinde kullanılan yapı veri tipleri

Yapı	Açıklama
<u>MQCIH</u>	CICS bilgi üstbilgisi
<u>MQCFH</u>	PCF üstbilgisi
<u>MQEPH</u>	Yerleşik PCF üstbilgisi
<u>MQDHD</u>	Dağıtım üstbilgisi
<u>MQDLH</u>	Teslim edilmemiş harf (teslim edilmemiş ileti) üstbilgisi
<u>MQIIH</u>	IMS bilgi üstbilgisi
<u>MQMDE</u>	İleti tanımlayıcı uzantısı
<u>MQRFH</u>	Kurallar ve biçimlendirme üstbilgisi
<u>MQRFH2</u>	Kurallar ve biçimlendirme üstbilgisi 2
<u>MQRMH</u>	Başvuru iletisi üstbilgisi
<u>MQTM</u>	Tetikleyici ileti
<u>MQTMC2</u>	Tetikleyici ileti (karakter biçimi 2)
<u>MQWIH</u>	İş bilgileri üstbilgisi
<u>MQXQH</u>	İletim kuyruğu üstbilgisi

Not: MQDXP yapısı (veri dönüştürme çıkış parametresi), ilişkili veri dönüştürme çağrılarıyla birlikte “Veri dönüştürme çıkışı” sayfa 895’inde açıklanmıştır.

MQI yapılarının tutarlı bir şekilde eşlenmesine ilişkin kurallar

Programlama dilleri, yapılara ilişkin destek düzeylerinde değişiklik gösterir ve MQI yapılarını her programlama dilinde tutarlı olarak eşlemek için belirli kurallar ve kurallar benimsenmiştir:

1. Yapılar doğal sınırlarıyla hizalanmalıdır.

- Çoğu MQI yapısı 4 baytlık hizalama gerektirir.
- IBM üzerinde, işaretçileri içeren yapılar 16 baytlık hizalama gerektirir; bunlar: MQCNO, MQOD, MQPMO.

2. Bir yapıdaki her alan doğal sınırında hizalanmalıdır.

- MQLONG ' ye eşit veri tiplerine sahip alanlar, 4 baytlık sınırlarda hizalanmalıdır.
- MQPTR ' ye eşit veri tiplerine sahip alanlar, IBM ürününde 16 baytlık sınırlara ve diğer ortamlarda 4 baytlık sınırlara hizalanmalıdır.
- Diğer alanlar 1 baytlık sınırlara hizalanır.

3. Bir yapının uzunluğu, yapının sınır hizalamasının katı olmalıdır.

- Çoğu MQI yapısının 4 byte 'ın katları olan uzunlukları vardır.
- IBM üzerinde, işaretçi içeren yapıların 16 baytlık katları olan uzunlukları vardır.

4. Gerektiğinde, önceki kurallara uyulmasını sağlamak için doldurma baytları ya da alanları eklenmelidir.

Her yapı veri tipi tanımında kullanılan kurallar

Her yapı veri tipinin tanımı şunları içerir:

- Yapının amacına ve kullanımına genel bakış
- Programlama dilinden bağımsız bir biçimde, yapıdaki alanların açıklamaları
- Yapının desteklenen programlama dillerinin her birinde nasıl bildirildiğine ilişkin örnekler

Her yapı veri tipinin açıklaması aşağıdaki kısımları içerir:

Yapı adı

Yapının adı ve ardından yapıdaki alanların özeti.

Genel Bakış

Yapının amacının ve kullanımının kısa bir açıklaması.

Alanlar

Alanların açıklamaları. Her alan için, alanın adını parantez () içindeki temel veri tipi izler. Metinde, alan adları italik yazı tipi kullanılarak gösterilir; örneğin, *Version*.

Alanın amacına ilişkin bir açıklama ve alanın alabileceği değerlerin bir listesi de vardır. Değişmezlerin adları büyük harfle gösterilir; örneğin, MQGMO_STRUC_ID. * karakteri kullanılarak, önceki aynı olan sabit değerler kümesi gösterilir; örneğin: MQIA_ *.

Alanların açıklamalarında aşağıdaki terimler kullanılır:

Giriş

Arama yaptığınızda alana bilgi girersiniz.

output

Çağrı tamamlandığında ya da başarısız olduğunda kuyruk yöneticisi alandaki bilgileri döndürür.

giriş/çıkış

Bir arama yaptığınızda alana bilgi girersiniz ve arama tamamlandığında ya da başarısız olduğunda kuyruk yöneticisi bilgileri değiştirir.

Başlangıçtaki değerler

MQI ile verilen veri tanımlaması dosyalarındaki her alana ilişkin başlangıç değerlerini gösteren bir çizelge.

C bildirim

C ' deki yapının tipik beyannamesi.

COBOL bildirim

COBOL ' da yapının tipik açıklaması.

PL/I bildirim

PL/I ' de yapının tipik açıklaması.

High Level Assembler bildirim

System/390 çevirici dilinde yapının tipik bildirim.

Visual Basic bildirim

Visual Basic 'de yapının tipik beyannamesi.



C programlama

C programlama dilindeki MQI ' ı kullanmanıza yardımcı olacak bilgiler.

- [“Üstbilgi dosyaları” sayfa 262](#)
- [“İşlevler” sayfa 262](#)
- [“Tanımlanmamış veri tipine sahip parametreler” sayfa 262](#)
- [“Veri türleri” sayfa 263](#)
- [“İkili dizgiler işleniyor” sayfa 263](#)

- “Karakter dizilimlerini işleme” sayfa 263
- “Yapılara ilişkin ilk değerler” sayfa 263
- “Dinamik yapılar için ilk değerler” sayfa 264
- “C++ dilinden kullan” sayfa 264
- “Gösterim kuralları” sayfa 264

Üstbilgi dosyaları

Çizelge 464. C üstbilgi dosyaları	
Dosya	İçerik
CMQC	Ana MQI için işlev prototipleri, veri tipleri ve adlandırılmış sabitler
CMQXC	Veri dönüştürme çıkışı için işlev prototipleri, veri tipleri ve adlandırılmış sabitler
CMQEC	Ana MQI için işlev prototipleri, veri tipleri ve adlandırılmış sabitler, veri dönüştürme çıkışı ve Arabirim Giriş Noktaları yapısı (CMQEC, CMQXC ve CMQC ' yi içerir)
CMQSTRC	MQI değişmez tanımlamalarını metin eşdeğerine dönüştüren işlevler.  Uyarı:  IBM MQ 9.1' den z/OS için geçerlidir. Bu üstbilgi dosyasını kullanan programlar LONGNAME derleyici seçeneğiyle derlenmelidir.

Uygulamaların taşınabilirliğini artırmak için, `#include` ön işlemci yönergesinde üstbilgi dosyasının adını küçük harfle kodlayın:

```
#include "cmqec.h"
```

İşlevler

Bir işlevi her çağırdığında adres tarafından geçirilen tüm parametreleri belirtmenize gerek yoktur.

- *yalnızca giriş* ve MQHCONN, MQHOBJ ya da MQLONG tipinde parametreleri değere göre geçirin.
- Diğer tüm parametreleri adrese göre iletir.

Belirli bir parametrenin gerekli olmadığı durumlarda, parametre verilerinin adresi yerine, işlev çağrısında parametre olarak boş bir gösterge kullanın. Bunun mümkün olduğu parametreler, çağrı açıklamalarında tanımlanır.

İşlevin değeri olarak parametre döndürülmez; C terminolojisinde bu, tüm işlevlerin voiddeğerini döndürdüğü anlamına gelir.

İşlevin öznitelikleri MQENTRY makro değişkeniyle tanımlanır; bu makro değişkeninin değeri ortama bağlıdır.

Tanımlanmamış veri tipine sahip parametreler

MQGET, MQPUT ve MQPUT1 işlevlerinde **Buffer** değiştirgesinin tanımlı olmayan bir veri tipi var. Bu parametre, uygulamanın ileti verilerini göndermek ve almak için kullanılır.

Bu sıralamaya ilişkin parametreler, C örneklerinde MQBYTE dizileri olarak gösterilir. Parametreleri bu şekilde bildirebilirsiniz, ancak genellikle bunları iletideki verilerin düzenini tanımlayan belirli bir yapı olarak bildirmek daha uygundur. Gerçek işlev değiştirgesini gösterge-to-void olarak bildirin ve işlev çağrısında parametre olarak herhangi bir veri türünün adresini belirtin.

Veri türleri

C `typedef` deyimini kullanarak tüm veri tiplerini tanımlayın. Her veri tipi için, ilgili gösterge veri tipini de tanımlayın. İşaretçi veri tipinin adı, bir işaretçiyi göstermek için öneki P harfi olan temel veri ya da yapı veri tipinin adıdır. `MQPOINTER` makro değişkenini kullanarak göstergenin özniteliklerini tanımlayın; bu makro değişkeninin değeri ortama bağlıdır. Aşağıda işaretçi veri tiplerinin nasıl bildirileceği gösterilmektedir:

```
#define MQPOINTER *          /* depends on environment */
...
typedef MQLONG MQPOINTER PMQLONG; /* pointer to MQLONG */
typedef MQMD MQPOINTER PMQMD; /* pointer to MQMD */
```

İkili dizgiler işleniyor

İkili veri dizgilerini `MQBYTE`n veri tiplerinden biri olarak bildirin.

Bu tipteki alanları kopyaladığınızda, karşılaştırdığınızda ya da ayarladığınızda, C işlevlerini **memcpy**, **memcpya** da **memset** kullanın; örneğin:

```
#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));
```

strcpy, **strncpy**, **strncpy**ya da **strncmp**dizgi işlevlerini kullanmayın; bunlar, `MQBYTE`n veri tipleriyle bildirilen veriler için doğru biçimde çalışmaz.

Karakter dizilimlerini işleme

Kuyruk yöneticisi uygulamaya karakter verileri döndürdüğünde, kuyruk yöneticisi karakter verilerini her zaman alanın tanımlı uzunluğuna kadar boşluklarla doldurur. Kuyruk yöneticisi boş sonlandırılmış dizgiler **döndürmez**.

Bu nedenle, bu tür dizgileri kopyalarken, karşılaştırırken ya da birleştirirken **strncpy**, **strncmp**ya da **strncat**dizgi işlevlerini kullanın.

Dizginin boş değerle (**strcpy**, **strncpy**, **strcat**) sonlandırılmasını gerektiren dizgi işlevlerini kullanmayın. Ayrıca, dizginin uzunluğunu saptamak için **strlen** işlevini kullanmayın; alanın uzunluğunu saptamak için **sizeof** işlevini kullanın.

Yapılara ilişkin ilk değerler

Üstbilgi dosyaları, bu yapıların eşgörünümlerini bildirirken MQ yapılarına ilişkin ilk değerleri sağlamak için kullanabileceğiniz çeşitli makro değişkenlerini tanımlar.

Bu makro değişkenlerinin adları `MQxxx_DEFAULT` biçiminde olur; burada `MQxxx`, yapının adını gösterir. Bunlar aşağıdaki şekilde kullanılır:

```
MQMD MyMsgDesc = {MQMD_DEFAULT};
MQPMO MyPutOpts = {MQPMO_DEFAULT};
```

Bazı karakter alanları için (örneğin, çoğu yapıda oluşan *StrucId* alanları ya da `MQMD` ' de oluşan *Format* alanı), `MQI` geçerli olan belirli değerleri tanımlar. Geçerli değerlerin her biri için *iki* makro değişkeni sağlanır:

- Bir makro değişkeni, değeri, örtük boş değer eşleşmeleri dışında, uzunluğu tam olarak alanın tanımlı uzunluğu olan bir dizgi olarak tanımlar. Örneğin, MQMD ' deki *Format* alanı için aşağıdaki makro değişkeni sağlar (¬ tek bir boş karakteri temsil eder):

```
#define MQFMT_STRING "MQSTR¬¬"
```

Bu formu memcpy ve memcmp işlevleriyle kullanın.

- Diğer makro değişkeni değeri bir karakter dizisi olarak tanımlar; bu makro değişkeninin adı, _ARRAY ile sonekli dizgi formunun adıdır. Örneğin:

```
#define MQFMT_STRING_ARRAY 'M','Q','S','T','R',' ',' ',' ',' '
```

Bir yapı eşgörünümünü MQMD_DEFAULT makro değişkeninin sağladıklarından farklı değerlerle bildirdiğinizde alanı kullanıma hazırlamak için bu formu kullanın. (Bu her zaman gerekli değildir; bazı ortamlarda değer dizgi biçimini her iki durumda da kullanabilirsiniz. Ancak, C++ programlama diliyle uyumluluk için bu gerektiğinden, bildirimler için dizi formunu kullanabilirsiniz.)

Dinamik yapılar için ilk değerler

Bir yapının değişken sayıda eşgörünümü gerektiğinde, eşgörünümler genellikle calloc ya da malloc işlevleri kullanılarak dinamik olarak elde edilen ana bellekte yaratılır. Bu tür yapılardaki alanları kullanıma hazırlamak için aşağıdaki tekniği göz önünde bulundurun:

1. Yapıyı kullanıma hazırlamak için uygun MQxxx_DEFAULT makro değişkenini kullanarak yapının bir eşgörünümünü bildirin. Bu eşgörünüm, diğer eşgörünümler için model olur:

```
MQMD Model = {MQMD_DEFAULT}; /* declare model instance */
```

Statik ya da auto anahtar sözcükleri, model eşgörünümüne gereken şekilde statik ya da dinamik geçerlik süresi vermek için bildirimde kodlanabilir.

2. Yapının dinamik bir eşgörünümü için depolama alanı elde etmek üzere calloc ya da malloc işlevlerini kullanın:

```
PMQMD Instance;  
Instance = malloc(sizeof(MQMD)); /* get storage for dynamic instance */
```

3. Model eşgörünümünü dinamik örneğe kopyalamak için memcpy işlevini kullanın:

```
memcpy(Instance,&Model,sizeof(MQMD)); /* initialize dynamic instance */
```

C++ dilinden kullan

C++ programlama dili için, üstbilgi dosyaları yalnızca C++ derleyicisi kullandığınızda içerilen aşağıdaki ek deyimleri içerir:

```
#ifndef __cplusplus  
extern "C" {  
#endif  
  
/* rest of header file */  
  
#ifdef __cplusplus  
}  
#endif
```

Gösterim kuralları

Bu bilgiler, işlevlerin nasıl çağrılacağını ve parametrelerin nasıl bildirileceğini gösterir.

Bazı durumlarda, parametreler değişmez büyüklükte olmayan dizilerdir. Bunlar için, sayısal bir sabiti göstermek için küçük harf n kullanılır. Bu değiştirgeye ilişkin bildirim kodladığınızda, n değerini gereken sayısal değerle değiştirin.

COBOL programlama

COBOL programlama dilindeki MQI ' ı kullanmanıza yardımcı olacak bilgiler.

- “Dosyaları kopyala” sayfa 265
- “Yapılar” sayfa 266
- “Göstergeler” sayfa 266
- “Adlandırılmış sabitler” sayfa 267
- “Gösterim kuralları” sayfa 267

Dosyaları kopyala

MQI kullanan COBOL uygulama programlarını yazmanıza yardımcı olmak için çeşitli COPY dosyaları sağlanır. Adlandırılmış değişmezleri içeren iki dosya ve her bir yapı için iki dosya vardır.

Her bir yapı iki biçimde sağlanır: ilk değerleri olan bir form ve aşağıdaki özelliklere sahip olmayan bir form:

- Bir COBOL programının ÇALIŞMA-DEPOLAMA bölümünde ilk değerleri olan yapıları kullanın; bunlar, adları V harfiyle sonekli COPY dosyalarında bulunur (Değerler için).
- Bir COBOL programının BAĞLANTı Bölümü ' ne ilişkin ilk değerleri olmayan yapıları kullanın; bunlar, adları L harfiyle eklenmiş COPY dosyalarında bulunur (Bağ için).

COPY dosyaları aşağıdaki çizelgede özetlenir. Listelenen dosyaların tümü tüm ortamlarda kullanılamaz.

Dosya (ilk değerlerle)	Dosya (başlangıç değerleri olmadan)	İçerik
CMQAIRV	CMQAIRL	Kimlik doğrulama bilgileri kaydı
CMQBOV	CMQBOL	Başlangıç seçenekleri yapısı
CMQCIHV	CMQCIHL	CICS bilgi üstbilgisi yapısı
CMQCNV	CMQCNOL	Bağlantı seçenekleri yapısı
CMQDHV	CMQDHL	Dağıtım üstbilgisi yapısı
CMQDLHV	CMQDLHL	Ölü harf üstbilgisi yapısı
CMQDXPV	CMQDXPL	Veri dönüştürme çıkış parametresi yapısı
CMQGMV	CMQGMOL	İleti alma seçenekleri yapısı
CMQIIHV	CMQIIHL	IMS bilgi üstbilgisi yapısı
CMQMDV	CMQMDL	İleti tanımlayıcı yapısı
CMQMDEV	CMQMDEL	İleti tanımlayıcı uzantısı yapısı
CMQMD1V	CMQMD1L	İleti tanımlayıcı yapısı sürüm 1
CMQODV	CMQODL	Nesne tanımlayıcı yapısı
CMQORV	CMQORL	Nesne kaydı yapısı
CMQPMV	CMQPMOL	Koyma iletisi seçenekleri yapısı
CMQRFHV	CMQRFHL	Kurallar ve biçimlendirme üstbilgi yapısı

Çizelge 465. COBOL COPY dosyaları (devamı var)

Dosya (ilk değerlerle)	Dosya (başlangıç değerleri olmadan)	İçerik
CMQRFH2V	CMQRFH2L	Kurallar ve biçimlendirme üstbilgi yapısı sürüm 2
CMQRMHV	CMQRMHL (QRMHL)	Başvuru iletisi üstbilgi yapısı
CMQRRV	CMQRRL	Yanıt kaydı yapısı
CMQSCOV	CMQSCOL	TLS yapılandırma seçenekleri
CMQTMV	CMQTML	Tetikleyici ileti yapısı
CMQTMCV	CMQTMCL	Tetikleyici ileti yapısı (karakter biçimi)
CMQTM2V	CMQTM2L	Tetikleyici ileti yapısı (karakter biçimi) sürüm 2
CMQWIHV	CMQWIHL	İş bilgileri üstbilgi yapısı
CMQXQHV	CMQXQHL	İletim kuyruğu üstbilgi yapısı
CMQV	-	Ana MQI için adlandırılmış sabitler
CMQXV	-	Veri dönüştürme çıkışı için adlandırılmış sabitler
CMQMD2V	CMQMD2L	İleti tanımlayıcı yapısı sürüm 2

Yapılar

COPY dosyasında, her yapı bildirimini bir level-10 ögesiyle başlar; bu, level-01 bildirimini kodlayarak ve yapı bildiriminin geri kalanında kopyalamak için COPY deyimini kullanarak yapının birkaç eşgörünümünü bildirmenizi sağlar. Uygun yönetim ortamına gönderme yapmak için IN anahtar sözcüğünü kullanın:

```
* 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.
```

Yapıları uygun sınırlar üzerinde hizalayın. COPY deyimini, level-01 ögesi olmayan bir ögeyi izleyen bir yapıyı içermek için kullanırsanız, yapının level-01 ögesinin başlangıcından uygun görel konumdan başladığından emin olun. Çoğu MQI yapısı 4 baytlık hizalama gerektirir; bunun kural dışı durumları, IBM üzerinde 16 baytlık hizalama gerektiren MQCNO, MQOD ve MQPMO ' dur.

Bu bölümde, yapıdaki alanların adları önek olmadan gösterilir. COBOL dilinde, alan adlarının başına yapının adı ve ardından bir kısa çizgi eklenir. Ancak yapı adı, yapının özgün yapının ikinci ya da sonraki bir sürümü olduğunu gösteren sayısal bir rakamla biterse, sayısal sayı önekten atlanır. COBOL içindeki alan adları büyük harfle gösterilir (ancak gerekirse küçük harf ya da büyük harf karışık kullanılabilir). Örneğin, "MQMD-İleti tanımlayıcı" sayfa 420 içinde açıklanan *MsgType* alanı COBOL içinde MQMD-MSGTYPE olur.

V soneki yapıları, tüm alanlar için ilk değerlerle bildirilir; yalnızca, sağlanan ilk değerden farklı bir değer istediğiniz alanları ayarlamanız gerekir.

Göstergeler

Bazı yapıların, MQOD yapısı tarafından adreslenen MQOR ve MQRR kayıtları gibi yapıyla bitişik olmayabilecek isteğe bağlı verileri ele almaları gerekir.

Bu isteğe bağlı verileri ele almak için yapılar, işaretçi veri tipiyle bildirilen alanları içerir. Ancak COBOL, işaretçi veri tipini tüm ortamlarda desteklemez. Bu nedenle, isteğe bağlı veriler, yapının başlangıcından itibaren verilerin görel konumunu içeren alanlar kullanılarak da adreslenebilirler.

Bir uygulamayı ortamlar arasında bağlantı kurmak istiyorsanız, işaretçi veri tipinin tüm amaçlanan ortamlarda kullanılabilir olup olmadığını belirleyin. Değilse, uygulamanın gösterge alanları yerine görel konum alanlarını kullanarak isteğe bağlı verileri ele alması gerekir.

İşaretçilerin desteklenmediği ortamlarda, işaretçi alanlarını uygun uzunluktaki bayt dizgileri olarak, ilk değeri tümü boş değerli byte dizgisi olarak bildirin. Görel konum alanlarını kullanıyorsanız, bu başlangıç değerini değiştirmeyin.

Adlandırılmış sabitler

Bu bilgilerde, adın bir parçası olarak alt çizgi karakterini (_) içeren sabitlerin adları gösterilir. COBOL dilinde, alt çizgi yerine kısa çizgi karakterini (-) kullanın.

Karakter dizilimi değerleri olan sabitler, dizilim sınırlayıcı (!) olarak tek tırnak işaretini kullanır. Bazı ortamlarda, derleyicinin çift tırnak işareti yerine tek tırnak işareti dizgi sınırlayıcı olarak kabul etmesini sağlamak için uygun bir derleyici seçeneği belirtmeniz gerekebilir.

Belirtilen sabitler, COPY dosyalarında level-10 öge olarak bildirilir. Değişmezleri kullanmak için, level-01 ögesini belirttik olarak bildirin ve değişmezlerin bildirimlerini kopyalamak için COPY deyimini kullanın:

```
* Declare a structure to hold the constants
01 MY-MQ-CONSTANTS.
   COPY CMQV.
```

Önceki yöntem, değişmezlere gönderme yapılmısa da, bu değişmezlerin programda saklama alanı kaplamasına neden olur. Değişmezleri aynı çalıştırma birimindeki birçok ayrı programa ekliyorsanız, değişmezlerin birden çok kopyası vardır ve bu da ana saklama alanını gereksiz yere tüketir. Aşağıdaki tekniklerden birini kullanarak bu etkiyi önleyin:

- GLOBAL yantümcesini level-01 bildirimine ekleyin:

```
* Declare a global structure to hold the constants
01 MY-MQ-CONSTANTS GLOBAL.
   COPY CMQV.
```

Bu, çalıştırma birimi içindeki tek bir sabit küme için saklama alanı ayırır. Ancak, bu değişmezlere yalnızca level-01 bildirimini içeren program değil, çalıştırma birimi içindeki herhangi bir program tarafından gönderme yapılabilir.

Not: GLOBAL yantümcesi tüm ortamlarda desteklenmez.

- Her programa yalnızca bu program tarafından başvuru alan değişmezleri el ile kopyalayın. Tüm değişmezleri programa kopyalamak için COPY deyimini kullanmayın.

Gösterim kuralları

Bu konudaki önceki kısımlar, çağrılarını nasıl çağıracağınızı ve parametreleri nasıl bildireceğinizi gösterir. Bazı durumlarda, parametreler boyutu sabit olmayan çizelge ya da karakter dizilimleridir. Bunlar için, sayısal bir sabiti göstermek için küçük harf n kullanılır. Bu değiştirgeye ilişkin bildirim kodladığınızda, n değerini gereken sayısal değerle değiştirin.

High Level Assembler programlama

System/390 Assembler programlama dilindeki MQI 'ı kullanmanıza yardımcı olacak bilgiler.

- [“Makrolar” sayfa 268](#)
- [“Yapılar” sayfa 268](#)
- [“CMQVERA makrosu” sayfa 268](#)

- “Gösterim kuralları” sayfa 269

Makrolar

Adlandırılmış sabitler için iki makro ve her yapı için bir makro vardır. Bu dosyalar aşağıdaki tabloda özetlenmiştir.

<i>Çizelge 466. Çevirici makroları</i>	
Dosya	İçerik
CMQA	Ana MQI için adlandırılmış sabitler (eşittir)
CMQCIHA	CICS bilgi üstbilgisi yapısı
CMQCNOA	Bağlantı seçenekleri yapısı
CMQDLHA	Ölü harf üstbilgisi yapısı
CMQDXPA	Veri dönüştürme çıkış parametresi yapısı
CMQGMOA	İleti alma seçenekleri yapısı
CMQIIHA	IMS bilgi üstbilgisi yapısı
CMQMDA	İleti tanımlayıcı yapısı
CMQMDEA	İleti tanımlayıcı uzantısı yapısı
CMQODA	Nesne tanımlayıcı yapısı
CMQPMOA	Koyma iletisi seçenekleri yapısı
CMQRFHA	Kurallar ve biçimlendirme üstbilgi yapısı
CMQRFH2A	Kurallar ve biçimlendirme üstbilgi yapısı sürüm 2
CMQRMHA	Başvuru iletisi üstbilgi yapısı
CMQTMA	Tetikleyici ileti yapısı
CMQTMC2A	Tetikleyici ileti yapısı (karakter biçimi) sürüm 2
CMQVERA	Yapı sürümü denetimi
CMQWIHA	İş bilgileri üstbilgi yapısı
CMQXA	Veri dönüştürme çıkışı için adlandırılmış sabitler
CMQXPA	API geçiş çıkış parametresi yapısı
CMQXQHA	İletim kuyruğu üstbilgi yapısı

Yapılar

Yapılar, makronun işlemini denetlemek için çeşitli parametreleri olan makrolar tarafından oluşturulur. Bakınız “Yapılar” sayfa 269

CMQVERA makrosu

Bu makro, yapı makrolarındaki DCLVER değiştirgesi için kullanılacak varsayılan değeri belirlemenizi sağlar.

CMQVERA tarafından belirtilen değer, yapı makrosu tarafından, DCLVER değiştirgesini yapı makrosunun çağrılmasından çıkarırsanız kullanılır. Varsayılan değer, DCLVER değiştirgesiyle CMQVERA makrosunun kodlanması ile belirlenir:

DCLVER=YRRLKTEKI

Varsayılan sürüm, geçerli (en son) sürüme ayarlanır.

DCLVER=BELIRTILDI

Varsayılan sürüm, VERSION deęiřtirgesiyle belirtilen sürüme ayarlanır.

DCLVER deęiřtirgesini belirtmeniz ve deęerin büyük harf olması gerekir. CMQVERA tarafından belirlenen deęer, CMQVERA ' nın bir sonraki çağrılıřına ya da yapıbiriminin sonuna kadar varsayılan deęer olarak kalır. CMQVERA ' yı atlarsanız, varsayılan deęer DCLVER=CURRENT olur.

Gösterim kuralları

Diđer konular, çağrılarının nasıl çağrılacaęını ve parametrelerin nasıl bildirileceęini gösterir. Bazı durumlarda, deęiřtirgeler, bir sayısal deęiřmezi göstermek için küçük bir n harfe sahip, deęiřmez büyüklüklü diziler ya da karakter dizileridir. Bu deęiřtirgeye iliřkin bildirim kodladığınızda, n deęerini gereken sayısal deęerle deęiřtirin.

Yapılar

Yapılar, makronun iřlemine denetlemek için çeřitli parametreleri olan makrolar tarafından oluşturulur.

Not: Zaman zaman IBM MQ yapılarının yeni sürümleri tanıtılır. Yeni sürümdeki ek alanlar, daha önce 256 bayttan küçük olan bir yapının 256 bayttan büyük olmasına neden olabilir. Bu nedenle, bir IBM MQ yapısını kopyalamak ya da bir IBM MQ yapısını boş deęere ayarlamak üzere 256 bayttan büyük olabilecek yapılarla doęru řekilde çalışmak için çevirici yönergelerini yazın. Diđer bir seęenek olarak, yapının belirli bir sürümünü bildirmek için DCLVER makro deęiřtirgesini ya da CMQVERA makrosunu VERSION deęiřtirgesiyle birlikte kullanın.

- “Yapının adının belirtilmesi” sayfa 269
- “Yapının biçimini belirtme” sayfa 269
- “Yapının sürümünü denetleme” sayfa 270
- “Bir yapının başka bir yapının içine gömülmüş olarak bildirilmesi” sayfa 270
- “Alanlar için başlangıç deęerlerinin belirtilmesi” sayfa 270
- “Listelemeyi denetleme” sayfa 270

Yapının adının belirtilmesi

Bir yapının birden çok örneęini bildirmek için, makro yapıdaki her bir alanın adının başına kullanıcı tarafından belirtilen bir dizgi ve alt çizgi ekler.

Kullanılan dizgi, makronun çağrılmasında belirtilen etikettir. Etiket belirtilmezse, öneki oluşturmak için yapının adı kullanılır:

```
* Declare two object descriptors
      CMQODA ,          Prefix used="MQOD_" (the default)
MY_MQOD CMQODA ,          Prefix used="MY_MQOD_"
```

Bu kısımda gösterilen yapı bildirimleri varsayılan öneki kullanır.

Yapının biçimini belirtme

Yapı bildirimleri, makro tarafından DSECT deęiřtirgesiyle denetlenen iki biçimden birinde oluşturulabilir:

DSECT = EVET

Yeni bir veri kısmı başlatmak için çevirici DSECT yönergesi kullanılır; yapı tanımı DSECT deyimini hemen izler. Makro çağrısındaki etiket, veri kısmının adı olarak kullanılır; etiket belirtilmezse, yapının adı kullanılır.

DSECT = HAYIR

Çevirici DC yönergeleri, yapıyı yordamın yürürlükteki konumunda tanımlamak için kullanılır. Alanlar, makro çağrısında ilgili deęiřtirgelerin kodlanmasıyla belirtilebilecek deęerlerle kullanıma hazırlanır. Makro çağrısında deęer belirtilmeyen alanlar varsayılan deęerlerle kullanıma hazırlanır.

Belirtilen deęer büyük harfli olmalıdır. DSECT parametresi belirtilmezse, DSECT = NO varsayılır.

Yapının sürümünü denetleme

Varsayılan olarak, makrolar her zaman her yapının en son sürümünü bildirir.

Yapıdaki *Version* alanı için bir değer belirtmek üzere *VERSION* makro parametresini kullanabilirsiniz, ancak bu değiştirge *Version* alanının ilk değerini tanımlar ve gerçekte bildirilen yapının sürümünü denetlemez. Bildirilen yapının sürümünü denetlemek için *DCLVER* değiştirgesini kullanın:

DCLVER=YRRLKTEKI

Bildirilen sürüm güncel (en son) sürümdür.

DCLVER=BELIRTILDI

Bildirilen sürüm, *VERSION* değiştirgesiyle belirtilen sürümdür. *VERSION* değiştirgesini atlarsanız, varsayılan değer sürüm 1 'dir.

VERSION değiştirgesini belirtirseniz, değer kendi kendini tanımlayan bir sayısal değişmez ya da gereken sürüm için belirtilen değişmez olmalıdır (örneğin, *MQCNO_VERSION_3*). Başka bir değer belirtirseniz, *VERSION* değeri geçerli bir değere çözülsün bile, yapı *DCLVER=CURRENT* belirtilmiş gibi bildirilir.

Belirtilen değer büyük harfli olmalıdır. *DCLVER* değiştirgesini atlarsanız, kullanılan değer *MQDCLVER* genel makro değişkeninden alınır. Bu değişkeni *CMQVERA* makrosunu kullanarak ayarlayabilirsiniz.

Bir yapının başka bir yapının içine gömülmüş olarak bildirilmesi

Bir yapıyı başka bir yapının bileşeni olarak bildirmek için *IÇIÇE* değiştirgesini kullanın:

NESTED=EVET

Yapı bildirimini başka bir yapının içine yerleştirilmiş.

NESTED=HAYIR

Yapı bildirimini başka bir yapının içine yerleştirilmedi.

Belirtilen değer büyük harfli olmalıdır. *IÇIÇE* değiştirgesini atlarsanız, *NESTED=NO* varsayılır.

Alanlar için başlangıç değerlerinin belirtilmesi

Bir yapıdaki bir alanı kullanıma hazırlamak için kullanılacak değeri, ilgili alanın adını (önek olmadan) makro çağrısında parametre olarak, gerekli değerle birlikte kodlayarak belirtin.

Örneğin, *MQMT_REQUEST* ile kullanıma hazırlanan *MsgType* alanıyla bir ileti tanımlayıcı yapısını ve "MY_REPLY_TO_QUEUE" dizgisiyle kullanıma hazırlanan *ReplyToQ* alanını bildirmek için aşağıdakileri kullanın:

```
MY_MQMD CMQMDA MSGTYPE=MQMT_REQUEST, X
          REPLYTOQ=MY_REPLY_TO_QUEUE
```

Makro çağrısında bir değer olarak adlandırılmış bir değişmez (eşit) belirtirseniz, adı belirtilen değişmezi tanımlamak için *CMQA* makrosunu kullanın. Karakter dizgisi değerlerini tek tırnak içine almayın.

Listelemeyi denetleme

LIST değiştirgesini kullanarak çevirici listesinde yapı bildirimini görünüşünü denetleyin:

LIST = EVET

Yapı bildirimini, çevirici listesinde görünür.

LISTE = HAYIR

Yapı bildirimini, çevirici listesinde görünmüyor.

Belirtilen değer büyük harfli olmalıdır. *LIST* parametresini çıkarırsanız, *LIST = NO* varsayılır.

MQAIR-Kimlik doğrulama bilgileri kaydı

MQAIR yapısı, IBM MQ MQI client olarak çalışan bir uygulamanın istemci bağlantısı için kullanılacak bir kimlik doğrulayıcıya ilişkin bilgileri belirtmesini sağlar. Yapı, MQCONNX çağrısındaki bir giriş değiştirgesidir.

Kullanılabilirlik

MQAIR yapısı aşağıdaki istemciler için kullanılabilir:

-  AIX
-  Linux
-  Windows

Karakter kümesi ve kodlama

MQAIR içindeki veriler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır; bunlar **CodedCharSetId** kuyruk yöneticisi özniteliği ve MQENC_NATIVE ile verilir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 467. MQAIR içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQAIR_STRUC_ID	'AIR-'
<u>Sürüm</u> (yapı sürümü numarası)	MQAIR_VERSION_1	1
<u>AuthInfoTip</u> (kimlik doğrulama bilgilerinin tipi)	MQAIT_CRL_LDAP	1
<u>AuthInfoConnName</u> (LDAP CRL sunucusunun bağlantı adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>LDAPUserNamePtr</u> (LDAP kullanıcı adının adresi)	Yok	Boş değerli gösterge ya da boş değerli byte
<u>LDAPUserNameGörelili Konum</u> (LDAP kullanıcı adının MQSCO ' nun başlangıcından görelili konumu)	Yok	0
<u>LDAPUserNameLength</u> (LDAP kullanıcı adının uzunluğu)	Yok	0
<u>LDAPPassword</u> (LDAP sunucusuna erişim parolası)	Yok	Boş değerli dizilim ya da boşluklar
Not: <i>Sürüm</i> MQAIR_VERSION_2' den küçükse kalan alanlar yoksayılır.		
<u>OCSPResponderURL</u> (OCSP yanıtlayıcısıyla iletişim kurulabilecekURL)	Yok	Boş değerli dizilim ya da boşluklar

Çizelge 467. MQAIR içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
Notlar:		
1. ~ simgesi, tek bir boş karakteri temsil eder.		
2. C programlama dilinde, makro değişkeniMQAIR_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:		
<pre>MQAIR MyAIR = {MQAIR_DEFAULT};</pre>		

Dil bildirimleri

MQAIR için C bildirimi

```
typedef struct tagMQAIR MQAIR;
struct tagMQAIR {
    MQCHAR4    StrucId;           /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    MQLONG     AuthInfoType;     /* Type of authentication
                                information */
    MQCHAR264  AuthInfoConnName; /* Connection name of CRL LDAP
                                server */
    PMQCHAR    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 */
    MQCHAR32   LDAPPassword;     /* Password to access LDAP server */
    MQCHAR256  OCSPResponderURL; /* URL of OCSP responder */
};
```

MQAIR için COBOL bildirimi

```
** 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).
```

MQAIR için Visual Basic bildirimi

```
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
End Type

As String*32 'Password to access LDAP server'

MQAIR için StrucId (MQCHAR4)

Bu, kimlik doğrulama bilgileri kayıt yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQAIR_STRUC_ID.

Değer şöyle olmalıdır:

MQAIR_STRUC_ID

Kimlik doğrulama bilgileri kaydının tanıtıcısı.

C programlama dili için, MQAIR_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQAIR_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQAIR için Sürüm (MQLONG)

Bu, kimlik doğrulama bilgileri kayıt yapısının sürüm numarasıdır. Her zaman bir giriş alanıdır.

Değer aşağıdakilerden biri olmalıdır:

MQAIR_VERSION_1

Version-1 kimlik doğrulama bilgileri kaydı.

Bu, bu alanın ilk değeridir.

MQAIR_VERSION_2

Version-2 kimlik doğrulama bilgileri kaydı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQAIR_CURRENT_VERSION

Kimlik doğrulama bilgileri kaydının geçerli sürümü.

MQAIR için AuthInfoTipi (MQLONG)

Bu, kayıttaki kimlik doğrulama bilgilerinin tipidir.

Değer aşağıdaki iki parametreden biri olabilir:

MQAIT_CRL_LDAP

LDAP sunucusu kullanılarak sertifika iptal denetimi.

MQAIT_OCSP

OCSP kullanılarak sertifika iptal denetimi.

Değer geçerli değilse, çağrı MQRC_AUTH_INFO_TYPE_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri MQAIT_CRL_LDAP.

MQAIR için AuthInfoConnName (MQCHAR264)

Bu, LDAP sunucusunun çalıştığı anasistemin anasistem adı ya da ağ adresidir. Bunu, parantez içinde yer alan isteğe bağlı bir kapı numarası izleyebilir. Varsayılan kapı numarası 389'dur.

Değer alanın uzunluğundan kısaysa, değeri boş bir karakterle sonlandırın ya da alanın uzunluğuna kadar boşluklarla doldurun. Değer geçerli değilse, çağrı MQRC_AUTH_INFO_CONN_NAME_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın uzunluğu MQ_AUTH_INFO_CONN_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki boş karakterdir.

LDAPUserNameMQAIR için Ptr (PMQCHAR)

Bu, LDAP kullanıcı adıdır.

LDAP CRL sunucusuna erişmeye çalışan kullanıcının Ayırt Edici Adından oluşur. Değer *LDAPUserNameLength* ile belirtilen uzunluktan kısaysa, değeri boş bir karakterle sonlandırın ya da *LDAPUserNameLength* uzunluğuna kadar boşluklarla doldurun. *LDAPUserNameLength* sıfır alan yoksaılır.

LDAP kullanıcı adını şu iki yoldan biriyle sağlayabilirsiniz:

- *LDAPUserNamePtr* gösterge alanını kullanarak

Bu durumda, uygulama MQAIR yapısından ayrı bir dizgi bildirebilir ve *LDAPUserNamePtr* dizgisini dizginin adresine ayarlayabilir.

İşaretçi veri tipini destekleyen programlama dilleri için farklı ortamlarda (örneğin, C programlama dili) taşınabilir bir şekilde *LDAPUserNamePtr* ' i kullanmayı düşünün.

- *LDAPUserNameOffset* görelî konum alanını kullanarak

Bu durumda, uygulama MQSCO yapısını ve ardından MQAIR kayıtları dizisini ve ardından LDAP kullanıcı adı dizgilerini içeren bir bileşik yapıyı bildirmeli ve *LDAPUserNameOffset* dizgisini MQAIR yapısının başlangıcından itibaren uygun ad dizgisinin görelî konumuna ayarlamalıdır. Bu değerin doğru olduğundan ve MQLONG içinde tutulabilecek bir değere sahip olduğundan emin olun (en kısıtlayıcı programlama dili COBOL 'dur; geçerli aralık -999 999 999 ile 999 999 999 999 arasındadır).

Gösterge veri tipini desteklemeyen programlama dilleri ya da gösterge veri tipini farklı ortamlara (örneğin, COBOL programlama dili) taşınmayacak şekilde uygulayan programlama dilleri için *LDAPUserNameOffset* ' i kullanmayı düşünün.

Hangi teknik seçilirse seçilsin, *LDAPUserNamePtr* ve *LDAPUserNameOffset* yöntemlerinden yalnızca birini kullanın; Her ikisi de sıfır dışında bir değerse, çağrı MQRC_LDAP_USER_NAME_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizgisidir.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir.

LDAPUserNameMQAIR için görelî konum (MQLONG)

Bu, LDAP kullanıcı adının MQAIR yapısının başlangıcındaki bayt cinsinden görelî konumudur.

Görelî konum pozitif ya da negatif olabilir. *LDAPUserNameLength* sıfır alan yoksayılır.

LDAP kullanıcı adını belirtmek için *LDAPUserNamePtr* ya da *LDAPUserNameOffset* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız; ayrıntılar için *LDAPUserNamePtr* alanının tanımına bakın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

LDAPUserNameMQAIR için uzunluk (MQLONG)

Bu, *LDAPUserNamePtr* ya da *LDAPUserNameOffset* alanı tarafından adreslenen LDAP kullanıcı adının bayt cinsinden uzunluğudur.

Değer, MQ_AYIRT edici ad_adi_uzunluk ile sıfır aralığında olmalıdır. Değer geçerli değilse, çağrı MQRC_LDAP_USER_NAME_LENGTH_ERR neden koduyla başarısız olur.

Dahil olan LDAP sunucusu bir kullanıcı adı gerektirmiyorsa, bu alanı sıfır olarak ayarlayın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

MQAIR için LDAPPASSWORD (MQCHAR32)

Bu, LDAP CRL sunucusuna erişmek için gereken paroladır. Değer alanın uzunluğundan kısaysa, değeri boş bir karakterle sonlandırın ya da alanın uzunluğuna kadar boşluklarla doldurun.

LDAP sunucusu parola gerektirmiyorsa ya da LDAP kullanıcı adını atlarsanız, *LDAPPASSWORD* boş değerli ya da boş olmalıdır. LDAP kullanıcı adını atlarsanız ve *LDAPPASSWORD* boş değerli ya da boş değilse, çağrı MQRC_LDAP_PASSWORD_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın uzunluğu MQ_LDAP_PASSWORD_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki boş karakterdir.

MQAIR için OCSPResponderURL (MQCHAR256)

Bir OCSP yanıtlayıcısına ilişkin bağlantı ayrıntılarını gösteren bir MQAIR yapısı için bu alan, yanıt verenin iletişim kurabileceği URL ' yi içerir.

Bu alanın değeri HTTP URL' dir. Bu alan, AuthorityInfoAccess (AIA) sertifika uzantısındaki bir URL ' den önceliklidir.

Aşağıdaki deyimlerin her ikisi de true değilse değer yoksayılr:

- MQAIR yapısı Sürüm 2 ya da üstü (Sürüm alanı MQAIR_VERSION_2 ya da üstü olarak ayarlanır).
- AuthInfoTip alanı MQAIT_OCSP olarak ayarlandı.

Alan doğru biçimde bir HTTP URL içermiyorsa (ve yoksayılmıyorsa), MQCONNX çağrısı başarısız olur. Neden kodu: MQRC_OCSP_URL_ERROR.

Bu alan büyük ve küçük harfe duyarlıdır. Küçük harfli http:// dizgisiyle başlamalıdır. OCSP sunucusu somutlamasına bağlı olarak, URL ' nin geri kalanı büyük ve küçük harfe duyarlı olabilir.

Bu alan veri dönüştürmeye tabi değildir.

MQBMHO-İleti işleme seçenekleri için arabellek

MQBMHO yapısı, uygulamaların, ileti tanıtıcılarının arabelleklerden nasıl üretildiğini denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQBUFMH çağrısındaki bir giriş deęiştirgesidir.

Karakter kümesi ve kodlama

MQBMHO içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında olmalıdır (MQENC_NATIVE).

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 468. MQBMHO içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQBMHO_STRUC_ID	'BMHO'
Sürüm (yapı sürümü numarası)	MQBMHO_VERSION_1	1
Seçenekler (MQBMHO işlemini denetleyen seçenekler)	MQBMHO_NONE	0

Notlar:

1. C programlama dilinde, makro deęişkeniMQBMHO_DEFAULT, çizelgede listelenen deęerleri içerir. Yapıdaki alanlar için başlangıç deęerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQBMHO MyBMHO = {MQBMHO_DEFAULT};
```

Dil bildirimleri

MQBMHO için C bildirimi

```
typedef struct tagMQBMHO MQBMHO;
struct tagMQBMHO {
    MQCHAR4 StrucId;          /* Structure identifier */
    MQLONG  Version;         /* Structure version number */
    MQLONG  Options;        /* Options that control the action of
```

```
}; MQBUFMH */
```

MQBMHO için COBOL bildirimi

```
** 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.
```

MQBMHO için PL/I bildirimi

```
Dcl
1 MQBMHO 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 MQBUFMH */
```

MQBMHO için High Level Assembler bildirimi

```
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)
```

MQBMHO için StrucId (MQCHAR4)

Bu, ileti işleme yapısına ilişkin arabelleğin yapı tanıtcısıdır. Her zaman bir giriş alanıdır. Değeri MQBMHO_STRUC_ID.

Değer şöyle olmalıdır:

MQBMHO_STRUC_ID

İleti işleme yapısına ilişkin arabellek tanıtıcısı.

C programlama dili için, MQBMHO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQBMHO_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQBMHO için MQBMHO Sürümü (MQLONG)

Bu, ileti işleme yapısına ilişkin arabelleğin sürüm numarasıdır. Her zaman bir giriş alanıdır.

Değer şöyle olmalıdır:

MQBMHO_VERSION_1

İleti işleme yapısına ilişkin arabelleğin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQBMHO_CURRENT_VERSION

İleti işleme yapısına ilişkin arabelleğin yürürlükteki sürümü.

MQBMHO için seçenekler (MQLONG)

İleti işleyecek arabellek yapısı-Seçenekler alanı

Değer şöyle olabilir:

MQBMHO_DELETE_PROPERTIES

İleti tanıtıcısına eklenen özellikler arabellekten silinir. Arama başarısız olursa, hiçbir özellik silinmez.

Varsayılan seçenekler: Açıklanan seçeneğe gerek duymuyorsanız, aşağıdaki seçeneği kullanın:

MQBMHO_NONE

Seçenek belirtilmedi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQBMHO_DELETE_PROPERTIES.

V 9.3.0 MQBNO-Dengeleme seçenekleri

Aşağıdaki tablo, yapıdaki alanları özetler.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 469. MQBNO içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQBNO_STRUC_ID	'BNO-'
Sürüm (yapı sürümü numarası)	MQBNO_VERSION_1	1
ApplicationType (yapıda ayarlanan dengeleme seçeneğinin tipi)	MQBNO_VALTYPE_SIMPLE	0
Zamanaşımı (yeniden dengeleme, uygulama etkinliğini kesintiye uğratabilir)	MQBNO_TIMEOUT_AS_DEFAULT	0
BalanceOptions (düzenleyen uygulama tarafından ayarlanan dengeleme seçenekleri)	MQBNO_OPTIONS_NONE	0

Notlar:

- simgesi, tek bir boş karakteri temsil eder.
- C programlama dilinde, MQBNO_DEFAULT makro değişkeni çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQBNO MyBNO = {MQBNO_DEFAULT};
```

Dil bildirimleri

MQBNO için C bildirimi

```
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 */
};
```

MQBNO için COBOL bildirimi

```
** 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.

```

MQBNO için PL/I bildirimi

```

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
                                interrupt application activity */
  3 BalanceOptions  fixed bin(31), /* Balancing options set by the issuing
                                application*/

```

İlgili başvurular

“MQCNO-Bağlantı seçenekleri” sayfa 316

MQCNO yapısı, uygulamanın kuyruk yöneticisine yönelik bağlantıyla ilgili seçenekleri belirtmesine olanak sağlar. Yapı, MQCONNX çağrısındaki bir giriş/çıkış deęiřtirgesidir.

V 9.3.0 MQBNO için StrucId (MQCHAR4)

Bu, dengeleme seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. İlk deęeri BNO.

Deęer řöyle olmalıdır:

BNO

Dengeleme seçenekleri yapısına iliřkin tanıtıcı.

C programlama dili için, MQBNO_STRUC_ID_ARRAY deęiřmezi de tanımlanır. Bu deęiřmezin deęeri BNO ile aynı, ancak bir dizgi yerine bir karakter dizisi.

StrucId için geęerli bir deęer saęlamalısınız ya da MQRC_BNO_ERROR döndürüldü.

V 9.3.0 MQBNO için Sürüm (MQLONG)

Bu, dengeleme seçenekleri yapısının sürüm numarasıdır. Her zaman bir giriş alanıdır.

Deęer řöyle olmalıdır:

MQBNO_VERSION_1

Dengeleme seçenekleri yapısına iliřkin sürüm numarası.

Version için geęerli bir deęer saęlamalısınız ya da MQRC_BNO_ERROR döndürüldü.

V 9.3.0 MQBNO için ApplicationType (MQLONG)

Yapıda ayarlanan dengeleme seçeneęinin tipi.

Olası deęerler řunlardır:

MQBNO_BALTYPE_SIMPLE

Basit dengeleme; Tek biçimli kümelerde uygulama yeniden dengelemesi konusunda anlatılanlara ek olarak belirli bir kural uygulanmaz.

MQBNO_BALTYPE_REQREP

İstek-Yanıt dengeleme; her MQPUT çağrısından sonra, yanıt iletisi için eşleşen bir MQGET çağrısı beklenir. Böyle bir ileti alınıncaya ya da istek iletisi EXPIRY aşılıncaya kadar dengeleme geciktirildi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk deęeri MQBNO_BALTYPE_SIMPLE.

ApplicationType alanı için tek bir deęer saęlamalısınız ya da MQRC_BNO_ERROR döndürülür.

Not: Bu MQBNO_BALTYPE_RA_MANAGED alanı için ek bir değer, IBM MQ Resource Adapter for JEE ortamları tarafından kullanılmak üzere ayrılmıştır. Bir uygulamanın bu değeri doğrudan sağlaması bir hata olsa da, örneğin, uygulama durumu sorgulanırken bildirilebilir.

V 9.3.0 MQBNO için zamanaşımı (MQLONG)

Yeniden dengeleme işleminden sonra **Timeout** uygulama etkinliğini kesebilir.

Olası değerler şunlardır:

MQBNO_TIMEOUT_AS_DEFAULT

Ayarlanan varsayılan zamanaşımı değeri.

MQBNO_TIMEOUT_IMMEDIATE

Anında zamanaşımı oluşur.

MQBNO_TIMEOUT_ASLA

Zamanaşımı oluşmaz.

Bu alanın ilk değeri MQBNO_TIMEOUT_AS_DEFAULT.

Timeout alanı ya da MQRC_BNO_ERROR döndürüldüğü için, yalnızca tanımlı değerlerden ya da 0-999999999 saniye arasında bir değer girmeniz gerekir.

V 9.3.0 MQBNO için BalanceOptions (MQLONG)

Düzenleyen uygulama tarafından ayarlanan dengeleme seçenekleri.

Olası değerler şunlardır:

MQBNO_OPTIONS_NONE

Seçenek belirlenmedi

MQBNO_OPTIONS_IGNORE_TRANS

Bu seçeneğin ayarlanması, bir hareketin ortasında bile uygulamaların yeniden dengelenmesini sağlar.

Bu alanın ilk değeri MQBNO_OPTIONS_NONE.

BalanceOptions alanı için mantıksal ya da karakter kullanarak tanımlı değerlerin herhangi bir birleşimini sağlayabilirsiniz. Geçerli olmayan değerler bir MQRC_BNO_ERROR döndürülmesine neden olur.

MQBO-Başlangıç seçenekleri

MQBO yapısı, uygulamanın bir iş birimi yaratılmasıyla ilgili seçenekleri belirtmesine olanak sağlar. Yapı, MQBEGIN çağrısında bir giriş/çıkış değiştirgesidir.

Kullanılabilirlik

MQBO yapısı aşağıdaki altyapılarda bulunur:

-  AIX
-  IBM i
-  Linux
-  Windows

IBM MQ MQI clients için MQBO yapısı kullanılmıyor.

Karakter kümesi ve kodlama

MQBO 'daki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 470. MQBO için MQBO 'daki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQBO_STRUC_ID	'BO- -'
Sürüm (yapı sürümü numarası)	MQBO_VERSION_1	1
Seçenekler (MQBEGIN işlemini denetleyen seçenekler)	MQBO_NONE	0

Notlar:

- simgesi, tek bir boş karakteri temsil eder.
- C programlama dilinde, makro değişkeniMQBO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQBO MyBO = {MQBO_DEFAULT};
```

Dil bildirimleri

MQBO için C bildirimi

```
typedef struct tagMQBO MQBO;
struct tagMQBO {
    MQCHAR4 StrucId; /* Structure identifier */
    MQLONG Version; /* Structure version number */
    MQLONG Options; /* Options that control the action of MQBEGIN */
};
```

MQBO için COBOL bildirimi

```
** 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.
```

MQBO için PL/I bildirimi

```
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 */
```

MQBO için Visual Basic bildirimi

```
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
```


MQBO için StrucId (MQCHAR4)

Bu, başlangıç seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQBO_STRUC_ID.

Değer şöyle olmalıdır:

MQBO_STRUC_ID

Başlangıç seçenekleri yapısına ilişkin tanıtıcı.

C programlama dili için, MQBO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQBO_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQBO için Sürüm (MQLONG)

Bu, başlangıç seçenekleri yapısının sürüm numarasıdır. Her zaman bir giriş alanıdır.

Değer şöyle olmalıdır:

MQBO_VERSION_1

Başlangıç seçenekleri yapısının sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQBO_CURRENT_VERSION

Başlangıç seçenekleri yapısının yürürlükteki sürümü.

MQBO için seçenekler (MQLONG)

Bu alan her zaman bir giriş alanıdır. İlk değeri MQBO_NONE.

Değer şöyle olmalıdır:

MQBO_NONE

Seçenek belirtilmedi.

MQCBC-Geri çağırma bağlamı

MQCBC yapısı, bir geri çağırma işlevine geçirilen bağlam bilgilerini belirtmek için kullanılır. Yapı, bir ileti tüketicisi yordamına yapılan çağrıdaki bir giriş/çıkış değiştirgesidir.

Kullanılabilirlik

MQCBC yapısı aşağıdaki altyapılarda bulunur:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

ve bu sistemlere bağlı IBM MQ MQI clients için.

Sürüm

MQCBC ' nin yürürlükteki sürümü MQCBC_VERSION_2.

Karakter kümesi ve kodlama

MQCBC içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olur.

Alanlar

MQCBC yapısı için başlangıç değeri yoktur. Yapı, bir geri çağırma yordamına parametre olarak geçirilir. Kuyruk yöneticisi yapıyı kullanıma hazırlar; uygulamalar yapıyı hiçbir zaman kullanıma hazırlamaz.

Notlar:

- Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.
- MQCBC yapısı için başlangıç değeri yok. Yapı, bir geri çağırma yordamına parametre olarak geçirilir. Kuyruk yöneticisi yapıyı kullanıma hazırlar; uygulamalar yapıyı hiçbir zaman kullanıma hazırlamaz.

Çizelge 471. MQCBC içindeki alanlar	
Alan	Açıklama
<u>StrucID</u>	Yapı tanıtıcısı
<u>Sürüm</u>	Yapı sürümü numarası
<u>CallType</u>	İşlev neden çağrıldı
<u>Hobj</u>	Nesne tanıtıcısı
<u>CallbackArea</u>	Geri çağırma işlevinin kullanılacağı alan
<u>ConnectionArea</u>	Geri çağırma işlevinin kullanılacağı alan
<u>CompCode</u>	Tamamlanma kodu
<u>Neden</u>	Neden Kodu
<u>STATE</u>	Geçerli tüketicinin durumunun göstergesi
<u>DataLength</u>	İleti uzunluğu
<u>BufferLength</u>	İleti arabelleğinin bayt cinsinden uzunluğu
<u>İşaretler</u>	Genel işaretler
Not: Sürüm MQCBC_VERSION_2 değerinden küçükse, geri kalan alan yoksayılır	
<u>ReconnectDelay</u>	Yeniden bağlanma girişiminden önceki milisaniye sayısı

Dil bildirimleri

MQCBC için C bildirimi

```
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 */
```

MQCBC için COBOL bildirimi

```
** 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 **
```

MQCBC için PL/I bildirimi

```
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 */
```

MQCBC için High Level Assembler (Üst Düzey Çevirici) bildirimi

```
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)
```

MQCBC için StrucId (MQCHAR4)

Bu, geri çağırma bağlamı yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQCBC_STRUC_ID.

Değer şöyle olmalıdır:

MQCBC_STRUC_ID

Geri çağırma bağlamı yapısına ilişkin tanıttıcı.

C programlama dili için, değişmez MQCBC_STRUC_ID_ARRAY da tanımlanır. Bu değer MQCBC_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQCBC için Sürüm (MQLONG)

Bu, geri çağırma bağlamı yapısının sürüm numarasıdır. Her zaman bir giriş alanıdır.

Değer şöyle olmalıdır:

MQCBC_VERSION_1

Sürüm 1 geri çağırma bağlamı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCBC_CURRENT_VERSION

Geri çağırma bağlamı yapısının yürürlükteki sürümü.

Geri çağırma işlevi her zaman yapının en son sürümünü geçirilir.

MQCBC için CallType (MQLONG)

Bu işlevin çağrılmasıyla ilgili bilgileri içeren alan; aşağıdaki değerler tanımlanır.

İleti teslim çağrısı tipleri: Bu çağrı tipleri bir iletiyle ilgili bilgi içerir. **DataLength** ve **BufferLength** parametreleri bu çağrı tipleri için geçerlidir.

MQCBCT_MSG_REMOVED

İleti tüketicisi işlevi, nesne tanıttıcısından yıkıcı bir şekilde kaldırılmış bir iletiyle çağrıldı.

CompCode değeri MQCC_WARNING ise, *Reason* alanının değeri MQRC_TRUNCATED_MSG_ACCEPTED ya da bir veri dönüştürme sorununu gösteren kodlardan biridir.

MQCBCT_MSG_NOT_REMOVED

İleti tüketici işlevi, henüz nesne tanıttıcısından yıkıcı bir şekilde kaldırılmamış bir iletiyle çağrıldı. İleti, *MsgToken* kullanılarak nesne tanıttıcısından yıkıcı bir şekilde kaldırılabilir.

İleti aşağıdaki nedenden ötürü kaldırılmamış olabilir:

- MQGMO seçenekleri bir göz atma işlemi istedi, MQGMO_BROWSE_*
- İleti kullanılabilir arabellekten büyük ve MQGMO seçenekleri MQGMO_ACCEPT_TRUNCATED_MSG belirtmiyor

CompCode değeri MQCC_WARNING ise, *Reason* alanının değeri MQRC_TRUNCATED_MSG_FAILED ya da veri dönüştürme sorununu gösteren kodlardan biridir.

Geri çağırma denetimi arama tipleri: Bu çağrı tipleri, geri çağırma denetimine ilişkin bilgileri içerir ve bir iletiyle ilgili ayrıntıları içermez. Bu çağrı tipleri, MQCBD yapısında Seçenekler kullanılarak isteniyor.

DataLength ve **BufferLength** parametreleri bu çağrı tipleri için geçerli değildir.

MQCBCT_REGISTER_CALL

Bu çağrı tipinin amacı, geri çağırma işlevinin bazı ilk ayarları gerçekleştirmesine izin vermektir.

Geri çağırma işlevi, geri çağırma kaydedildikten hemen sonra (yani, MQOP_REGISTER ' in *Operation* alanı için bir değer kullanılarak bir MQCB çağrısından döndükten sonra) çağrılır.

Bu çağrı tipi hem ileti tüketicileri hem de olay işleyiciler için kullanılır.

İstenirse, bu, geri çağırma işlevinin ilk çağrısıdır.

Reason alanının deęeri MQRC_NONE.

MQCBCT_START_CALL

Bu çağrı tipinin amacı, geri çağırma işlevinin başlatıldığında bazı ayarları gerçekleştirmesine izin vermektir; örneğin, önceden durdurulduğunda temizlenen kaynakları yeniden sağlamak.

Geri çağırma işlevi, bağlantı MQOP_START ya da MQOP_START_WAIT kullanılarak başlatıldığında çağrılır.

Bir geri çağırma işlevi başka bir geri çağırma işlevinde kayıtlıysa, geri çağırma döndüğünde bu çağrı tipi çağrılır.

Bu çağrı tipi yalnızca ileti tüketicileri için kullanılır.

Reason alanının deęeri MQRC_NONE.

MQCBCT_STOP_CALL

Bu çağrı tipinin amacı, geri çağırma işlevinin bir süre durdurulduğunda (örneğin, iletilerin tüketimi sırasında edinilen ek kaynakların temizlenmesi) bir temizleme işlemi gerçekleştirmesini sağlamaktır.

MQOP_STOP ' un *Operation* alanı için bir deęer kullanılarak bir MQCTL çağrısı verildiğinde geri çağırma işlevi çağrılır.

Bu çağrı tipi yalnızca ileti tüketicileri için kullanılır.

Reason alanının deęeri, durdurma nedenini gösterecek şekilde ayarlanır.

MQCBCT_DEREGISTER_CALL

Bu çağrı tipinin amacı, geri çağırma işlevinin, tüketme işleminin sonunda son temizlemeyi gerçekleştirmesine izin vermektir. Geri çağırma işlevi aşağıdaki durumda çağrılır:

- MQOP_DEREGISTER ile bir MQCB çağrısı kullanılarak geri çağırma işlevinin kaydı kaldırılır.
- Kuyruk kapatıldı, örtük bir kayıttan çıkarıcıya neden oluyor. Bu yönetim ortamında, geri çağırma işlevine nesne tanıtıcısı olarak MQHO_UNUSABLE_HOBJ geçirilir.
- MQDISC çağrısı tamamlandı-örtük bir kapatmaya ve dolayısıyla kayıttan kaldırılmasına neden oldu. Bu durumda, bağlantı hemen kesilmez ve devam eden hareketler henüz kesinleştirilmez.

Geri çağırma işlevinin kendisinde bu işlemlerden herhangi biri gerçekleştirilirse, geri çağırma işlemi geri döndürdükten sonra işlem çağrılır.

Bu çağrı tipi hem ileti tüketicileri hem de olay işleyiciler için kullanılır.

İstenirse, bu, geri çağırma işlevinin son çağrısıdır.

Reason alanının deęeri, durdurma nedenini gösterecek şekilde ayarlanır.

MQCBCT_EVENT_CALL

Olay işleyici işlevi

Kuyruk yöneticisi ya da bağlantı durduğunda ya da susturulduğunda olay işleyici işlevi ileti olmadan çağrıldı.

Bu çağrı, tüm geri çağırma işlevleri için uygun işlemi gerçekleştirmek üzere kullanılabilir.

İleti tüketici işlevi

Nesne tanıtıcısına özgü bir hata (*CompCode* = MQCC_FAILED) saptandığında ileti tüketici işlevi bir ileti olmadan çağrıldı; örneğin, *Reason* code = MQRC_GET_INHIBE.

Reason alanının deęeri, aramanın nedenini gösterecek şekilde ayarlanır.

MQCBCT_MC_EVENT_CALL

Çok hedefli olaylar için olay işleyici işlevi çağrıldı; Olay işleyiciye 'normal' IBM MQ olaylar yerine IBM MQ Çok hedefli olaylar gönderildi.

MQCBCT_MC_EVENT_CALL ile ilgili ek bilgi için [Multicast exception reporting](#) başlıklı konuya bakın.

MQCBC için Hobj (MQHOBj)

Bu, ileti tüketicisine yapılan çağrıların nesne tanıtıcısıdır.

Bir olay işleyici için bu değer MQHO_NONE değeridir.

Uygulama, kuyruktan bir ileti kaldırılmamışsa, iletiyi almak için İleti Alma Seçenekleri bloğundaki ileti simgesini ve bu tanıtıcıyı kullanabilir.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri: MQHO_UNUSABLE_HOBj

MQCBC için CallbackArea (MQPTR)

Bu alan, geri çağırma işlevinin kullanması için kullanılabilir.

Kuyruk yöneticisi bu alanın içeriğine dayalı olarak karar vermez ve geri çağırma işlevini tanımlamak için kullanılan MQCB çağrısında bir parametre olan MQCBD yapısındaki CallbackArea alanından değiştirilmeden geçirilir.

CallbackArea üzerinde yapılan değişiklikler, bir *Hobj* için geri çağırma işlevinin çağrılmaları boyunca korunur. Bu alan, diğer tutamaçlar için geri çağırma işlevleriyle paylaşılmaz.

Bu, geri çağırma işlevine ilişkin bir giriş/çıkış alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

MQCBC için ConnectionArea (MQPTR)

Bu alan, geri çağırma işlevinin kullanması için kullanılabilir.

Kuyruk yöneticisi bu alanın içeriğine dayalı olarak karar vermez ve geri çağırma işlevini denetlemek için kullanılan MQCTL çağrısındaki bir parametre olan MQCTLO yapısındaki ConnectionArea alanından değiştirilmeden geçirilir.

Geri çağırma işlevleri tarafından bu alanda yapılan değişiklikler, geri çağırma işlevinin çağrılmaları boyunca korunur. Bu alan, tüm geri çağırma işlevleri tarafından paylaşılacak bilgileri iletmek için kullanılabilir. *CallbackArea*' in tersine, bu alan bir bağlantı tanıtıcısı için tüm geri çağırma arasında yaygındır.

Bu bir giriş ve çıkış alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

MQCBC için CompCode (MQLONG)

Bu alan, tamamlanma kodudur. Bu, iletiyi tüketen herhangi bir sorun olup olmadığını gösterir.

Değer aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı

MQCC_UYARISI

Uyarı (kısmi tamamlama)

MQCC_FAILED

Arama başarısız oldu

Bu bir giriş alanıdır. Bu alanın ilk değeri MQCC_OK.

MQCBC için neden (MQLONG)

Bu, *CompCode*' i niteleyen neden kodudur.

Bu bir giriş alanıdır. Bu alanın ilk değeri MQRC_NONE.

MQCBC için Durum (MQLONG)

Geçerli tüketicinin durumuna ilişkin bir gösterge. Bu alan, tüketici işlevine sıfır dışında bir neden kodu iletildiğinde bir uygulamaya en çok değer verir.

Her neden kodu için davranışı kodlamanız gerekmediğinden uygulama programlamayı basitleştirmek için bu alanı kullanabilirsiniz.

Bu bir giriş alanıdır. Bu alanın ilk değeri MQCS_NONE.

Çizelge 472.		
İl	Kuyruk yöneticisi işlemi	Değişmezin değeri
<i>MQCS_NONE</i> Bu neden kodu, ek neden bilgisi olmadan olağan bir çağrıyı gösterir	Yok, bu normal bir işlem.	0
<i>MQCS_SUSPENDED_TEMPORARY</i> Bu neden kodları geçici koşulları gösterir.	Koşulu bildirmek için geri çağırma yordamı çağrılır ve sonra askıya alınır. Belirli bir süre sonra sistem işlemi yeniden deneyebilir, bu da aynı koşulun yeniden oluşmasına neden olabilir.	1
<i>MQCS_SUSPENDED_USER_ACTION</i> Bu neden kodları, geri çağırmanın koşulu çözmek için işlem yapması gereken durumları gösterir.	Tüketici askıya alındı ve koşulu bildirmek için geri çağırma yordamı çağrıldı. Geri çağırma yordamı olanağı varsa koşulu çözmeli ve RESUME ya da bağlantıyı kapatmalıdır.	2
<i>MQCS_SUSPENDED</i> Bu neden kodları, ileti geri çağırılmalarını önleyen başarısızlıkları gösterir.	Kuyruk yöneticisi, geri çağırma işlevini otomatik olarak askıya alır. Geri çağırma işlevi sürdürülürse, aynı neden kodunu yeniden alma olasılığı yüksektir.	3
<i>MQCS_STOPPED</i> Bu neden kodları, ileti tüketiminin sonunu gösterir.	Kural dışı durum işleyicisine ve belirtilen MQCBDO_STOP_CALL geri çağırma teslim edildi. Başka ileti tüketilemez.	4

MQCBC için DataLength (MQLONG)

Bu, iletideki uygulama verilerinin bayt cinsinden uzunluğudur. Değer sıfırsa, iletinin uygulama verisi içermediği anlamına gelir.

DataLength alanı iletinin uzunluğunu içerir, ancak tüketiciye iletilen ileti verilerinin uzunluğunu içermez. İleti kesilmiş olabilir. Tüketicide gerçekte ne kadar veri aktarıldığını belirlemek için MQGMO 'daki ReturnedLength alanını kullanın.

Neden kodu iletinin kesildiğini gösteriyorsa, gerçek iletinin ne kadar büyük olduğunu saptamak için DataLength alanını kullanabilirsiniz. Bu, ileti verilerini yerleştirmek için gereken arabellek büyüklüğünü belirlemenizi ve daha sonra, MaxMsgLength değerini uygun bir değerle güncellemek için bir MQCB çağrısı yapmanızı sağlar.

MQGMO_CONVERT seçeneği belirtilirse, dönüştürülen ileti DataLength için döndürülen değerden büyük olabilir. Bu tür durumlarda, uygulamanın MaxMsgLength değerini DataLengthkuyruk yöneticisinin döndürdüğü değerden büyük olacak şekilde güncellemek için bir MQCB çağrısı yapması gerekir.

İleti kesme sorunlarını önlemek için, MQCBD_FULL_MSG_LENGTH olarak MaxMsgUzunluğu belirtin. Bu, kuyruk yöneticisinin veri dönüştürme işleminden sonra tam ileti uzunluğu için bir arabellek ayırmasına neden olur. Ancak, bu seçenek belirtilse bile, isteği doğru olarak işlemek için yeterli saklama alanı bulunmadığını unutmayın. Uygulamalar her zaman döndürülen neden kodunu kontrol etmelidir.

Örneğin, iletiyi dönüştürmek için yeterli bellek ayrılmadıysa, iletiler uygulamaya dönüştürülmemiş olarak döndürülür.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; bir olay işleyici işleviyle ilgili değildir.

MQCBC için BufferLength (MQLONG)

Bu alan, bu işleve geçirilen ileti arabelleğinin bayt cinsinden uzunluğudur.

Arabellek, tüketici için tanımlanan MaxMsgUzunluğu değerinden ve MQGMO ' da ReturnedLength değerinden büyük olabilir.

Gerçek ileti uzunluğu DataLength alanında sağlanır.

Uygulama, geri çağırma işlevi süresince tüm arabelleği kendi amaçları için kullanabilir.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; kural dışı durum işleyici işleviyle ilgili değildir.

MQCBC için İşaretler (MQLONG)

Bu tüketici hakkında bilgi içeren işaretler.

Aşağıdaki seçenek tanımlandı:

MQCBCF_READA_BUFFER_EMPTY

MQCO_QUIESCE seçeneğini kullanan önceki bir MQCLOSE çağrısı MQRC_READ_AHEAD_MSGS neden koduyla başarısız olursa bu işaret döndürülebilir.

Bu kod, son önden okuma iletinin döndürüldüğünü ve arabelleğin artık boş olduğunu belirtti. Uygulama MQCO_QUIESCE seçeneğini kullanarak başka bir MQCLOSE çağrısı yaparsa başarılı olur.

Önden okuma arabelleğinde yürürlükteki seçim ölçütleriyle eşleşmeyen iletiler olabileceği için, bir uygulamaya bu işaret kümesiyle bir ileti verileceği garanti edilmez. Bu örnekte, tüketici işlevi MQRC_HOBJ_QUIESCED neden koduyla çağrılır.

Önden okuma arabelleği tamamen boşsa, tüketici MQCBCF_READA_BUFFER_EMPTY işareti ve neden kodu MQRC_HOBJ_QUIESCED_NO_MSGS ile çağrılır.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; bir olay işleyici işleviyle ilgili değildir.

MQCBC için ReconnectDelay (MQLONG)

ReconnectDelay , kuyruk yöneticisinin yeniden bağlanmayı denemeden önce ne kadar bekleyeceğini gösterir. Alan, gecikmeyi değiştirmek ya da yeniden bağlantıyı tamamen durdurmak için bir olay işleyici tarafından değiştirilebilir.

Geri Çağırma Bağlamındaki Neden alanının değeri MQRC_RECONNECTINGise ReconnectDelay alanını kullanın.

Olay işleyiciye girişte ReconnectDelay değeri, kuyruk yöneticisinin yeniden bağlantı girişiminde bulunmadan önce bekleyeceği milisaniye sayısıdır. Çizelge 473 sayfa 288 içinde, olay işleyiciden dönüştürme kuyruk yöneticisinin davranışını değiştirmek için ayarlayabileceğiniz değerler listelenir.

<i>Çizelge 473. ReconnectDelay değerleri</i>		
Ad	Değer	Açıklama
MQRD_NO_RECONNECT	-1	Yeniden bağlanma girişiminde bulunma. Uygulamaya bir hata döndürüldü.
MQRD_NO_DELAY	0	Hemen yeniden bağlanmayı deneyin.
<i>Milliseconds</i>	>0	Bağlantıyı yeniden denemeden önce bu kadar milisaniye bekleyin.

MQCBD-Geri çağırma tanımlayıcısı

MQCBD yapısı, bir geri çağırma işlevini ve kuyruk yöneticisi tarafından kullanımını denetleyen seçenekleri belirtmek için kullanılır. Yapı, MQCB çağırısındaki bir giriş deęiştirgesidir.

Kullanılabilirlik

MQCBD yapısı ařaęıdaki altyapılarda kullanılabilir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

ve bu sistemlere baęlı IBM MQ MQI clients için.

Sürüm

MQCBD ' nin yürürlükteki sürümü MQCBD_VERSION_1.

Karakter kümesi ve kodlama

MQCBD içindeki veriler, **CodedCharSetId** kuyruk yöneticisi öznitelięi tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında olmalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Alanlar

Not: Ařaęıdaki tabloda, alanlar alfabetik olarak deęil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 474. MQCBD ' deki Alanlar		
Alan adı ve açıklaması	Deęişmezin adı	Deęişmezin ilk deęeri (varsa)
<u>StrucID</u> (yapı tanıtıcısı)	MQCBD_STRUC_ID	' CBD- '
<u>Sürüm</u> (yapı sürümü numarası)	MQCBD_VERSION_1	1
<u>CallbackType</u> (geri çağırma işlevinin tipi)	MQCBT_MESSAGE_CONSUMER	1
<u>Seçenekler</u> (ileti tüketimini denetleyen seçenekler)	MQCBDO_NONE	0
<u>CallbackArea</u> (geri çağırma işlevinin kullanacağı alan)	Yok	Boş deęerli gösterge ya da boş deęerli boşluklar
<u>CallbackFunction</u> (işlevin bir API çağırısı olarak çağırılıp çağırılmayacağı)	Yok	Boş deęerli gösterge ya da boş deęerli boşluklar
<u>CallbackName</u> (işlevin dinamik olarak baęlı bir program olarak çağırılıp çağırılmayacağı)	Yok	Boş deęerli dizilim ya da boşluklar
<u>MaxMsgLength</u> (okunabilen en uzun iletinin uzunluęu)	MQCBD_FULL_MSG_LENGTH	-1

Çizelge 474. MÖCBDB 'deki Alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
Notlar:		
<p>1. ~ simgesi, tek bir boş karakteri temsil eder.</p> <p>2. Boş değeri dizim ya da boşluklar değeri, C programlama dilindeki boş değeri dizgiyi ve diğeri programlama dillerindeki boş karakterleri gösterir.</p> <p>3. C programlama dilinde, makro değeri MÖCBDB_DEFAULT, çizelgede listelenen değeri içerir. Yapıdaki alanlar için başlangıç değeri sağlamak üzere bunu aşağıdaki şekilde kullanın:</p>		
<pre>MÖCBDB MyCBD = {MÖCBDB_DEFAULT};</pre>		

Dil bildirimleri

MÖCBDB için C bildirimleri

```
typedef struct tagMÖCBDB MÖCBDB;
struct tagMÖCBDB {
    MÖCHAR4    StrucId;           /* Structure identifier */
    MÖLONG     Version;          /* Structure version number */
    MÖLONG     CallBackType;     /* Callback function type */
    MÖLONG     Options;          /* Options controlling message
    consumption */
    MÖPTR      CallbackArea;     /* User data passed to the function */
    MÖPTR      CallbackFunction; /* Callback function pointer */
    MÖCHAR128  CallbackName;     /* Callback name */
    MÖLONG     MaxMsgLength;     /* Maximum message length */
};
```

MÖCBDB için COBOL bildirimleri

```
** MÖCBDB structure
10 MÖCBDB.
** Structure Identifier
15 MÖCBDB-STRUCID                PIC X(4).
** Structure Version
15 MÖCBDB-VERSION                PIC S9(9) BINARY.
** Callback Type
15 MÖCBDB-CALLBACKTYPE          PIC S9(9) BINARY.
** Options
15 MÖCBDB-OPTIONS                PIC S9(9) BINARY.
** Callback User Area
15 MÖCBDB-CALLBACKAREA          POINTER
** Callback Function Pointer
15 MÖCBDB-CALLBACKFUNCTION      FUNCTION-POINTER
** Callback Program Name
15 MÖCBDB-CALLBACKNAME          PIC X(128)
** Maximum Message Length
15 MÖCBDB-MAXMSGLENGTH          PIC S9(9) BINARY.
```

MÖCBDB için PL/I bildirimleri

```
dcl
1 MÖCBDB based,
3 StrucId          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 */
```

MQCBD için StrucId (MQCHAR4)

Bu, geri çağırma tanımlayıcısı yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQCBD_STRUC_ID.

Değer şöyle olmalıdır:

MQCBD_STRUC_ID

Geri çağırma tanımlayıcısı yapısına ilişkin tanıttıcı.

C programlama dili için, değişmez MQCBD_STRUC_ID_ARRAY da tanımlanır. Bu değer MQCBD_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQCBD için Sürüm (MQLONG)

Bu, geri çağırma tanımlayıcısı yapısının sürüm numarasıdır. Her zaman bir giriş alanıdır.

Değer şöyle olmalıdır:

MQCBD_VERSION_1

Sürüm 1 geri çağırma tanımlayıcısı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCBD_CURRENT_VERSION

Geri çağırma tanımlayıcısı yapısının yürürlükteki sürümü.

MQCBD için CallbackType (MQLONG)

Geri çağırma tanımlayıcısı yapısı- CallbackType alanı

Bu, geri çağırma işlevinin tipidir. Değer şunlardan biri olmalıdır:

MQCBT_MESSAGE_CONSUMER

Bu geri çağırma ileti tüketici işlevi olarak tanımlar.

Belirtilen seçim ölçütlerini karşılayan bir ileti bir nesne tanıttıcısında kullanılabilir olduğunda ve bağlantı başlatıldığında, ileti tüketicisi geri çağırma işlevi çağırılır.

MQCBT_EVENT_HANDLER

Bu geri çağırma zamanuyumsuz olay yordamı olarak tanımlar; bir tanıttıcı için iletileri tüketmeye yönlendirilmez.

Olay işleyiciyi tanımlayan MQCB çağrısında *Hobj* gerekli değildir ve belirtildiyse yoksayılır.

Olay işleyici, tüm ileti tüketici ortamını etkileyen koşullar için çağırılır. Bir olay (örneğin, kuyruk yöneticisi ya da bağlantı durdurma ya da susturma) oluştuğunda, tüketici işlevi ileti olmadan çağırılır. Tek bir ileti tüketicisine özgü koşullar için çağırılmaz; örneğin, MQRC_GET_INENGELLENDI.

Olaylar, aşağıdaki ortamlar dışında, bağlantının başlatılıp başlatılmamasından ya da durdurulmasından bağımsız olarak uygulamaya teslim edilir:

- z/OS ortamında CICS
- iş parçacıklı olmayan uygulamalar

Çağırma bu değerlerden birini geçirmese, çağrı *Reason* MQRC_CALLBACK_TYPE_ERROR koduyla başarısız olur.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQCBT_MESSAGE_CONSUMER 'dir.

MQCBD için seçenekler (MQLONG)

Geri çağırma tanımlayıcısı yapısı-Seçenekler alanı

Bu seçeneklerden birini ya da birkaçını belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyse).

MQCBDO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturma durumundaysa MQCB çağrısı başarısız olur.

z/OS sistemlerinde bu seçenek, bağlantı (bir CICS ya da IMS uygulaması için) susturma durumundaysa MQCB çağrısına da başarısız olmasına neden olur.

MQCB çağrısında geçirilen MQGMO seçeneklerinde, ileti tüketicilerine susturulurken bildirim gönderilmesini sağlamak için MQGMO_FAIL_IF QUIESCING belirtin.

Denetim seçenekleri: Aşağıdaki seçenekler, tüketicinin durumu değiştiğinde, ileti olmadan geri çağırma işlevinin çağrılıp çağrılmayacağını denetler:

MQCBDO_REGISTER_CALL

Geri çağırma işlevi MQCBCT_REGISTER_CALL çağrı tipiyle çağrılır.

MQCBDO_START_CALL

Geri çağırma işlevi MQCBCT_START_CALL çağrı tipiyle çağrılır.

MQCBDO_STOP_CALL

Geri çağırma işlevi MQCBCT_STOP_CALL çağrı tipiyle çağrılır.

MQCBDO_DEREGISTER_CALL

Geri çağırma işlevi MQCBCT_DEREGISTER_CALL çağrı tipiyle çağrılır.

MQCBDO_EVENT_CALL

Geri çağırma işlevi MQCBCT_EVENT_CALL çağrı tipiyle çağrıldı.

MQCBDO_MC_EVENT_CALL

Geri çağırma işlevi MQCBCT_MC_EVENT_CALL çağrı tipiyle çağrıldı.

Bu arama tipleriyle ilgili daha fazla ayrıntı için bkz. [CallType](#).

Varsayılan seçenek: Açıklanan seçeneklerden herhangi birine gerek duymuyorsanız, aşağıdaki seçeneği kullanın:

MQCBDO_NONE

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar.

MQCBDO_NONE, program belgelerine yardımcı olmak üzere tanımlandı; bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğundan bu tür bir kullanım algılanamaz.

Bu bir giriş alanıdır. *Options* alanının ilk değeri MQCBDO_NONE.

MQCBD için CallbackArea (MQPTR)

Geri çağırma tanımlayıcı yapısı- CallbackArea alanı

Bu, geri çağırma işlevinin kullanabileceği bir alandır.

Kuyruk yöneticisi bu alanın içeriğine dayalı olarak karar vermez ve geri çağırma işlevi bildirimindeki bir parametre olan MQCBC yapısındaki [CallbackArea](#) alanından değiştirilmeden geçirilir.

Değer yalnızca MQOP_REGISTER değerine sahip bir *Operation* için kullanılır; şu anda tanımlı geri çağırma yok, önceki bir tanımlamanın yerine geçmez.

Bu, geri çağırma işlevine ilişkin bir giriş ve çıkış alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

MQCBD için CallbackFunction (MQPTR)

Geri çağırma tanımlayıcı yapısı- CallbackFunction alanı

Geri çağırma işlevi bir işlev çağrısı olarak çağrılır.

Geri çağırma işlevine ilişkin bir gösterge belirtmek için bu alanı kullanın.

CallbackFunction ya da *CallbackName* belirtmeniz gerekir. Her ikisini de belirtirseniz, MQRC_CALLBACK_ROUTINE_ERROR neden kodu döndürülür.

CallbackName ya da *CallbackFunction* ayarlanmazsa, çağrı MQRC_CALLBACK_ROUTINE_ERROR neden koduyla başarısız olur.

Bu seçenek şu ortamda desteklenmez: İşlev göstergesi başvurularını desteklemeyen programlama dilleri ve derleyiciler. Böyle durumlarda, çağrı MQRC_CALLBACK_ROUTINE_ERROR neden koduyla başarısız olur.

z/OS z/OS işletim sisteminde, işlevin işletim sistemi bağlantı kurallarıyla çağrılması beklenir. Örneğin, C programlama dilinde şunları belirtin:

```
#pragma linkage(MQCB_FUNCTION,OS)
```

Bu bir giriş alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

Not: CICS komutunu IBM WebSphere MQ 7.0.1 ile kullanırken, aşağıdakiler destekleniyorsa, zamanuyumsuz tüketim desteklenir:

- Apar PK66866 CICS TS 3.2 ' ye uygulanır
- Apar PK89844 , CICS TS 4.1 ürününe uygulanır

MQCBD için CallbackName (MQCHAR128)

Geri çağırma tanımlayıcı yapısı- CallbackName alanı

Geri çağırma işlevi dinamik olarak bağlı bir program olarak çağrılır.

CallbackFunction ya da *CallbackName* belirtmeniz gerekir. Her ikisini de belirtirseniz, MQRC_CALLBACK_ROUTINE_ERROR neden kodu döndürülür.

CallbackName ya da *CallbackFunction* ayarlanmazsa, çağrı MQRC_CALLBACK_ROUTINE_ERROR neden koduyla başarısız olur.

Kullanılacak ilk geri çağırma yordamı kaydedildiğinde birim yüklenir ve bu yordamı kullanan son geri çağırma yordamı kayıttan kaldırıldığında birim kaldırılır.

Aşağıdaki metinde belirtilenler dışında, ad alan içinde sola yaslanır ve boşluk bırakılmaz; adın kendisi alan uzunluğuna kadar boşluklarla doldurulur. Aşağıdaki açıklamalarda, köşeli ayraç ([]) isteğe bağlı bilgileri gösterir:

IBM i

Geri çağırma adı aşağıdaki biçimlerden biri olabilir:

- Kitaplık "/" Programı
- Kitaplık "/" ServiceProgram ("FunctionName")

Örneğin, MyLibrary/MyProgram(MyFunction).

Kitaplık adı *LIBL olabilir. Hem kitaplık hem de program adları en çok 10 karakterle sınırlıdır.

AIX and Linux

Geri çağırma adı, dinamik olarak yüklenebilir bir modülün ya da kitaplığın adıdır; soneki, o kitaplıkta bulunan bir işlevin adıdır. İşlev adı ayraç içine alınmalıdır. Kitaplık adının başına isteğe bağlı olarak bir dizin yolu konabilir:

```
[path]library(function)
```

Yol belirtilmezse, sistem arama yolu kullanılır.

Ad en çok 128 karakterle sınırlıdır.

Windows

Geri çağırma adı, bu kitaplıkta bulunan bir işlevin adının eklendiği dinamik bağlantı kitaplığının adıdır. İşlev adının parantez içine alınması gerekir. Kitaplık adının öneki, isteğe bağlı olarak bir dizin yolu ve sürücü olabilir:

```
[d:][path]library(function)
```

Sürücü ve yol belirtilmezse, sistem arama yolu kullanılır.

Ad en çok 128 karakterle sınırlıdır.

z/OS

Geri çağırma adı, LINK ya da LOAD makrosunun EP parametresinde belirtim için geçerli olan bir yükleme modülünün adıdır.

Ad en çok 8 karakterle sınırlıdır.

z/OS CICS

Geri çağırma adı, EXEC CICS LINK komut makrosunun PROGRAM parametresinde belirtim için geçerli olan bir yükleme modülünün adıdır.

Ad en çok 8 karakterle sınırlıdır.

Program, kurulu PROGRAM tanımının REMOTESYTEM seçeneği ya da dinamik yönlendirme programı kullanılarak uzak olarak tanımlanabilir.

Program IBM MQ API çağrılarını kullanacaksa, uzak CICS bölgesi IBM MQ ' e bağlanmalıdır. Ancak, MQCBC yapısındaki Hobj alanının uzak bir sistemde geçerli olmadığını unutmayın.

CallbackName yüklenirken bir hata oluşursa, uygulamaya aşağıdaki hata kodlarından biri döndürülür:

- MQRC_MODULE_NOT_FOUND
- MQRC_MODULE_INVALID
- MQRC_MODULE_ENTRY_NOT_FOUND

Ayrıca, yüklemenin denendiği birimin adını ve işletim sisteminden gelen neden kodunu içeren hata günlüğüne bir ileti yazılır.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş bir dizilim ya da boşluktur.

MaxMsgMQCBD için uzunluk (MQLONG)

Bu, tanıtıcıdan okunabilen ve geri çağırma yordamına verilen en uzun iletinin bayt cinsinden uzunluğudur. Geri Çağırma tanımlayıcı yapısı- MaxMsgUzunluk alanı

Bir iletinin uzunluğu daha uzunsa, geri çağırma yordamı iletinin *MaxMsgLength* byte 'ını alır ve neden kodu:

- MQRC_TRUNCATED_MSG_FAILED ya da
- MQGMO_ACCEPT_TRUNCATED_MSG belirttiyseniz MQRC_TRUNCATED_MSG_ACCEPTED.

Gerçek ileti uzunluğu, MQCBC yapısının DataLength alanında sağlanır.

Aşağıdaki özel değer tanımlandı:

MQCBD_FULL_MSG_LENGTH

Arabellek uzunluğu, sistem tarafından kesilmeden ileti döndürecek şekilde ayarlanır.

İletiyi almak üzere bir arabellek ayırmak için yeterli bellek yoksa, sistem geri çağırma işlevini MQRC_STORAGE_NOT_ALLOWED neden koduyla çağırır.

Örneğin, veri dönüştürme isteğinde bulunursanız ve ileti verilerini dönüştürmek için yeterli bellek yoksa, dönüştürülemeyen ileti geri çağırma işlevine geçirilir.

Bu bir giriş alanıdır. *MaxMsgLength* alanının ilk değeri MQCBD_FULL_MSG_LENGTH ' dir.

MQCHARV-Değişken Uzunluklu Dizgi

Değişken uzunluklu bir dizgiyi tanımlamak için MQCHARV yapısını kullanın.

Kullanılabilirlik

MQCHARV yapısı aşağıdaki altyapılarda bulunur:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere baęlı IBM MQ MQI clients için.

Karakter kümesi ve kodlama

MQCHARV içindeki veriler, MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında ve yapı içindeki VSCCSID alanının karakter kümesinde olmalıdır. Uygulama bir MQ istemcisi olarak çalışıyorsa, yapı istemcinin kodlamasında olmalıdır. Bazı karakter kümelerinin kodlamaya baęlı bir gösterimi vardır. VSCCSID bu karakter kümelerinden biriye, kullanılan kodlama, MQCHARV ' deki dięer alanlarla aynı kodlamadır. VSCCSID ile tanıtılan karakter takımı çift baytlık karakter takımı (DBCS) olabilir.

Kullanım

MQCHARV yapısı, bunu içeren yapıyla bitişik olmayabilecek verileri adresler. Bu verileri ele almak için, işaretçi veri tipiyle bildirilen alanlar kullanılabilir. COBOL ' un tüm ortamlarda işaretçi veri tipini desteklemediğini unutmayın. Bu nedenle, veriler MQCHARV ' yi içeren yapının başlangıcındaki verilerin görelî konumunu içeren alanlar kullanılarak da ele alınabilir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak deęil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 475. MQCHARV içindeki alanlar		
Alan adı ve açıklaması	Deęişmezin adı	Deęişmezin ilk deęeri (varsa)
VSPtr (deęişken uzunluklu dizgiyi gösteren gösterge)	Yok	Boş deęerli gösterge ya da boş deęerli byte 'lar.
VSOOffset (bu MQCHARV yapısını içeren yapının başlangıcındaki deęişken uzunluklu dizginin bayt cinsinden görelî konumu)	Yok	0
VSBuSize (VSPtr ya da VSOOffset alanı tarafından adreslenen arabelleğin bayt cinsinden boyutu)	MQVS_USE_VSLENGTH	0
VSLkalınlık (VSPtr ya da VSOOffset alanı tarafından adreslenen deęişken uzunluklu dizginin bayt cinsinden uzunluęu)	Yok	0
VSCCSID (VSPtr ya da VSOOffset alanı tarafından adreslenen deęişken uzunluklu dizginin karakter takımı tanıtıcısı)	MQCCSI_APPL	-3

Not: C programlama dilinde, MQCHARV_DEFAULT makro deęişkeni çizelgede listelenen deęerleri içerir. Yapıdaki alanlar için başlangıç deęerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQCHARV MyVarStr = {MQCHARV_DEFAULT};
```

Dil bildirimleri

MQCHARV için C bildirimi

```
typedef struct tagMQCHARV MQCHARV;
struct tagMQCHARV {
    MQPTR      VSPtr;                /* Address of variable length string */
    MQLONG     VSOffset;             /* Offset of variable length string */
    MQLONG     VSBufSize;           /* Size of buffer */
    MQLONG     VSLength;            /* Length of variable length string */
    MQLONG     VSCCSID;             /* CCSID of variable length string */
};
```

MQCHARV için COBOL bildirimi

```
** 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.
```

Not: Bir COBOL uygulamasını ortamlar arasında aktarmak istiyorsanız, işaretçi veri tipinin tüm amaçlanan ortamlarda kullanılabilir olup olmadığını öğrenmeniz gerekir. Değilse, uygulamanın işaretçi alanları yerine kayma alanlarını kullanarak verileri ele alması gerekir. Göstergelerin desteklenmediği ortamlarda, işaretçi alanlarını uygun uzunlukta bayt dizgileri olarak bildirebilirsiniz; ilk değer tümüyle boş değerli byte dizgisi olur. Göreli konum alanlarını kullanıyorsanız, bu başlangıç değerini değiştirmeyin. Sağlanan kopya kitaplarını değiştirmeden bunu yapmanın bir yolu, aşağıdakileri kullanmaktır:

```
COPY CMQCHRVV REPLACING POINTER BY ==BINARY PIC S9(9)==.
```

Burada CMQCHRVV , kullanılacak kopya defteriyle değiştirilebilir.

MQCHARV için PL/I bildirimi

```
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 */
```

MQCHARV için High Level Assembler bildirimi

```
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
                  ORG  MQCHARV
MQCHARV_AREA     DS  CL(MQCHARV_LENGTH)
```

MQCHARV için VSPtr (MQPTR)

Bu, değişken uzunluklu dizgiyi gösteren bir işaretçidir.

Değişken uzunluklu dizgiyi belirtmek için VSPtr ya da VSOffset alanını kullanabilirsiniz, ancak ikisini birden kullanamazsınız.

Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

MQCHARV için VSOffset (MQLONG)

Görelî konum pozitif ya da negatif olabilir. Değişken uzunluklu dizgiyi belirtmek için VSPtr ya da VSOffset alanını kullanabilirsiniz, ancak ikisini birden kullanamazsınız. MQCHARV ' nin başlangıcından itibaren değişken uzunluklu dizginin bayt cinsinden görelî konumu ya da bunu içeren yapı.

MQCHARV yapısı başka bir yapıya yerleştirildiğinde, bu değer, bu MQCHARV yapısını içeren yapının başlangıcındaki değişken uzunluklu dizginin bayt cinsinden görelî konumudur. MQCHARV yapısı başka bir yapıya yerleştirilmediğinde, örneğin, bir işlev çağrısında parametre olarak belirtilirse, görelî konum MQCHARV yapısının başlangıçla görelî olur.

Bu alanın ilk değeri 0 'dır.

MQCHARV için VSBufSize (MQLONG)

Bu, VSPtr ya da VSOffset alanı tarafından adreslenen arabelleğin bayt cinsinden boyutudur.

MQCHARV yapısı bir işlev çağrısında çıkış alanı olarak kullanıldığında, bu alan sağlanan arabellek uzunluğuyla kullanıma hazırlanmalıdır. VSLkalınlık değeri VSBufSize değerinden büyükse, arabellekteki çağırana yalnızca VSBufSize byte veri döndürülür.

Bu değer, sıfırdan büyük ya da sıfıra eşit bir değer ya da tanınan şu özel değer olmalıdır:

MQVS_USE_VSLENGTH

Belirtildiğinde, arabelleğin uzunluğu MQCHARV yapısındaki VSLkalınlık alanından alınır. Yapıyı bir çıkış alanı olarak kullanırken bu değeri kullanmayın ve bir arabellek sağlanır.

Bu, bu alanın ilk değeridir.

MQCHARV için VSLkalınlık (MQLONG)

VSPtr ya da VSOffset alanı tarafından adreslenen değişken uzunluklu dizginin bayt cinsinden uzunluğu.

Bu alanın ilk değeri 0 'dır. Değer sıfırdan büyük ya da sıfıra eşit ya da tanınan şu özel değer olmalıdır:

MQVS_NULL_SONLANDIRILDI

MQVS_NULL_TERMINATED belirtilmezse, VSLkalınlık byte 'ları dizginin bir parçası olarak eklenir. Boş karakterler varsa, dizgiyi sınırlamazlar.

MQVS_NULL_TERMINATED belirtilirse, dizgi, dizgide karşılaşılan ilk boş değerle sınırlanmıştır. Boş değer, o dizginin bir parçası olarak dahil edilmez.

Not: MQVS_NULL_TERMINATED belirtilirse, bir dizgiyi sonlandırmak için kullanılan boş karakter, VSCCSID ile belirtilen kod kümesinde boş değerlidir.

Örneğin, UTF-16 (CCSID 1200, 13488 ve 17584) içinde, bu iki baytlık Unicode kodlamasıdır; burada boş değer, tüm sıfırların 16 bitlik sayısıyla gösterilir. UTF-16 'da, karakterlerin (örneğin, 7 bit ASCII karakterleri) bir parçası olan tüm sıfıra ayarlı tek baytların bulunması yaygındır, ancak dizgiler yalnızca çift byte sınırında iki 'sıfır ' bayt bulunduğu boş sonlandırılır. Her biri geçerli karakterlerin bir parçası olduğunda tek bir sınırda iki 'sıfır' bayt almak mümkündür. Örneğin x '01' x '00' 00 'x' 30 ' geçerli iki Unicode karakteri temsil eder ve dizgiyi boş değerli olarak sonlandırmaz.

MQCHARV için VSCCSID (MQLONG)

Bu, **VSPtr** ya da **VSOffset** alanı tarafından adreslenen değişken uzunluklu dizginin karakter kümesi tanıtıcısıdır.

Bu alanın ilk değeri *MQCCSI_APPL* ' dir; bu değer, MQ tarafından yürürlükteki işlemin gerçek karakter kümesi tanıtıcısı olarak değiştirilmesi gerektiğini belirtmek için tanımlanır. Sonuç olarak, *MQCCSI_APPL* değişiminin değeri hiçbir zaman değişken uzunluklu bir dizgiyle ilişkilendirilmez.

Bu alanın ilk değeri, derleme biriminize ilişkin *MQCCSI_APPL* değişimi için farklı bir değer tanımlanarak değiştirilebilir. Bunu nasıl yapacağınız, uygulamanızın programlama diline bağlıdır.

z/OS z/OS sistemlerinde, *MQCCSI_APPL* tarafından kullanılan CCSID varsayılan uygulaması aşağıdaki gibi tanımlanır:

- DLL arabirimini kullanan toplu LE uygulamaları için varsayılan değer, **MQCONN** ' un yayınlandığı sırada yürürlükteki ülke değeriyle ilişkilendirilen CODESET değeridir (varsayılan değer 1047 'dir).
- Toplu iş MQ sınırlı kod öbeklerinden biriyle bağ tanımlanan toplu LE uygulamaları için varsayılan değer, **MQCONN** sonrasında yayınlanan ilk MQI çağrısı sırasında yürürlükteki ülke değeriyle ilişkilendirilmiş CODESET değeridir (varsayılan değer 1047 'dir).
- z/OS UNIX System Services iş parçacığında çalışan LE dışı toplu uygulamalar için varsayılan değer, **MQCONN** sonrasında yayınlanan ilk MQI çağrısı sırasında THLICCSID değeridir (varsayılan değer 1047 'dir).
- Diğer toplu iş uygulamaları için varsayılan değer, kuyruk yöneticisinin CCSID değeridir.

MQCCSI_APPL ' in yeniden tanımlanması

Aşağıdaki örneklerde, çeşitli programlama dillerinde MQCCSI_APPL değerini nasıl geçersiz kılabileceğiniz gösterilmiştir. MQCCSI_APPL değerini değiştirerek, her değişken uzunluklu dizgi için VSCCSID ' yi ayrı olarak ayarlama gereksinimini kaldırabilirsiniz. Bu örneklerde CCSID 1208 olarak ayarlanır; bu değeri, gerek duyduğunuz değerle değiştirin. Bu değer, belirli bir MQCHARV eşgörünümünde VSCCSID ' yi ayarlayarak geçersiz kılabileceğiniz varsayılan değer olur.

C kullanımı

```
#define MQCCSI_APPL 1208
#include <cmqc.h>
```

COBOL kullanımı

```
COPY CMQXYZV REPLACING -3 BY 1208.
```

PL/I kullanımı

```
%MQCCSI_APPL = '1208';
%include syslib(cmqp);
```

High Level Assembler kullanımı

```
MQCCSI_APPL EQU 1208
CMQA LIST=NO
```

MQCIH- CICS bridge üstbilgisi

MQCIH yapısı, CICS bridge üzerinden CICS ' e gönderilen bir iletiye ilişkin üstbilgi bilgilerini açıklar.

Desteklenen herhangi bir IBM MQ altyapısı için MQCIH yapısını içeren bir ileti yaratabilir ve iletebilirsiniz, ancak CICS bridge yalnızca bir IBM MQ for z/OS kuyruk yöneticisi tarafından kullanılabilir. Bu nedenle, iletinin z/OS dışı bir kuyruk yöneticisinden CICS ' e ulaşabilmesi için kuyruk yöneticisi ağınızın, iletinin yöneltilebileceği en az bir z/OS kuyruk yöneticisi içermesi gerekir.

IBM MQ 9.0 tarafından desteklenen tüm CICS sürümleri ve daha sonra köprüünün CICS tarafından sağlanan sürümü kullanılır. IBM MQ CICS bağdaştırıcısının ve IBM MQ CICS bridge bileşenlerinin yapılandırılmasıyla ilgili daha fazla bilgi için CICS belgelerinin [MQ bağlantılarının yapılandırılması](#) bölümüne bakın.

Kullanılabilirlik

MQCIH yapısı aşağıdaki altyapılarda bulunur:

-  AIX
-  Linux
-  Windows
-  z/OS

ve bu sistemlere baęlı IBM MQ MQI clients için.

Biçim Adı

MQFMT_CICS

Sürüm

Yürürlükteki MQCIH sürümü: MQCIH_VERSION_2. Yapının yalnızca daha yeni sürümünde var olan alanlar, izleyen tanımlarda bu şekilde tanımlanır.

Desteklenen programlama dilleri için sağlanan üstbilgi, COPY ve INCLUDE dosyaları, *Version* alanının ilk değeri MQCIH_VERSION_2olarak ayarlanmış MQCIH ' nin en son sürümünü içerir.

Karakter kümesi ve kodlama

MQCIH yapısı ve uygulama iletisi verileri için kullanılan karakter kümesi ve kodlama için özel koşullar geçerlidir:

- CICS bridge kuyruğunun sahibi olan kuyruk yöneticisine bağlanan uygulamaların, kuyruk yöneticisinin karakter kümesi ve kodlamasında bulunan bir MQCIH yapısı sağlaması gerekir. Bunun nedeni, bu durumda MQCIH yapısının veri dönüştürmesinin gerçekleştirilmemiş olmasıdır.
- Diğer kuyruk yöneticilerine bağlanan uygulamalar, desteklenen karakter kümelerinde ve kodlamasında bulunan bir MQCIH yapısı sağlayabilir; CICS bridge kuyruğunun sahibi olan kuyruk yöneticisine baęlı alıcı ileti kanalı aracısı, MQCIH yapısını dönüştürür.
- MQCIH yapısını izleyen uygulama iletisi verileri, MQCIH yapısıyla aynı karakter kümesinde ve kodlamada olmalıdır. MQCIH yapısındaki *CodedCharSetId* ve *Encoding* alanlarını, uygulama ileti verilerinin karakter kümesini ve kodlamasını belirtmek için kullanamazsınız.

Veriler, kuyruk yöneticisi tarafından desteklenen yerleşik biçimlerden biri değilse, uygulama ileti verilerini dönüştürmek için bir veri dönüştürme çıkışı sağlamanız gerekir.

Kullanım

Uygulama Çizelge 477 sayfa 300içinde gösterilen ilk değerlerle aynı değerleri gerektiriyorsa ve köprü AUTH=LOCAL ya da AUTH=TANIMLA ile çalışıyorsa, iletiden MQCIH yapısını atlayabilirsiniz. Diğer tüm durumlarda, yapı mevcut olmalıdır.

Köprü version-1 ya da version-2 MQCIH yapısını kabul eder, ancak 3270 hareketleri için version-2 yapısı kullanmanız gerekir.

Uygulama, istek alanları olarak belgelenen alanların, köprüye gönderilen iletide uygun değerlere sahip olmasını sağlamalıdır; bu alanlar köprüye girişir.

Yanıt alanları olarak belgelenen alanlar, köprünün uygulamaya gönderdiği yanıt iletisinde CICS bridge tarafından ayarlanır. *ReturnCode*, *Function*, *CompCode*, *Reason* ve *AbendCode* alanlarında hata bilgileri döndürülür, ancak bunların tümü her durumda ayarlanmaz. Aşağıdaki tabloda, farklı *ReturnCode*değerleri için hangi alanların ayarlandığı gösterilmektedir.

Çizelge 476. MQCIH için MQCIH yapısındaki hata bilgisi alanlarının içerięi				
ReturnCode	Function	CompCode	Reason	AbendCode
MQCRC_OK	-	-	-	-

Çizelge 476. MQCIH için MQCIH yapısındaki hata bilgisi alanlarının içeriği (devamı var)				
ReturnCode	Function	CompCode	Reason	AbendCode
MQCRC_BRIDGE_HATA	-	-	MQFB_CICS_*	-
MQCRC_MQ_API_ERROR MQCRC_BRIDGE_TIMEOUT	MQ çağrı adı	MQ CompCode	MQ Reason	-
MQCRC_CICS_EXEC_ERROR MQCRC_SECURITY_ERROR MQCRC_PROGRAM_NOT_KULLANILABILIR MQCRC_TRANSID_NOT_VAR	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	-
MQCRC_BRIDGE_ABEND MQCRC_APPLICATION_ABEND	-	-	-	CICS KODU

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 477. MQCIH için MQCIH içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQCIH_STRUC_ID	'CIH'
<u>Sürüm</u> (yapı sürümü numarası)	MQCIH_VERSION_2	2
<u>StrucLength</u> (MQCIH yapısının uzunluğu)	MQCIH_LENGTH_2	180
<u>Kodlama</u> (ayrılmış)	Yok	0
<u>CodedCharSetId</u> (ayrılmış)	Yok	0
<u>Biçim</u> (MQCIH ' yi izleyen verilerinMQ biçimi adı)	MQFMT_NONE	Boşluklar
<u>İşaretler</u> (işaretler)	MQCIH_YOK	0
<u>ReturnCode</u> (köprüden dönüş kodu)	MQCRC_OK	0
<u>CompCode</u> (MQ tamamlama kodu ya da CICS EIBRESP)	MQCC_OK	0
<u>Neden</u> (MQ neden ya da geribildirim kodu ya da CICS EIBRESP2)	MQRC_NONE	0
<u>UOWControl</u> (iş birimi denetimi)	YALNIZCA MQCUOWC_ONLY	273
<u>GetWaitInterval</u> (Köprü görevi tarafından verilen MQGET çağrısı için bekleme aralığı)	MQCGWI_DEFAULT	-2
<u>LinkType</u> (link tipi)	MQCLT_PROGRAM	1
<u>OutputDataLength</u> (çıkış COMMAREA veri uzunluğu)	MQCODL_AS_INPUT	-1
<u>FacilityKeepTime</u> (köprü tesisini serbest bırakma süresi)	Yok	0
<u>ADSDescriptor</u> (ADS tanımlayıcısını gönder/al)	MQCADSD_NONE	0
<u>ConversationalTask</u> (görevin etkileşimli olup olmayabileceği)	MQCCT_NO	0

Çizelge 477. MÇCIH için MÇCIH içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>TaskEndStatus</u> (görevin sonundaki durum)	MÇCTES_NOSYNC	0
<u>Tesis</u> (köprü olanağı simgesi)	MÇCFAC_NONE	Boş Değerler
<u>İşlev</u> (MÇ çağrı adı ya da CICS EIBFN işlevi)	MÇCFUNC_NONE	Boşluklar
<u>AbendCode</u> (olağandışı bitiş kodu)	Yok	Boşluklar
<u>Authenticator</u> (parola ya da geçiş bileti)	Yok	Boşluklar
<u>Reserved1</u> (ayrılmış)	Yok	Boşluklar
<u>ReplyToBiçim</u> (MÇ yanıt iletisinin biçim adı)	MÇFMT_NONE	Boşluklar
<u>RemoteSysId</u> (kullanılacak uzak CICS sistem tanıtıcısı)	Yok	Boşluklar
<u>RemoteTransTnt</u> (kullanılacak CICS RTRANSID)	Yok	Boşluklar
<u>TransactionId</u> (eklenecek işlem)	Yok	Boşluklar
<u>FacilityLike</u> (uçbirim öykünülen öznelikler)	Yok	Boşluklar
<u>AttentionId</u> (AID tuşu)	Yok	Boşluklar
<u>StartCode</u> (hareket başlatma kodu)	MÇCSC_NONE	Boşluklar
<u>CancelCode</u> (hareket kodunu olağandışı sona erdir)	Yok	Boşluklar
<u>NextTransactionId</u> (eklenecek sonraki işlem)	Yok	Boşluklar
<u>Reserved2</u> (ayrılmış)	Yok	Boşluklar
<u>Reserved3</u> (ayrılmış)	Yok	Boşluklar
Not: <i>Version</i> , MÇCIH_VERSION_2' den küçükse kalan alanlar yoktur.		
<u>CursorPosition</u> (imleç konumu)	Yok	0
<u>ErrorOffset</u> (iletideki hatanın görel konumu)	Yok	0
<u>InputItem</u> (giriş öğesi)	Yok	0
<u>Reserved4</u> (ayrılmış)	Yok	0
<p>Notlar:</p> <ol style="list-style-type: none"> – simgesi, tek bir boş karakteri temsil eder. C programlama dilinde, makro değişkeni MÇCIH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın: <pre>MÇCIH MyCIH = {MÇCIH_DEFAULT};</pre>		

Dil bildirimleri

MÇCIH için C bildirim

```
typedef struct tagMÇCIH MÇCIH;
struct tagMÇCIH {
    MÇCHAR4 StructId;          /* Structure identifier */
    MÇLONG Version;          /* Structure version number */
};
```

```

MQLONG  StruLength;          /* 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 */
};

```

MQCIH için COBOL bildirimi

```

** 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-STRUCLNGTH 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-REMOETTRANSID 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.

```

MQCIH için PL/I bildirimi

```

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
conversational */
3 TaskEndStatus fixed bin(31), /* Status at end of task */
3 Facility char(8), /* Bridge facility token */
3 Function char(4), /* 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 */

```

MQCIH için High Level Assembler bildirimi

```

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_CODEDCCHARSETID 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_FACILITYRELEASETIME DS F Bridge facility release time
MQCIH_ADSDSCRIPTOR 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
               ORG MQCIH
MQCIH_AREA     DS CL(MQCIH_LENGTH)

```

MQCIH için Visual Basic bildirimi

```

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		

MQCIH için StrucId (MQCHAR4)

Bu, CICS bilgi üstbilgisi yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQCIH_STRUC_ID.

Değer şöyle olmalıdır:

MQCIH_STRUC_ID

CICS bilgi üstbilgisi yapısına ilişkin tanıttıcı.

C programlama dili için, MQCIH_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQCIH_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQCIH için Sürüm (MQLONG)

Bu alan bir istek alanıdır. İlk değeri: MQCIH_VERSION_2.

Değer aşağıdakilerden biri olmalıdır:

MQCIH_VERSION_1

Version-1 CICS bilgi üstbilgisi yapısı.

MQCIH_VERSION_2

Version-2 CICS bilgi üstbilgisi yapısı.

Yalnızca yapının daha yeni sürümünde var olan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCIH_CURRENT_VERSION

CICS bilgi üstbilgisi yapısının geçerli sürümü.

MQCIH için StrucLength (MQLONG)

Bu alan, başlangıç değeri MQCIH_LENGTH_2 olan bir istek alanıdır.

Değer aşağıdakilerden biri olmalıdır:

MQCIH_LENGTH_1

version-1 CICS bilgi üstbilgisi yapısının uzunluğu.

MQCIH_LENGTH_2

version-2 CICS bilgi üstbilgisi yapısının uzunluğu.

Aşağıdaki değişmez, yürürlükteki sürümün uzunluğunu belirtir:

MQCIH_CURRENT_LENGTH

CICS bilgi üstbilgisi yapısının geçerli sürümünün uzunluğu.

MQCIH için kodlama (MQLONG)

Bu alan ayrılmış bir alandır; değeri önemli değildir. İlk değeri 0 'dır.

MQCIH yapısını izleyen desteklenen yapılar için Kodlama, MQCIH yapısının kendisiyle ve önceki IBM MQ üstbilgisinden alınan Kodlamayla aynıdır.

MQCIH için CodedCharSetId (MQLONG)

CodedCharSetId ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri 0 'dır.

MQCIH yapısını izleyen desteklenen yapılarla ilişkin Karakter Kümesi Tanıtıcısı, MQCIH yapısının kendisinin Karakter Kümesi Tanıtıcısı ile aynıdır ve önceki IBM MQ üstbilgisinden alınır.

MQCIH için biçim (MQCHAR8)

Bu alan, MQCIH yapısını izleyen verilerin IBM MQ biçim adını gösterir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *Format* alanını kodlamaya ilişkin kurullarla aynıdır.

ReplyToFormat alanında MQFMT_NONE değeri varsa, yanıt iletisi için de bu biçim adı kullanılır.

- DPL istekleri için, *Format* COMMAREA ' nın biçim adı olmalıdır.
- 3270 istekleri için *Format* CSQCBDCI olmalı ve köprü, Yanıt iletileri için biçimi CSQCBDCO olarak ayarlamalıdır.

Bu biçimlere ilişkin veri dönüştürme çıkışları, çalışacakları kuyruk yöneticisine kurulmalıdır.

İstek iletisi bir hata yanıtı iletisi oluşturursa, hata yanıt iletisinin biçim adı MQFMT_STRING olur.

Bu alan bir istek alanıdır. Bu alanın uzunluğu MQ_FORMAT_LENGTH tarafından verilir. Bu alanın ilk değeri MQFMT_NONE.

MQCIH için İşaretler (MQLONG)

Bu alan bir istek alanıdır. Bu alanın ilk değeri MQCIH_NONE.

Değer şöyle olmalıdır:

MQCIH_YOK

Bayrak yok.

MQCIH_PASS_SÜRE SONU

Yanıt iletisi şunları içerir:

- İstek iletisiyle aynı süre bitimi raporu seçenekleri.
- Köprünün işleme saati için ayarlama yapılmadan istek iletisinden kalan süre sonu.

Bu değeri atlarsanız, süre sonu *sınırsız* olarak ayarlanır.

MQCIH_REPLY_WITHOUT_NULLS

CICS DPL program isteğinin yanıt iletisi uzunluğu, DPL programı tarafından döndürülen COMMAREA 'nın sonunda sondaki boş değerleri (X'00') dışlayacak şekilde ayarlanır. Bu değer ayarlanmazsa, boş değerler önemli olabilir ve tam COMMAREA döndürülür.

MQCIH_SYNC_ON_RETURN

DPL isteklerine ilişkin CICS bağlantısı SYNCONRETURN seçeneğini kullanır; bu, program başka bir CICS bölgesine gönderildiğinde program tamamlandığında CICS ' un bir eşitleme noktası almasına neden olur. Köprü, isteği hangi CICS bölgesine sevk edileceğini belirtmez; bu, CICS program tanımı ya da iş yükü dengeleme olanakları tarafından denetlenir.

MQCİH için ReturnCode (MQLONG)

Bu alanın değeri, köprü tarafından gerçekleştirilen işlemlerin sonucunu açıklayan CICS bridge dönüş kodudur. Bu alan, ilk değeri MQCRC_OK olan bir yanıt alanıdır.

Function, CompCode, Reasonve AbendCode alanları ek bilgi içerebilir (bkz. [Çizelge 476 sayfa 299](#)). Değer aşağıdakilerden biridir:

MQCRC_APPLICATION_ABEND

(5, X'005 ') Uygulama olağandışı bitti.

MQCRC_BRIDGE_ABEND

(4, X'004 ') CICS bridge olağandışı bitti.

MQCRC_BRIDGE_HATA

(3, X'003 ') CICS bridge bir hata algıladı.

MQCRC_BRIDGE_TIMEOUT

(8, X'008 ') Yürürlükteki iş birimi içindeki ikinci ya da sonraki ileti, belirtilen süre içinde alınmadı.

MQCRC_CICS_EXEC_ERROR (MQCRC_CICS_HATASI)

(1, X'001 ') EXEC CICS deyimi bir hata saptadı.

MQCRC_MQ_API_ERROR

(2, X'002 ') MQ çağrısı bir hata saptadı.

MQCRC_OK

(0, X'000 ') Hata yok.

MQCRC_PROGRAM_YOK

(7, X'007 ') Program kullanılmıyor.

MQCRC_SECURITY_HATA

(6, X'006 ') Güvenlik hatası oluştu.

MQCRC_TRANSID_NOT_KULLANILAMIYOR

(9, X'009 ') İşlem kullanılmıyor.

MQCİH için CompCode (MQLONG)

Bu alan bir yanıt alanıdır. İlk değeri MQCC_OK

Bu alanda döndürülen değer *ReturnCode* ; bkz. [Çizelge 476 sayfa 299](#).

MQCİH için neden (MQLONG)

Bu alan bir yanıt alanıdır. İlk değeri MQRC_NONE.

Bu alanda döndürülen değer *ReturnCode* ; bkz. [Çizelge 476 sayfa 299](#).

MQCİH için UOWControl (MQLONG)

Bu alan, CICS bridgetarafından gerçekleştirilen iş birimi işlemlerini denetleyen bir istek alanıdır. Bu alanın ilk değeri MQCUOWC_ONLY 'dir.

Köprüden tek bir hareket ya da bir iş birimi içinde bir ya da daha çok program çalıştırmasını isteyebilirsiniz. Bu alan, CICS bridge ' in bir iş birimi başlatmasını, yürürlükteki iş birimi içinde istenen işlevi gerçekleştirmesini ya da iş birimini kesinleştirerek ya da geri çekerek sona erdirmesini gösterir. Veri iletim akışlarının eniyilenmesi için çeşitli birleşimler desteklenir.

Değer aşağıdakilerden biri olmalıdır:

YALNIZCA MQCUOWC_ONLY

İş birimini başlatın, işlevi gerçekleştirin ve iş birimini kesinleştirin.

MQCUOWC_CONTINUE

Yürürlükteki iş birimine ilişkin ek veriler (yalnızca 3270).

MQCUOWC_FIRST

İş birimini başlatın ve işlevi gerçekleştirin.

MQCUOWC_MIDDLE

İşlevi yürürlükteki iş birimi içinde gerçekleştir

MQCUOWC_SON

İşlevi gerçekleştirin ve iş birimini kesinleştirin.

MQCUOWC_COMMIT

İş birimini kesinleştirin (yalnızca DPL).

MQCUOWC_BACKOUT

İş birimini geri çevirin (yalnızca DPL).

MQCIH için GetWaitAralığı (MQLONG)

Bu alan bir istek alanıdır. İlk değeri MQCGWI_DEFAULT.

Bu alan yalnızca *UOWControl*, MQCUOWC_FIRST değerine sahip olduğunda geçerlidir. Gönderen uygulamanın, köprü tarafından yayınlanan MQGET çağrılarının bu ileti tarafından başlatılan iş birimine ilişkin ikinci ve sonraki istek iletilerini bekleyeceği yaklaşık süreyi milisaniye cinsinden belirtmesini sağlar. Bu olanak, köprü tarafından kullanılan varsayılan bekleme aralığını geçersiz kılar. Aşağıdaki özel değerleri kullanabilirsiniz:

MQCGWI_DEFAULT

Varsayılan bekleme aralığı.

Bu değer, CICS bridge ' in köprü'nün başlatıldığı zaman belirtilen süreyi beklemesine neden olur.

MQWI_SINIRSIZ

Sınırsız bekleme aralığı.

MQCIH için LinkType (MQLONG)

Bu alan bir istek alanıdır. İlk değeri MQCLT_PROGRAM.

Bu değer, köprü'nün bağlantı oluşturmaya çalıştığı nesnenin tipini gösterir. Aşağıdaki değerlerden biri olmalıdır:

MQCLT_PROGRAM

DPL programı.

MQCLT_TRANSACTION

3270 hareketi.

MQCIH için OutputDataUzunluğu (MQLONG)

Bu alan, yalnızca DPL programları için kullanılan bir istek alanıdır. İlk değeri MQCODL_AS_INPUT.

Bu değer, yanıt iletilinde istemciye döndürülecek kullanıcı verilerinin uzunluğudur. Bu uzunluk, 8 baytlık program adını içerir. Bağlı programa geçirilen COMMAREA ' nın uzunluğu, bu alanın üst sınırı ve istek iletilisindeki kullanıcı verilerinin uzunluğu eksi 8 'dir.

Not: Bir iletideki kullanıcı verilerinin uzunluğu, MQCIH yapısı dışında iletinin uzunluğudur.

İstek iletilisindeki kullanıcı verilerinin uzunluğu *OutputDataLength* değerinden küçükse, LINK komutunun DATALENGTH seçeneği kullanılır ve LINK işlevinin başka bir CICS bölgesine verimli bir şekilde gönderilmesini sağlar.

Aşağıdaki özel değeri kullanabilirsiniz:

MQCODL_AS_INPUT

Çıkış uzunluğu, giriş uzunluğuyla aynı.

Bağlantı oluşturulan programa geçirilen COMMAREA ' nın yeterli büyüklükte olmasını sağlamak için herhangi bir yanıt istenmese de bu değer gerekli olabilir.

FacilityKeepTime (MQLONG) for MQCIH

FacilityKeep, kullanıcı işlemi sona erdikten sonra köprü olanağının alıkondduğu süreyi saniye cinsinden belirtir.

Sözde etkileşimli hareketler için, sözde etkileşimin beklenen süresine karşılık gelen bir değer belirtin; sözde etkileşimin son hareketi için sıfır belirtin ve diğer işlem tipleri için sıfır belirtin.

Bu alan, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri 0 'dır.

MQCIH için ADSDescriptor (MQLONG)

Bu alan, ADS tanımlayıcılarının SEND ve RECEIVE BMS isteklerinde gönderilip gönderilmeyeceğini belirten bir göstergedir.

Aşağıdaki değerler tanımlanır:

MQCADSD_NONE

ADS tanımlayıcılarını göndermeyin ya da almayın.

MQCADSD_SEND

ADS tanımlayıcılarını gönderin.

MQCADSD_RECV

ADS tanımlayıcılarını alın.

MQCADSD_MSGFORMAT

ADS tanımlayıcıları için ileti biçimini kullanın.

Bu, ADS tanımlayıcısının uzun biçimini kullanarak ADS tanımlayıcılarını gönderir ya da alır. Uzun form, 4 baytlık sınırlara hizalanmış alanlar içerir.

ADSDescriptor alanını aşağıdaki gibi ayarlayın:

- ADS tanımlayıcıları kullanmıyorsanız, alanı MQCADSD_NONE olarak ayarlayın.
- Her ortamda *aynı* CCSID ' ye sahip ADS tanımlayıcıları kullanıyorsanız, alanı MQCADSD_SEND ve MQCADSD_RECV toplamına ayarlayın.
- Her ortamda *farklı* CCSID ' leri olan ADS tanımlayıcıları kullanıyorsanız, alanı MQCADSD_SEND, MQCADSD_RECV ve MQCADSD_MSGFORMAT toplamına ayarlayın.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri MQCADSD_NONE.

MQCIH için ConversationalTask (MQLONG)

Bu alan, görevin daha fazla bilgi için istekte bulunmasına ya da görevi durdurup olağandışı bir ileti yayınlamasına izin verilip verilmeyeceğini belirten bir göstergedir.

Değer aşağıdaki seçeneklerden biri olmalıdır:

MQCCT_YES

Görev etkileşimli.

MQCCT_NO

Görev etkileşimli değil.

Bu alan, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri MQCCT_NO.

MQCIH için TaskEndDurumu (MQLONG)

Bu alan, görevin sonunda kullanıcı işleminin durumunu gösteren bir yanıt alanıdır. Alan yalnızca 3270 hareketleri için kullanılır ve ilk değeri MQCTES_NOSYNC olur.

Aşağıdaki değerlerden biri döndürülür:

MQCTES_NOSYNC

Senkronize değil.

Kullanıcı işlemi henüz tamamlanmadı ve eşitlenmedi. MQMD ' deki *MsgType* alanı bu durumda MQMT_REQUEST.

MQCTES_COMMIT

Kesinleştirme iş birimi.

Kullanıcı hareketi henüz tamamlanmadı, ancak ilk iş birimini eşitlendi. MQMD 'deki *MsgType* alanı bu durumda MQMT_DATAGRAM' dir.

MQCTES_BACKOUT

Geri çekilin.

Kullanıcı hareketi henüz tamamlanmadı. Yürürlükteki iş birimi geriletildi. MQMD 'deki *MsgType* alanı bu durumda MQMT_DATAGRAM' dir.

MQCTES_ENDTASK

Görevi sona erdir.

Kullanıcı hareketi sona erdi (ya da olağandışı bitti). MQMD ' deki *MsgType* alanı, bu durumda MQMT_REPLY.

MQCIH için tesis (MQBYTE8)

Bu alan, 8 baytlık köprü olanağı simgesini gösterir.

Köprü olanağı simgesi, sözde etkileşimde birden çok işlemin aynı köprü olanağını (sanal 3270 uçbirimi) kullanmasını sağlar. Sözde etkileşimde birinci ya da tek iletide MQCFAC_NONE değerini ayarlayın. Bu değer, CICS ' e bu ileti için yeni bir köprü olanağı ayırmasını bildirir. Giriş iletisinde sıfır dışında bir *FacilityKeepTime* belirtildiğinde yanıt iletilerinde bir köprü olanağı simgesi döndürülür. Bir sözde etkileşim içindeki sonraki giriş iletilerinin aynı köprü olanağı simgesini kullanması gerekir.

Aşağıdaki özel değer tanımlandı:

MQCFAC_NONE

Tesis simgesi belirtilmedi.

C programlama dili için, MQCFAC_NONE_ARRAY değişmezi de tanımlanır ve MQCFAC_NONE ile aynı değere sahiptir, ancak dizgi yerine bir karakter dizisidir.

Bu alan, yalnızca 3270 hareketleri için kullanılan bir istek ve yanıt alanıdır. Bu alanın uzunluğu MQ_FACILITY_LENGTH tarafından verilir. Bu alanın ilk değeri MQCFAC_NONE.

MQCIH için işlev (MQCHAR4)

Bu alan bir yanıt alanıdır. Bu alanın uzunluğu MQ_FUNCTION_LENGTH tarafından verilir. Bu alanın ilk değeri MQCFUNC_NONE.

Bu alanda döndürülen değer *ReturnCode* ; bkz. [Çizelge 476 sayfa 299](#). *Function* bir IBM MQ arama adı içerdiğinde aşağıdaki değerler kullanılabilir:

MQCFUNC_MQCONN

MQCONN çağrısı.

MQCFUNC_MQGET

MQGET çağrısı.

MQCFUNC_MQINQ

MQINQ çağrısı.

MQCFUNC_MQOPEN

MQOPEN çağrısı.

MQCFUNC_MQPUT

MQPUT çağrısı.

MQCFUNC_MQPUT1

MQPUT1 çağrısı.

MQCFUNC_NONE

Telefon yok.

Her durumda, C programlama dili için, MQCFUNC_*_ARRAY değişmezleri de tanımlanır; bu değişmezler, karşılık gelen MQCFUNC_* değişmezleriyle aynı değerlere sahiptir, ancak dizgiler yerine karakter dizileridir.

MQCIH için AbendCode (MQCHAR4)

AbendCode bir yanıt alanıdır. Bu alanın uzunluğu MQ_ABEND_CODE_LENGTH tarafından verilir. Bu alanın ilk değeri 4 boş karakterdir.

Bu alanda döndürülen değer, *ReturnCode* alanında MQCRC_APPLICATION_ABEND ya da MQCRC_BRIDGE_ABEND değeri varsa önemlidir. Varsa, *AbendCode* CICS ABCODE değerini içerir.

MQCIH için kimlik doğrulayıcı (MQCHAR8)

Bu alanın değeri, parola ya da geçiş biletidir.

CICS bridge için kullanıcı kimliği kimlik doğrulaması etkinse, *Authenticator*, iletiyi gönderenin kimliğini doğrulamak için MQMD kimlik bağlamındaki kullanıcı kimliğiyle birlikte kullanılır.

Bu bir istek alanıdır. Bu alanın uzunluğu MQ_AUTHENTICATOR_LENGTH tarafından verilir. Bu alanın ilk değeri 8 boşluktur.

MQCIH için Reserved1 (MQCHAR8)

Bu alan ayrılmış bir alandır. Değer 8 boşluk olmalıdır.

MQCIH için ReplyToBiçimi (MQCHAR8)

Bu alanın değeri, yürürlükteki iletiye yanıt olarak gönderilen yanıt iletisinin IBM MQ biçim adıdır.

Bu alanı kodlamaya ilişkin kurallar, MQMD 'deki *Format* alanını kodlamaya ilişkin kurullarla aynıdır.

Bu alan, yalnızca DPL programları için kullanılan bir istek alanıdır. Bu alanın uzunluğu MQ_FORMAT_LENGTH tarafından verilir. Bu alanın ilk değeri MQFMT_NONE.

MQCIH için RemoteSysTanıtıcısı (MQCHAR4)

Bu alanda, isteği işleyen CICS sisteminin CICS sistem tanıtıcısı gösterilir.

Bu alan boşsa, CICS sistem isteği köprü izleyiciyle aynı CICS sistemde işlenir. Kullanılan SYSID, Yanıt iletisinde döndürülür.

3270 sözde etkileşimi için, etkileşimde sonraki tüm iletiler ilk yanıtta döndürülen uzak SYSID değerini belirtmelidir. Belirtildiyse, SYSID aşağıdaki şekilde olmalıdır:

- Aktif ol.
- IBM MQ İstek kuyruğuna erişim yetkisi vardır.
- Köprü monitörünün CICS sisteminden CICS ISC bağlantılarına erişebilirsiniz.

MQCIH için RemoteTransTanıtıcısı (MQCHAR4)

Bu alan isteğe bağlı bir İstek alanıdır. Bu alanın uzunluğu, MQ_TRANSACTION_ID_LENGTH tarafından verilir.

Bu seçenek belirlenirse, alan CICS START değerinin RTRANSID değeri olarak kullanılır.

MQCIH için TransactionId (MQCHAR4)

Bu alan bir istek alanıdır. Uzunluğu MQ_TRANSACTION_ID_LENGTH tarafından verilir. Bu alanın ilk değeri dört boşluktur.

LinkType MQCLT_TRANSACTION değerine sahipse, *TransactionId* çalıştırılacak kullanıcı işleminin işlem tanıtıcısıdır; bu durumda boş olmayan bir değer belirtin.

LinkType MQCLT_PROGRAM değerine sahipse, *TransactionId* iş birimi içindeki tüm programların çalıştırılacağı hareket kodudur. Boş bir değer belirtirseniz, CICS DPL köprüsü varsayılan işlem kodu (CKBP) kullanılır. Değer boş değilse, ilk program CSQCBP00 olan bir yerel hareket olarak CICS olarak tanımlanmış olmanız gerekir. Bu alan yalnızca *UOWControl* MQCUOWC_FIRST ya da MQCUOWC_ONLY değerine sahip olduğunda geçerlidir.

FacilityLike (MQCHAR4)-MQCIH için

FacilityLike, köprü olanağı için model olarak kullanılacak kurulu bir uçbirimin adıdır.

Boşluk değeri, *FacilityLike* ögesinin köprü hareket profili tanımından alındığı ya da varsayılan bir değer kullanıldığı anlamına gelir.

Bu alan, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın uzunluğu MQ_FACILITY_LIKE_LENGTH tarafından verilir. Bu alanın ilk değeri dört boşluktur.

MQCIH için AttentionId (MQCHAR4)

Bu alandaki değer, işlem başlatıldığında AID anahtarının ilk değerini belirler. Sola hizalanmış 1 baytlık bir değerdir.

AttentionId , yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın uzunluğu MQ_ATTENTION_ID_LENGTH tarafından verilir. Bu alanın ilk değeri dört boşluktur.

MQCIH için StartCode (MQCHAR4)

Bu alanın değeri, köprünün bir uçbirim hareketine öykünme mi, yoksa START ile başlatılan bir harekete mi öyküneceğini belirten bir göstergedir.

Değer aşağıdakilerden biri olmalıdır:

MQCSC_START

Başlatma.

MQCSC_STARTDATA

Verileri başlatın.

MQCSC_TERMINPUT

Uçbirim girişi.

MQCSC_NONE

Yok.

Her durumda, C programlama dili için, MQCSC_ * _ARRAY değişmezleri de tanımlanır; bu değişmezler, karşılık gelen MQCSC_ * değişmezleriyle aynı değerlere sahiptir, ancak dizgiler yerine karakter dizileridir.

Köprüden gelen yanıtta bu alan, *NextTransactionId* alanında bulunan sonraki hareket tanıtıcısına uygun başlangıç koduna ayarlanır. Yanıtta aşağıdaki başlatma kodları olabilir:

- MQCSC_START
- MQCSC_STARTDATA
- MQCSC_TERMINPUT

CICS Transaction Server 1.2 için bu alan yalnızca bir istek alanıdır; yanıtta değeri tanımlanmamıştır.

CICS Transaction Server 1.3 ve sonraki yayınlar için bu alan hem bir istek hem de bir yanıt alanıdır.

Bu alan yalnızca 3270 hareketleri için kullanılır. Bu alanın uzunluğu MQ_START_CODE_LENGTH tarafından verilir. Bu alanın ilk değeri MQCSC_NONE.

MQCIH için CancelCode (MQCHAR4)

Bu alandaki değer, hareketi sonlandırmak için kullanılacak olağandışı bitiş kodudur (normalde daha fazla veri isteyen bir etkileşimli hareket). Ters durumda, bu alan boş olarak ayarlanır.

Bu alan, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın uzunluğu MQ_CANCEL_CODE_LENGTH tarafından verilir. Bu alanın ilk değeri dört boşluktur.

NextTransactionId (MQCHAR4) for MQCIH

Bu değer, kullanıcı hareketi tarafından döndürülen sonraki hareketin adıdır (genellikle EXEC CICS RETURN TRANSID tarafından). Sonraki hareket yoksa, bu alan boş olarak ayarlanır.

Bu alan, yalnızca 3270 hareketleri için kullanılan bir yanıt alanıdır. Bu alanın uzunluğu, MQ_TRANSACTION_ID_LENGTH tarafından verilir. Bu alanın ilk değeri dört boşluktur.

MQCIH için Reserved2 (MQCHAR8)

Bu alan ayrılmış bir alandır. Değer 8 boşluk olmalıdır.

MQCIH için Reserved3 (MQCHAR8)

Bu alan ayrılmış bir alandır. Değer 8 boşluk olmalıdır.

MQCIH için CursorPosition (MQLONG)

Bu alandaki değer, hareket başlatıldığında ilk imleç konumunu gösterir. Etkileşimli hareketlerde, imleç konumu RECEIVE vektöründe olur.

Bu alan, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQCIH_VERSION_2' den küçükse bu alan yoktur.

MQCIH için ErrorOffset (MQLONG)

ErrorOffset alanı, köprü çıkışı tarafından saptanan geçersiz verilerin konumunu gösterir. Bu alan, iletinin başlangıcından geçersiz verilerin konumuna kadar olan göreceli konumu sağlar.

ErrorOffset , yalnızca 3270 hareketleri için kullanılan bir yanıt alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQCIH_VERSION_2' den küçükse bu alan yoktur.

MQCIH için InputItem (MQLONG)

Bu alan ayrılmış bir alandır. Değer 0 olmalıdır.

Version , MQCIH_VERSION_2' den küçükse bu alan yoktur.

Reserved4 (MQCIH için MQLONG)

Bu alan ayrılmış bir alandır. Değer 0 olmalıdır.

Version , MQCIH_VERSION_2' den küçükse bu alan yoktur.

MQCMHO-İleti tanıtıcı seçenekleri yarat

MQCMHO yapısı, uygulamaların ileti tanıtıcılarının nasıl yaratılacağını denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, **MQCRTMH** çağrısındaki bir giriş parametresidir.

Kullanılabilirlik

MQCMHO yapısı aşağıdaki platformlarda bulunur:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

ve bu sistemlere bağlı IBM MQ MQI clients için.

Karakter kümesi ve kodlama

MQCMHO içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında olmalıdır (**MQENC_NATIVE**).

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 478. MQCMHO içindeki alanlar

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQCMHO_STRUC_ID	'CMHO'
Sürüm (yapı sürümü numarası)	MQCMHO_VERSION_1	1
Seçenekler (seçenekler)	MQCMHO_DEFAULT_VAL IDATION	0

Notlar:

1. C programlama dilinde, makro değişkeni MQCMHO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQCMHO MyCMHO = {MQCMHO_DEFAULT};
```

Dil bildirimleri

MQCMHO için C bildirimini

```
struct tagMQCMHO {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    Options;          /* Options that control the action of MQCRTMH */
};
```

MQCMHO için COBOL bildirimini

```
** 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.
```

MQCMHO için PL/I bildirimini

```
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 */
```

MQCMHO için High Level Assembler bildirimini

```
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)
```

MQCMHO için StrucId (MQCHAR4)

Bu, ileti yaratma tanıtıcısı seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQCMHO_STRUC_ID.

Değer şöyle olmalıdır:

MQCMHO_STRUC_ID

İleti tanıtıcısı yaratma seçenekleri yapısına ilişkin tanıtıcı.

C programlama dili için, **MQCMHO_STRUC_ID_ARRAY** sabiti de tanımlanır. Bu, **MQCMHO_STRUC_ID** ile aynı değere sahiptir, ancak bir dizgi yerine bir karakter dizisidir.

MQCMHO için Sürüm (MQLONG)

Bu alan her zaman bir giriş alanıdır. İlk değeri **MQCMHO_VERSION_1**' dir.

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQCMHO_VERSION_1

Version-1 ileti tanıtıcı seçenekleri yapısı yaratır.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCMHO_CURRENT_VERSION

İleti tanıtıcısı yaratma seçenekleri yapısının yürürlükteki sürümü.

MQCMHO için Seçenekler (MQLONG)

Bu alan her zaman bir giriş alanıdır. İlk değeri **MQCMHO_DEFAULT_VALIDATION**.

Aşağıdaki seçeneklerden biri belirtilebilir:

MQCMHO_VALIDATE

Bu ileti tanıtıcısında bir özellik ayarlamak için **MQSETMP** çağrıldığında, özellik adının geçerliliği denetlenir ve aşağıdakilerden emin olur:

- geçersiz karakter içermiyor.
- JMS ya da usr başlamaz. Aşağıdakiler dışında JMS :
 - JMSCorrelationID
 - JMSReplyTo
 - JMSType
 - JMSXGroupID
 - JMSXGroupSeq

Bu adlar JMS özellikleri için ayrılmıştır.

- büyük ya da küçük harflerin herhangi bir karışımında aşağıdaki anahtar sözcüklerden biri değildir:
 - VE
 - Arasında
 - Escape
 - YANLIŞ
 - Yazılım parçası
 - BUDUR
 - Bunun benzeri
 - DEĞİL
 - BOŞ DEĞERLİ
 - VEYA
 - DOĞRU
- Gövde başlamaz. ya da Kök. (Root.MQMD dışında).

Özellik MQ-defined (mq. *) ise Ad tanınır ve özellik tanımlayıcı alanları, özellik için doğru değerlere ayarlanır. Özellik tanınmazsa, özellik tanımlayıcısının *Support* alanı **MQPD_OPTIONAL** olarak ayarlanır.

MQCMHO_DEFAULT_VALIDATION

Bu değer, özellik adlarının varsayılan geçerlilik denetimi düzeyinin oluştuğunu belirtir.

Varsayılan geçerlilik denetimi düzeyi, **MQCMHO_VALIDATE** ile belirtilen düzeye eşdeğerdir.

Bu değer varsayılan değerdir.

MQCMHO_NO_VALIDATION

Özellik adı üzerinde geçerlilik denetimi gerçekleştirilmez. **MQCMHO_VALIDATE** açıklamasına bakın.

Varsayılan seçenek: Açıklanan önceki seçeneklerden hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

MQCMHO_NONE

Tüm seçenekler varsayılan değerlerini varsayar. Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın. **MQCMHO_NONE** yardımcı program belgeleri; bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

MQCNO-Bağlantı seçenekleri

MQCNO yapısı, uygulamanın kuyruk yöneticisine yönelik bağlantıyla ilgili seçenekleri belirtmesine olanak sağlar. Yapı, MQCONNX çağrısındaki bir giriş/çıkış değiştirgesidir.

Paylaşılan tanıtıcıları ve MQCONNX çağrısıyla ilgili daha fazla bilgi için bkz. [MQCONNX ile paylaşılan \(iş parçacığından bağımsız\) bağlantılar](#).

Kullanılabilirlik

MQCNO yapısı şu altyapılarda bulunur:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

Sürüm

Desteklenen programlama dilleri için sağlanan üstbilgi, COPY ve INCLUDE dosyaları, MQCNO ' nun en son sürümünü içerir, ancak *Version* alanının ilk değeri MQCNO_VERSION_1 olarak ayarlanmıştır. version-1 yapısında var olmayan alanları kullanmak için uygulamanın *Version* alanını gerekli sürüm numarasına ayarlaması gerekir.

Karakter kümesi ve kodlama

MQCNO içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında yer almalıdır. Ancak, uygulama IBM MQ MQI client olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 479. MQCNO için MQCNO içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQCNO_STRUC_ID	'CNO-'
Sürüm (yapı sürümü numarası)	MQCNO_VERSION_1	1
Seçenekler (MQCONNX işlemi denetleyen seçenekler)	MQCNO_NONE	0
Not: <i>Version</i> , MQCNO_VERSION_2' den küçükse kalan alanlar yoksayılr.		
ClientConnGörelı Konum (istemci bağlantısı için MQCD yapısının görelı konumu)	Yok	0
ClientConnPtr (istemci bağlantısı için MQCD yapısının adresi)	Yok	Boş değeri gösterge ya da boş değeri byte
Not: <i>Version</i> , MQCNO_VERSION_3' den küçükse kalan alanlar yoksayılr.		
ConnTag (kuyruk yöneticisi bağlantı etiketi)	MQCT_NONE	Boş Değerler
Not: <i>Version</i> , MQCNO_VERSION_4' den küçükse kalan alanlar yoksayılr.		
SSLConfigPtr (istemci bağlantısı için MQSCO yapısının adresi)	Yok	Boş değeri gösterge ya da boş değeri byte
SSLConfigOffset (istemci bağlantısı için MQSCO yapısının görelı konumu)	Yok	0
Not: <i>Version</i> , MQCNO_VERSION_5' den küçükse kalan alanlar yoksayılr.		
ConnectionId (benzersiz bağlantı tanıtıcısı)	Yok	Boş değeri gösterge ya da boş değeri byte
SecurityParmsGörelı Konumu (güvenlik değıştirgeleri için MQSCO yapısının görelı konumu)	Yok	Boş değeri gösterge ya da boş değeri byte
SecurityParmsPtr (güvenlik değıştirgeleri için MQSCO yapısının adresi)	Yok	Boş değeri gösterge ya da boş değeri byte
Not: <i>Version</i> , MQCNO_VERSION_6' dan küçükse kalan alanlar yoksayılr.		
Ayrılmış (ayrılmış alan)	Yok	Yapıyı 64 bitlik bir sınıra çıkarmak için ayrılmış alan.
CCDTUrlLength (CCDT URL uzunluğu)	Yok	CCDTUrlPtr ya da CCDTUrlOffset ile tanımlanan dizginin uzunluğu
CCDTUrlPtr (CCDT URL göstergesi)	Yok	Bağlantı için kullanılacak istemci bağlantı kanalı çizelgesinin yerini saptamak için, URLiçeren bir dizgiyi gösteren gösterge.

Çizelge 479. MQCNO için MQCNO içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>CCDTUrlOffset</u> (CCDT URL görelî konumu)	Yok	Bağlantı için kullanılacak istemci bağlantı kanalı çizelgesinin konumunu tanımlayan URL içeren bir dizginin bayt cinsinden görelî konumu.
Not: <i>Version</i> , MQCNO_VERSION_7' den küçükse kalan alanlar yoksayılr.		
<u>ApplName</u> (uygulama tarafından ayarlanan ad)	Yok	Kuyruk yöneticisine yönelik bağlantıyı tanımlamak için uygulama tarafından ayarlanan ad
<u>Reserved2</u> (ayrılmış alan)	Yok	Yapıyı 64 bitlik bir sınıra çıkarmak için ayrılmış alan.
<div style="background-color: #e0e0e0; padding: 2px; margin-bottom: 5px;"> V 9.3.0 V 9.3.0 </div> Not: <i>Version</i> , MQCNO_VERSION_8' den küçükse kalan alanlar yoksayılr.		
<div style="background-color: #e0e0e0; padding: 2px; margin-bottom: 5px;"> V 9.3.0 BalanceParms Görelî Konum </div>	Yok	MQBNO yapısına byte olarak görelî konum
<div style="background-color: #e0e0e0; padding: 2px; margin-bottom: 5px;"> V 9.3.0 BalanceParmsPtr </div>	Yok	MQBNO yapısının yerini gösteren gösterge
Notlar: <ol style="list-style-type: none"> – simgesi, tek bir boş karakteri temsil eder. C programlama dilinde, makro değişkeniMQCNO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın: <div style="background-color: #e0e0e0; padding: 10px; margin-top: 10px;"> <pre>MQCNO MycNO = {MQCNO_DEFAULT};</pre> </div>		

Dil bildirimleri

Not: V 9.3.0 Aşağıdaki bildirimlerin her birinde, CD kullanıcıları için IBM MQ 9.2.4 ve LTS kullanıcıları için IBM MQ 9.3.0 adresinde son iki satır (Offset of the MQBMO structure ve Address of the location of the MQBMO structure) eklenmiştir.

MQCNO için C bildirimi

```
typedef struct tagMQCNO MQCNO;
struct tagMQCNO {
    MQCHAR4    StructId;          /* Structure identifier */
    MQLONG     Version;          /* Structure version number */
    MQLONG     Options;          /* Options that control the action of
                                MQCONN * */
    MQLONG     ClientConnOffset; /* Offset of MQCD structure for client
                                connection */
    MQPTR      ClientConnPtr;    /* Address of MQCD structure for client
                                connection */
};
```

```

MQBYTE128 ConnTag; /* Queue managerconnection 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 */
};

```

MQCNO için COBOL bildirimi

```

** 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-CONNTAG 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

```

MQCNO için PL/I bildirimi

```

dcl
1 MQCNO 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 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 managerconnection 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 security parameters */
3 CCDUrlLength	fixed bin(31)	/* Length of string identified by <i>CCDUrlPtr</i> or <i>CCDUrlOffset</i> */
3 CCDUrlOffset	fixed bin(31)	/* Offset in bytes to URL of client connection channel */
3 CCDUrlPtr	pointer	/* Pointer to string containing URL */
3 Reserved	char(4)	/* Reserved field to pad out to 64 bit boundary */
3 ApplName	char(28)	/* Name set by the application to identify the connection to
		the queue manager */
3 Reserved2	char(4)	/* 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 */

MQCNO için High Level Assembler (Üst Düzey Çevirici) bildirimi

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_CCDURLLENGTH	DS F	Length of string identified by CCDURLPTR or CCDURLOFFSET	
* MQCNO_CCDURLOFFSET	DS F	Offset in bytes to URL of client connection channel	
MQCNO_CCDURLPTR	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	

MQCNO için Visual Basic bildirimi

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'
CCDUrlLength	As Long	'Length of string identified by <i>CCDUrlPtr</i> or <i>CCDUrlOffset</i> '
CCDUrlOffset	As Long	'Offset in bytes to URL of client connection channel'

CCDUrlPtr	As MQPTR	'Pointer to string containing URL'
Reserved	As MQBYTE4	'Reserved field to pad out to 64 bit boundary'
ApplName	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 MQBNO structure'
BalanceParmsPtr	As MQPTR	'Address of MQBNO structure'
End Type		

İlgili görevler

[MQCONN kullanılıyor](#)

MQCNO için StrucId (MQCHAR4)

Bu, bağlantı seçenekleri yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQCNO_STRUC_ID.

Değer şöyle olmalıdır:

MQCNO_STRUC_ID

Bağlantı seçenekleri yapısına ilişkin tanıttıcı.

C programlama dili için, MQCNO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değişmezin değeri MQCNO_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQCNO için Sürüm (MQLONG)

Sürüm her zaman bir giriş alanıdır. İlk değeri MQCNO_VERSION_1' dir.

Değer aşağıdakilerden biri olmalıdır:

MQCNO_VERSION_1

Version-1 bağlantı seçenekleri yapısı.

MQCNO_VERSION_2

Version-2 bağlantı seçenekleri yapısı.

MQCNO_VERSION_3

Version-3 bağlantı seçenekleri yapısı.

MQCNO_VERSION_4

Version-4 bağlantı seçenekleri yapısı.

MQCNO_VERSION_5

Version-5 bağlantı seçenekleri yapısı.

MQCNO_VERSION_6

Version-6 bağlantı seçenekleri yapısı.

MQCNO_VERSION_7

Version-7 bağlantı seçenekleri yapısı.

V 9.3.0 MQCNO_VERSION_8

Version-8 bağlantı seçenekleri yapısı.

Yalnızca yapının daha yeni sürümlerinde bulunan alanlar, alanların tanımlarında bu şekilde tanımlanır.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCNO_CURRENT_VERSION

Bağlantı seçenekleri yapısının yürürlükteki sürümü.

MQCNO için seçenekler (MQLONG)

MQCONNX işlemi denetleyen seçenekler.

Muhasebe seçenekleri

Aşağıdaki seçenekler, **AccountingConnOverride** kuyruk yöneticisi özneliği MQMON_ENABLED olarak ayarlanırsa muhasebe tipini denetler:

MQCNO_ACCOUNTING_MQI_ENABLED

MQIAccounting özniteliği MQMON_OFF olarak ayarlanarak kuyruk yöneticisi tanımlamasında izleme verilerinin toplanması geçersiz kılındığında, bu işaretin ayarlanması MQI muhasebe verilerinin toplanmasını etkinleştirir.

MQCNO_ACCOUNTING_MQI_DISABLED

MQIAccounting özniteliği MQMON_OFF olarak ayarlanarak kuyruk yöneticisi tanımlamasında izleme verilerinin toplanması geçersiz kılındığında, bu işaretin ayarlanması MQI muhasebe verilerinin toplanmasını durdurur.

MQCNO_ACCOUNTING_Q_ENABLED

Kuyruk yöneticisi tanımlamasında **MQIAccounting** özniteliğini MQMON_OFF olarak ayarlayarak kuyruk muhasebesi verilerinin toplanması geçersiz kılındığında, bu işaretin ayarlanması, kuyruk tanımlamasının *MQIAccounting* alanında bir kuyruk yöneticisi belirten kuyruklar için muhasebe verilerinin toplanmasını sağlar.

MQCNO_ACCOUNTING_Q_DISABLED

Kuyruk yöneticisi tanımlamasında **MQIAccounting** özniteliği MQMON_OFF olarak ayarlanarak kuyruk muhasebesi veri toplaması geçersiz kılındığında, bu işaret ayarlandığında, kuyruk tanımlamasının *MQIAccounting* alanında bir kuyruk yöneticisi belirten kuyruklar için muhasebe verileri toplaması kapatılır.

Bu işaretlerin hiçbiri tanımlanmazsa, bağlantıya ilişkin muhasebe, kuyruk yöneticisi özniteliklerinde tanımlandığı gibi olur.

Bağ tanımlama seçenekleri

Aşağıdaki seçenekler, kullanılacak IBM MQ bağ tanımının tipini denetler. Bu seçeneklerden yalnızca birini belirleyin:

MQCNO_STANDARD_BINDING

Uygulama ve yerel kuyruk yöneticisi aracı (kuyruğa alma işlemlerini yöneten bileşen) ayrı yürütme birimlerinde (genellikle ayrı işlemlerde) çalışır. Bu düzenleme, kuyruk yöneticisinin bütünlüğünü korur; yani, kuyruk yöneticisini hatalı programlardan korur.

Kuyruk yöneticisi birden çok bağ tanımlama tipini destekliyorsa ve MQCNO_STANDARD_BINDING ayarını tanımlarsanız, kuyruk yöneticisi gerçek bağ tanımlama tipini seçmek için `qm.ini` dosyasındaki `Connection` bölümündeki **DefaultBindType** özniteliğini kullanır. Bu bölüm tanımlı değilse ya da değer kullanılamıyorsa ya da uygulama için uygun değilse, kuyruk yöneticisi uygun bir bağ tanımlama tipi seçer. Kuyruk yöneticisi, bağlanma seçeneklerinde kullanılan gerçek bağ tanımlama tipini ayarlar.

Uygulamanın tam olarak sınanmamış olabileceği ya da güvenilir olmadığı ya da güvenilir olmadığı durumlarda MQCNO_STANDARD_BINDING kullanın. MQCNO_STANDARD_BINDING varsayılan değerdir.

Bu seçenek tüm ortamlarda desteklenir.

mqm kitaplığına bağlantı oluşturursanız, önce varsayılan bağ tanımlama tipini kullanarak standart bir sunucu bağlantısı kurma girişiminde bulunulur. Temeldeki sunucu kitaplığı yüklenemediyse, bunun yerine bir istemci bağlantısı denir.

- MQCONN (ya da MQCNO_STANDARD_BINDING belirtildiyse MQCONNX) davranışını değiştirmek için, MQ_CONNECT_TYPE ortam değişkenini aşağıdaki seçeneklerden birine ayarlayın. Bunun bir kural dışı durumu olduğunu unutmayın: MQ_CONNECT_TYPE LOCAL ya da STANDARD olarak ayarlanarak MQCNO_FASTPATH_BINDING belirtilirse, fastpath bağlantıları uygulamada ilgili bir değişiklik yapılmadan yönetici tarafından indirgenabilir.

<i>Çizelge 480. MQCONN ya da MQCONNX davranışını değiştiren MQ_CONNECT_TYPE değerleri</i>	
Değer	Anlamı
CLIENT	Yalnızca bir istemci bağlantısı kurma girişiminde bulunuldu.

Çizelge 480. MQCONN ya da MQCONNX davranışını değiştiren MQ_CONNECT_TYPE değerleri (devamı var)	
Değer	Anlamı
FastPath	Bu değer önceki yayınlarda destekleniyordu, ancak belirtildiyse şimdi yoksayılr.
LOCAL	Yalnızca sunucu bağlantısı denir. Fastpath bağlantıları standart bir sunucu bağlantısına indirgendi.
Standart	Önceki yayınlarla uyumluluk için desteklenir. Bu değer şimdi LOCAL olarak işlenir.

- MQCONNX çağrıldığında MQ_CONNECT_TYPE ortam değişkeni ayarlanmazsa, varsayılan bağ tanımlama tipini kullanan standart bir sunucu bağlantısı denir. Sunucu kitaplığı yüklenemezse, bir istemci bağlantısı denir.

MQCNO_FASTPATH_BINDING

Uygulama ve yerel kuyruk yöneticisi aracısı aynı yürütme biriminin bir parçasıdır. Bu, uygulamanın ve yerel kuyruk yöneticisi aracısının ayrı yürütme birimlerinde çalıştığı tipik bağlama yönteminin tersidir.

Kuyruk yöneticisi bu tip bir bağ tanımını desteklemiyorsa MQCNO_FASTPATH_BINDING yoksayılr; işlem, seçenek belirtilmemiş gibi devam eder.

MQCNO_FASTPATH_BINDING, birden çok sürecin uygulama tarafından kullanılan genel kaynaktan daha fazla kaynak tükettiği durumlarda avantajlı olabilir. Fastpath bağ tanımını kullanan bir uygulama, *güvenilir uygulama* olarak bilinir.

Fastpath bağlamasının kullanılıp kullanılmayacağına karar verirken aşağıdaki önemli noktaları göz önünde bulundurun:

- MQCNO_FASTPATH_BINDING seçeneğinin kullanılması, bir uygulamanın iletileri ve kuyruk yöneticisine ait diğer veri alanlarını değiştirmesini ya da bozmasını önlemez. Bu seçeneği yalnızca bu sorunları tam olarak değerlendirdiğiniz durumlarda kullanın.
- Uygulama, MQCNO_FASTPATH_BINDING ile zamanuyumsuz sinyaller ya da süreölçer kesintileri (sigkill gibi) kullanmamalıdır. Paylaşılan bellek parçalarının kullanımına ilişkin kısıtlamalar da vardır.
- Kuyruk yöneticisiyle bağlantıyı kesmek için uygulama MQDISC çağrısı kullanılmalıdır.
- Kuyruk yöneticisini endmqm komutuyla sona erdirmeden önce uygulama bitmelidir.

► **IBM i** IBM i' de iş, QMQADM grubuna ait bir kullanıcı tanıtımı altında çalıştırılmalıdır. Ayrıca, program olağandışı bir şekilde durmamalıdır, aksi takdirde öngörülemeyen sonuçlar ortaya çıkabilir.

► **Linux** ► **AIX** AIX and Linux sistemlerinde, mqm kullanıcı kimliği geçerli kullanıcı kimliği olmalı ve mqm grup tanıtıcısı geçerli grup tanıtıcısı olmalıdır. Uygulamanın bu şekilde çalışmasını sağlamak için, programı mqm kullanıcı kimliği ve mqm grup tanıtıcısına sahip olacak şekilde yapılandırın ve daha sonra, programda setuid ve setgid izin bitlerini ayarlayın.

IBM MQ Object Authority Manager (OAM), yetki denetimi için gerçek kullanıcı kimliğini kullanır.

► **Windows** Windows işletim tarihinde, programın mqm grubunun bir üyesi olması gerekir. Fastpath bağ tanımı 64 bit uygulamalar için desteklenmez.

MQCNO_FASTPATH_BINDING seçeneği aşağıdaki ortamlarda desteklenir:

- **AIX** AIX
- **IBM i** IBM i
- **Linux** Linux
- **Windows** Windows

► **z/OS** z/OS' da seçenek kabul edilir, ancak yoksayılr.

Güvenilir uygulamaların kullanılmasının etkileri hakkında daha fazla bilgi için bkz. [Güvenilen uygulamalara ilişkin kısıtlamalar](#).

MQCNO_SHARED_BINDING

MQCNO_SHARED_BINDING ile uygulama ve yerel kuyruk yöneticisi aracısı bazı kaynakları paylaşır. Kuyruk yöneticisi bu tip bir bağ tanımını desteklemiyorsa, MQCNO_SHARED_BINDING yoksayılr. İşlem, seçenek belirtilmemiş gibi devam eder.

MQCNO_ISOLATED_BINDING

Bu durumda, uygulama süreci ve yerel kuyruk yöneticisi aracısı, kaynakları paylaşmadıkları için birbirinden yalıtılır. Kuyruk yöneticisi bu tip bir bağ tanımını desteklemiyorsa, MQCNO_ISOLATED_BINDING yoksayılr. İşlem, seçenek belirtilmemiş gibi devam eder.

MQCNO_CLIENT_BINDING

Uygulamanın yalnızca istemci bağlantısı girişiminde bulunmasını sağlamak için bu seçeneği belirleyin. Bu seçenek aşağıdaki sınırlamaları içerir:

- ► **z/OS** MQCNO_CLIENT_BINDING z/OSüzerinde yoksayıldı.
- MQCNO_CLIENT_BINDING, MQCNO_STANDARD_BINDING dışında bir MQCNO bağlama seçeneğiyle belirtildiyse, MQRC_OPTIONS_ERROR ile reddedilir.
- Java ya da .NET için MQCNO_CLIENT_BINDING kullanılamıyor; bağ tanımlama tipini seçmek için kendi mekanizmaları var.

MQCNO_LOCAL_BINDING

Uygulamanın sunucu bağlantısı girişiminde bulunmasını sağlamak için bu seçeneği belirleyin. MQCNO_FASTPATH_BINDING, MQCNO_ISOLATED_BINDING ya da MQCNO_SHARED_BINDING da belirtilirse, bağlantı o tipte olur ve bu kısımda belgelenir. Ters durumda, varsayılan bağ tanımlama tipi kullanılarak standart bir sunucu bağlantısı kurma girişiminde bulunulur. MQCNO_LOCAL_BINDING aşağıdaki sınırlamaları içeriyor:

- ► **z/OS** MQCNO_LOCAL_BINDING z/OSüzerinde yoksayıldı.
- MQCNO_LOCAL_BINDING, MQCNO_RECONNECT_AS_DEF dışında bir MQCNO yeniden bağlanma seçeneğiyle belirtilirse MQRC_OPTIONS_ERROR ile reddedilir.
- Java ya da .NET için MQCNO_LOCAL_BINDING kullanılamıyor; bağ tanımlama tipini seçmek için kendi mekanizmaları var.

Aşağıdaki altyapılarda, kullanılan bağ tanımının tipini denetlemek için, Options alanıyla belirtilen bağ tanımlama tipiyle MQ_CONNECT_TYPE ortam değişkenini kullanabilirsiniz.

- ► **AIX** AIX
- ► **Linux** Linux
- ► **Windows** Windows

Bu ortam değişkenini belirtirseniz, ortam değişkeninin değeri FASTPATH ya da STANDARD olmalıdır; farklı bir değeri varsa yoksayılr. Ortam değişkeninin değeri büyük ve küçük harfe duyarlıdır; ek bilgi için MQCONNX ortam değişkenine bakın.

Ortam değişkeni ve Options alanı aşağıdaki gibi etkileşimde bulunur:

- Ortam değişkenini atlarsanız ya da ona desteklenmeyen bir değer verirsiniz, fastpath bağ tanımının kullanımı yalnızca Options alanı tarafından belirlenir.
- Ortam değişkenine desteklenen bir değer verirsiniz, fastpath bağ tanımı yalnızca hem ortam değişkeni hem de Options alanı fastpath bağ tanımını belirtiyorsa kullanılır.

Bağlantı etiketi seçenekleri

Bu seçenekler yalnızca bir z/OS kuyruk yöneticisine bağlanırken desteklenir ve ConnTagbağlantı etiketinin kullanımını denetler. Bu seçeneklerden yalnızca birini belirleyebilirsiniz.

Bağlantı etiketlerinin kesin uygulaması IBM MQ for z/OS ve IBM MQ for Multiplatformsarasında farklılık gösterir:

- ▶ **z/OS** **MQCNO_GENERATE_CONN_TAG**dışında aşağıdaki seçenekler yalnızca z/OS kuyruk yöneticisine bağlanırken desteklenir ve bağlantı etiketinin kullanımını denetler. Desteklenen seçeneklerden yalnızca birini belirtebilirsiniz.
- ▶ **ALW** **MQCNO_GENERATE_CONN_TAG** yalnızca z/OSdışındaki altyapılarda desteklenir.

▶ **ALW** **MQCNO_GENERATE_CONN_TAG**

Çıkış MQCNO yapısında, kuyruk yöneticisinin bu bağlantıyla ilişkilendirdiği bağlantı etiketini döndürür. Döndürülen bağlantı etiketi, kuyruk yöneticisinin tek bir uygulama yönetim ortamı olarak kabul ettiği tüm bağlantılar için aynı olacaktır.

▶ **z/OS** **MQCNO_SERIALIZE_CONN_TAG_Q_MGR**

Bu seçenek, yerel kuyruk yöneticisinde bağlantı etiketinin özel olarak kullanılmasını ister. Bağlantı etiketi yerel kuyruk yöneticisinde kullanıyorsa, MQCONNX çağrısı MQRC_CONN_TAG_IN_USE neden koduyla başarısız olur. Çağrı sonucu, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun başka bir yerinde bağlantı etiketi kullanılarak etkilenmez.

▶ **z/OS** **MQCNO_SERIALIZE_CONN_TAG_QSG**

Bu seçenek, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubu içinde bağlantı etiketinin özel olarak kullanılmasını ister. Bağlantı etiketi kuyruk paylaşım grubunda kullanıyorsa, MQCONNX çağrısı MQRC_CONN_TAG_IN_USE neden koduyla başarısız olur.

▶ **z/OS** **MQCNO_RESTRICT_CONN_TAG_Q_MGR**

Bu seçenek, yerel kuyruk yöneticisinde bağlantı etiketinin paylaşılan kullanımını ister. Bağlantı etiketi yerel kuyruk yöneticisinde kullanıyorsa, istekte bulunan uygulama, etiketin var olan kullanıcısıyla aynı işleme kapsamında çalışıyorsa MQCONNX çağrısı başarılı olabilir. Bu koşul karşılanmazsa, MQCONNX çağrısı MQRC_CONN_TAG_IN_USE neden koduyla başarısız olur. Çağrı sonucu, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun başka bir yerinde bağlantı etiketinin kullanılmasından etkilenmez.

- Bağlantı etiketini paylaşmak için uygulamaların aynı MVS adres alanında çalışması gerekir. Bağlantı etiketini kullanan uygulama bir istemci uygulamasıysa, MQCNO_RESTRICT_CONN_TAG_Q_MGR ' ye izin verilmez.

▶ **z/OS** **MQCNO_RESTRICT_CONN_TAG_QSG**

Bu seçenek, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubu içinde bağlantı etiketinin paylaşılan kullanımını ister. Bağlantı etiketi kuyruk paylaşım grubunda kullanılıyorsa, istekte bulunan uygulama aynı işleme kapsamında çalışıyorsa ve etiketin var olan kullanıcısıyla aynı kuyruk yöneticisine bağlıysa, MQCONNX çağrısı başarılı olabilir.

Bu koşullar karşılanmazsa, MQCONNX çağrısı MQRC_CONN_TAG_IN_USE neden koduyla başarısız olur.

- Bağlantı etiketini paylaşmak için uygulamaların aynı MVS adres alanında çalışması gerekir. Bağlantı etiketini kullanan uygulama bir istemci uygulamasıysa, MQCNO_RESTRICT_CONN_TAG_QSG ' ye izin verilmez.

Bu seçeneklerden hiçbiri belirtilmezse, ConnTag kullanılmaz. Version , MQCNO_VERSION_3' den küçükse bu seçenekler geçerli değildir.

İşle paylaşım seçenekleri

Multi

Bu seçenekler aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

Bunlar, aynı işlem içindeki farklı iş parçacıkları (koşut işleme birimleri) arasında tanıtıcı paylaşımını denetler. Aşağıdaki seçeneklerden yalnızca birini belirleyebilirsiniz:

MQCNO_HANDLE_SHARE_NONE

Bu seçenek, bağlantı ve nesne tanıtıcılarının yalnızca, tanıtıcının ayrılmasına neden olan iş parçacığı (MQCONN, MQCONNX ya da MQOPEN çağrısını yayınlayan iş parçacığı) tarafından kullanılabilceğini gösterir. Tutamaçlar, aynı işleme ait diğer iş parçacıkları tarafından kullanılamaz.

MQCNO_HANDLE_SHARE_BLOCK

Bu seçenek, bir işlemin bir iş parçacığı tarafından ayrılan bağlantı ve nesne tanıtıcılarının aynı işleme ait diğer iş parçacıkları tarafından kullanılabilceğini gösterir. Ancak, aynı anda yalnızca bir iş parçacığı belirli bir tutamacı kullanabilir; yani, yalnızca bir tutamacın seri kullanımına izin verilir. Bir iş parçacığı, başka bir iş parçacığı tarafından kullanılmakta olan bir tanıtıcıyı kullanmayı denerse, çağrı blokları (tutamaç kullanılabilir oluncaya kadar bekler).

MQCNO_HANDLE_SHARE_NO_BLOCK


Bu, MQCNO_HANDLE_SHARE_BLOCK ile aynıdır; ancak, tanıtıcı başka bir iş parçacığı tarafından kullanılırsa, çağrı, tanıtıcı kullanılabilir oluncaya kadar engelleme yerine MQCC_FAILED ve MQRC_CALL_IN_PROGRESS ile hemen tamamlanır.

Bir iş parçacığının sıfır ya da bir paylaşılmayan tanıtıcısı olabilir:

- MQCNO_HANDLE_SHARE_NONE belirten her MQCONN ya da MQCONNX çağrısı, birinci çağrıda yeni bir paylaşılmayan tanıtıcı ve ikinci ve sonraki çağrılarda aynı paylaşılmayan tanıtıcı (MQDISC çağrısına müdahale edilmediği varsayılarak) döndürür. Neden kodu, ikinci ve sonraki çağrılar için MQRC_ALREADY_CONNECTED ' dir.
- MQCNO_HANDLE_SHARE_BLOCK ya da MQCNO_HANDLE_SHARE_NO_BLOCK belirten her MQCONNX çağrısı, her çağrıda yeni bir paylaşılan tanıtıcı döndürür.

Nesne tanıtıcıları, nesne tanıtıcısını yaratan MQOPEN çağrısında belirtilen bağlantı tanıtıcısıyla aynı paylaşım özelliklerini devralır. Ayrıca, iş birimleri, iş birimini başlatmak için kullanılan bağlantı tanıtıcısıyla aynı paylaşım özelliklerini devralır; iş birimi, paylaşılan bir tanıtıcı kullanılarak bir iş parçacığında başlatılırsa, iş birimi aynı tanıtıcı kullanılarak başka bir iş parçacığında güncellenebilir.

Bir tanıtıcı paylaşım seçeneği belirlemezseniz, varsayılan değer ortam tarafından belirlenir:

-  Microsoft Transaction Server (MTS) ortamında varsayılan değer MQCNO_HANDLE_SHARE_BLOCK ile aynıdır.
- Diğer ortamlarda varsayılan değer MQCNO_HANDLE_SHARE_NONE ile aynıdır.

Yeniden bağlanma seçenekleri

Yeniden bağlanma seçenekleri, bir bağlantının yeniden bağlanabilir olup olmadığını belirler. Yalnızca istemci bağlantıları yeniden bağlanabilir.

MQCNO_RECONNECT_AS_DEF

Yeniden bağlanma seçeneği varsayılan değerine çözülür. Varsayılan değer ayarlanmazsa, bu seçeneğin değeri DEVRE DIŞI olarak çözülür. Seçeneğin değeri sunucuya geçirilir ve PCF ve MQSC tarafından sorgulanabilir.

MQCNO_RECONNECT

Uygulama, MQCONNX' in **QmgrName** değiştirgesinin değeriyle tutarlı herhangi bir kuyruk yöneticisine yeniden bağlanabilir. MQCNO_RECONNECT seçeneğini, istemci uygulamasıyla başlangıçta bağlantı kurduğu kuyruk yöneticisi arasında benzerlik yoksa kullanın. Seçeneğin değeri sunucuya geçirilir ve PCF ve MQSC tarafından sorgulanabilir.

MQCNO_RECONNECT_DISABLED

Uygulama yeniden bağlanamıyor. Seçeneğin değeri sunucuya iletilmedi.

MQCNO_RECONNECT_Q_MGR

Uygulama yalnızca, başlangıçta bağlandığı kuyruk yöneticisine yeniden bağlanabilir. Bir istemci yeniden bağlanabiliyorsa, ancak istemci uygulaması ile başlangıçta bağlantı kurduğu kuyruk yöneticisi arasında bir benzerlik varsa bu değeri kullanın. Bir istemcinin yüksek kullanılabilirlikli bir kuyruk yöneticisinin yedek yönetim ortamına otomatik olarak yeniden bağlanmasını istiyorsanız bu değeri seçin. Seçeneğin değeri sunucuya geçirilir ve PCF ve MQSC tarafından sorgulanabilir.

Yalnızca istemci bağlantıları için MQCNO_RECONNECT, MQCNO_RECONNECT_DISABLED ve MQCNO_RECONNECT_Q_MGR seçeneklerini kullanın. Seçenekler bir bağ tanımlama bağlantısı için kullanılıyorsa, MQCONNX tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_OPTIONS_ERROR ile başarısız olur. Otomatik istemci yeniden bağlantısı IBM MQ classes for Java tarafından desteklenmez

Etkileşim paylaşım seçenekleri

Aşağıdaki seçenekler yalnızca TCP/IP istemci bağlantıları için geçerlidir. SNA, SPX ve NetBios kanallarında bu değerler yoksayılr ve kanal, ürünün önceki sürümlerinde olduğu gibi çalışır.

MQCNO_NO_CONV_PAYLAŞIMI

Bu seçenek, sohbet paylaşımına izin vermez.

MQCNO_NO_CONV_PAYLAŞIM özelliğini, etkileşimlerin yoğun olarak yüklendiği ve bu nedenle çekişmenin, paylaşım etkileşimlerinin bulunduğu kanal eşgörünümünün sunucu bağlantısı ucunda bir olasılık olduğu durumlarda kullanabilirsiniz. MQCNO_NO_CONV_SHARING, etkileşim paylaşımını destekleyen bir kanala bağlanıldığında SHARECNV (1) ve etkileşim paylaşımını desteklemeyen bir kanala bağlanıldığında SHARECNV (0) gibi davranır.

MQCNO_ALL_CONVS_SHARE

Bu seçenek etkileşim paylaşımına izin verir; uygulama, kanal örneğindeki bağlantı sayısına herhangi bir sınır koymaz. Bu seçenek varsayılan değerdir.

Uygulama kanal eşgörünümünün paylaşılabilirliğini belirtiyorsa, ancak kanalın sunucu bağlantısı ucundaki *SharingConversations* (SHARECNV) tanımı bir olarak ayarlandıysa, paylaşım gerçekleşmez ve uygulamaya uyarı verilmez.

Benzer şekilde, uygulama paylaşımına izin verildiğini, ancak sunucu bağlantısı *SharingConversations* tanımının sıfır olarak ayarlandığını belirtiyorsa, uyarı verilmez ve uygulama, IBM WebSphere MQ 7.0 öncesi sürümlerde bir istemciyle aynı davranışı gösterir; paylaşım etkileşimleriyle ilgili uygulama ayarı yoksayılr.

MQCNO_NO_CONV_PAYLAŞIM ve MQCNO_ALL_CONVS_SHARE karşılıklı olarak birbirini dışlar. Belirli bir bağlantıda her iki seçenek de belirtilirse, bağlantı MQRC_OPTIONS_ERROR neden koduyla reddedilir.

Kanal tanımı seçenekleri

Aşağıdaki seçenekler, MQCNO ' da geçirilen kanal tanımlaması yapısının kullanımını denetler:

YALNIZCA MQCNO_CD_FOR_OUTPUT_ONLY

Bu seçenek, MQCNO ' daki kanal tanımlaması yapısının yalnızca başarılı bir MQCONNX çağrısında kullanılan kanal adını döndürmek için kullanılmasına izin verir.

Geçerli bir kanal tanımlama yapısı sağlanmazsa, çağrı MQRC_CD_ERROR neden koduyla başarısız olur.

Uygulama istemci olarak çalışmıyorsa, seçenek yoksayılr.

Döndürülen kanal adı, aynı kanal tanımlamasını kullanarak yeniden bağlanmak için MQCNO_USE_CD_SELECTION seçeneği kullanılarak sonraki bir MQCONNX çağrısında kullanılabilir. Bu, istemci kanal çizelgesinde birden çok uygulanabilir kanal tanımı olduğunda yararlı olabilir.

MQCNO_USE_CD_SELECTION

Bu seçenek, MQCONNX çağrısının MQCNO ' da geçirilen kanal tanımlaması yapısında bulunan kanal adını kullanarak bağlanmasına izin verir.

MQSERVER ortam değişkeni ayarlıysa, onun tanımladığı kanal tanımlaması kullanılır. MQSERVER ayarlanmazsa, istemci kanal çizelgesi kullanılır.

Kanal adı ve kuyruk yöneticisi adı eşleşen bir kanal tanımlaması bulunamazsa, çağrı MQRC_Q_MGR_NAME_ERROR neden koduyla başarısız olur.

Geçerli bir kanal tanımlama yapısı sağlanmazsa, çağrı MQRC_CD_ERROR neden koduyla başarısız olur.

Uygulama istemci olarak çalışmıyorsa, seçenek yoksayılr.

Varsayılan seçenek

Yukarıda açıklanan seçeneklerin hiçbirine gerek duymazsanız, aşağıdaki seçeneği kullanabilirsiniz:

MQCNO_NONE

Seçenek belirtilmedi.

Program belgelerine yardımcı olmak için MQCNO_NONE komutunu kullanın. Bu seçeneğin başka bir MQCNO_ * seçeneğiyle kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

ClientConnMQCNO için görelî konum (MQLONG)

ClientConnKayma, MQCNO yapısının başlangıcından itibaren bir MQCD kanal tanımlaması yapısının bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir. Bu alan, ilk değeri 0 olan bir giriş alanıdır.

ClientConnOffset komutunu yalnızca MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI clientolarak çalıştığıında kullanın. Bu alanın nasıl kullanılacağına ilişkin bilgi için *ClientConnPtr* alanının açıklamasına bakın.

Version , MQCNO_VERSION_2' den küçükse bu alan yoksayılr.

ClientConnMQCNO için Ptr (MQPTR)

ClientConnPtr bir giriş alanıdır. İlk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümüyle boş değerli bir byte dizgisidir.

ClientConnOffset ve *ClientConnPtr* öğelerini yalnızca MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI clientolarak çalıştığıında kullanın. Bu alanlardan birini ya da diğerini belirterek uygulama, gerekli değerleri içeren bir MQCD kanal tanımlaması yapısı sağlayarak istemci bağlantı kanalının tanımlamasını denetleyebilir.

Uygulama IBM MQ MQI clientolarak çalışıyorsa, ancak MQCD yapısı sağlamıyorsa, kanal tanımlamasını seçmek için MQSERVER ortam değışkeni kullanılır. MQSERVER ayarlanmazsa, istemci kanal çizelgesi kullanılır.

Uygulama IBM MQ MQI clientolarak çalışmıyorsa, *ClientConnOffset* ve *ClientConnPtr* yoksayılr.

Uygulama bir MQCD yapısı sağlarsa, listelenen alanları gerekli değere ayarlayın; MQCD 'deki diđer alanlar yoksayılr. Karakter dizilimlerini alan uzunluđuna kadar boşluklarla doldurabilir ya da boş karakterle sonlandırabilirsiniz. MQCD yapısındaki alanlarla ilgili ek bilgi için bkz. [“Alanlar” sayfa 1455](#).

Çizelge 481. MQCD 'deki Alanlar

MQCD 'deki Alan	Deđer
<i>ChannelName</i>	Kanal adı.
<i>Version</i>	Yapı sürümü numarası. MQCD_VERSION_7' den küçük olmamalıdır.
<i>TransportType</i>	Desteklenen herhangi bir iletim tipi.
<i>ModeName</i>	LU 6.2 kip adı.
<i>TpName</i>	LU 6.2 hareket programı adı.
<i>SecurityExit</i>	Kanal güvenlik çıkışının adı.
<i>SendExit</i>	Kanal gönderme çıkışının adı.
<i>ReceiveExit</i>	Kanal alma çıkışının adı.
<i>MaxMsgLength</i>	İstemci bağlantı kanalı üzerinden gönderilebilecek iletilerin bayt cinsinden uzunluk üst sınırı.
<i>SecurityUserData</i>	Güvenlik çıkışı için kullanıcı verileri.
<i>SendUserData</i>	Gönderme çıkışına ilişkin kullanıcı verileri.
<i>ReceiveUserData</i>	Alma çıkışı için kullanıcı verileri.
<i>UserIdentifier</i>	LU 6.2 oturumu oluşturmak için kullanılacak kullanıcı kimliği.
<i>Password</i>	LU 6.2 oturumu oluşturmak için kullanılacak parola.
<i>ConnectionName</i>	Bađlantı adı.
<i>HeartbeatInterval</i>	Sađlıklı iřletim bildirim akışları arasındaki saniye cinsinden süre.
<i>StrucLength</i>	MQCD yapısının uzunluđu.
<i>ExitNameLength</i>	<i>SendExitPtr</i> ve <i>ReceiveExitPtr</i> tarafından adreslenen çıkış adlarının uzunluđu. <i>SendExitPtr</i> ya da <i>ReceiveExitPtr</i> boş değeri gösterge olmayan bir değere ayarlanırsa sıfırdan büyük olmalıdır.
<i>ExitDataLength</i>	<i>SendUserDataPtr</i> ve <i>ReceiveUserDataPtr</i> tarafından adreslenen çıkış verilerinin uzunluđu. <i>SendUserDataPtr</i> ya da <i>ReceiveUserDataPtr</i> boş değeri gösterge olmayan bir değere ayarlanırsa sıfırdan büyük olmalıdır.
<i>SendExitsDefined</i>	<i>SendExitPtr</i> tarafından adreslenen gönderme çıkışlarının sayısı. Sıfır ise, <i>SendExit</i> ve <i>SendUserData</i> çıkış adını ve verilerini sağlar. Sıfırdan büyükse, <i>SendExitPtr</i> ve <i>SendUserDataPtr</i> çıkış adlarını ve verilerini sağlar ve <i>SendExit</i> ve <i>SendUserData</i> boş olmalıdır.
<i>ReceiveExitsDefined</i>	<i>ReceiveExitPtr</i> tarafından adreslenen alma çıkışlarının sayısı. Sıfır ise, <i>ReceiveExit</i> ve <i>ReceiveUserData</i> çıkış adını ve verilerini sağlar. Sıfırdan büyükse, <i>ReceiveExitPtr</i> ve <i>ReceiveUserDataPtr</i> çıkış adlarını ve verilerini sağlar ve <i>ReceiveExit</i> ve <i>ReceiveUserData</i> boş olmalıdır.
<i>SendExitPtr</i>	İlk gönderme çıkışının adının adresi.

Çizelge 481. MQCD 'deki Alanlar (devamı var)

MQCD 'deki Alan	Değer
<i>SendUserDataPtr</i>	İlk gönderme çıkışına ilişkin verilerin adresi.
<i>ReceiveExitPtr</i>	İlk alma çıkışının adresi.
<i>ReceiveUserDataPtr</i>	İlk alma çıkışına ilişkin verilerin adresi.
<i>LongRemoteUserIdLength</i>	Uzun uzak kullanıcı kimliğinin uzunluğu.
<i>LongRemoteUserIdPtr</i>	Uzun uzak kullanıcı kimliğinin adresi.
<i>RemoteSecurityId</i>	Uzak güvenlik tanıtıcısı.
<i>SSLCipherSpec</i>	TLS CipherSpec.
<i>SSLPeerNamePtr</i>	TLS eş adının adresi.
<i>SSLPeerNameLength</i>	TLS eş düzey adının uzunluğu.
<i>KeepAliveInterval</i>	Kanala ilişkin canlı tutma zamanlaması için iletişim yığınına geçirilen değer
<i>LocalAddress</i>	Kullanılacak yerel ağ bağdaştırıcısının IP adresi ve giden bağlantılar için kullanılacak bir kapı aralığı da içinde olmak üzere yerel iletişim adresi.

Kanal tanımlama yapısını şu iki yoldan biriyle sağlayın:

- *ClientConnOffset* görelî konum alanını kullanarak

Bu durumda, uygulama MQCNO içeren bir bileşik yapıyı ve ardından kanal tanımlaması yapısı MQCD 'yi bildirmeli ve *ClientConnOffset* 'yi MQCNO ' nun başlangıcından kanal tanımlaması yapısının görelî konumuna ayarlamalıdır. Bu görelî konumların doğru olduğundan emin olun. *ClientConnPtr* boş değerli göstergeye ya da boş değerli byte 'lara ayarlanmalıdır.

İşaretçi veri tipini desteklemeyen ya da gösterge veri tipini farklı ortamlara (örneğin, COBOL programlama dili) taşınmayacak şekilde uygulayan programlama dilleri için *ClientConnOffset* kullanın.

Visual Basic programlama dili için, MQCNOCD, CMQXB.BAS; bu yapı bir MQCNO yapısı ve ardından bir MQCD yapısı içerir. MQCNOCD_DEFAULTS alt yordamını çağırarak MQCNOCD ' yi kullanıma hazırlayın. MQCNOCD, MQCONNX çağrısının MQCONNXAny çeşitlemesi; daha fazla ayrıntı için MQCONNX çağrısının tanımına bakın.

- *ClientConnPtr* gösterge alanını kullanarak

Bu durumda, uygulama kanal tanımlama yapısını MQCNO yapısından ayrı olarak bildirebilir ve *ClientConnPtr* değerini kanal tanımlama yapısının adresine ayarlayabilir. *ClientConnOffset* değerini sıfıra ayarlayın.

İşaretçi veri tipini farklı ortamlara (örneğin, C programlama dili) taşınabilir bir şekilde destekleyen programlama dilleri için *ClientConnPtr* kullanın.

C programlama dilinde, MQCD_CLIENT_CONN_DEFAULT makro değişkenini kullanarak, MQCD_DEFAULT tarafından sağlanan ilk değerlerden daha uygun yapı için başlangıç değerleri sağlayabilirsiniz.

Hangi tekniği seçerseniz seçin, *ClientConnOffset* ve *ClientConnPtr* yöntemlerinden yalnızca birini kullanabilirsiniz; Her ikisi de sıfır değilse, çağrı MQRC_CLIENT_CONN_ERROR neden koduyla başarısız olur.

MQCONNX çağrısı tamamlandığında, MQCD yapısına yeniden gönderme yapılmaz.

Version , MQCNO_VERSION_2' den küçükse bu alan yoksayılır.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir ve ilk değer tümüyle boş değerli byte dizgisi olur.

ConnTag (MQBYTE128) for MQCNO on Multiplatforms

Bir bağlantı etiketi kavramsal olarak bir bağlantı tanıtıcısına benzer, ancak bunları tek bir uygulama örneği olarak tanımlayan birden çok ilgili bağlantıya yayılabilir. Multiplatforms üzerinde bağlantı etiketi, bağlantı sırasında kuyruk yöneticisi tarafından oluşturulur.

Daha fazla bilgi için bkz. [bağlantı tanıtıcısı](#) ve [uygulama örneği](#).

Oluşturulan bağlantı etiketleri semi kullanıcı tarafından okunabilir. Yani, bunlar MQSC ' de yerel karakter kümesindeki dizgiler gibi görüntülenebilir ve süzülebilir. IBM MQ tarafından ilgili olduğu bilinen bağlantılara otomatik olarak aynı bağlantı etiketi atanır. Bu atama, özellikle [uygulama dengeleme](#) için önemlidir.

Oluşturulan bağlantı etiketi üç şekilde görünür:

- [MQCNO_GENERATE_CONN_TAG](#) belirtildiğinde, bir MQCONNX çağrısına ilişkin çıkış MQCNO yapısında.
- [DISPLAY CONN](#) çıkışında (ya da programlı eşdeğerleri).
- [DISPLAY APSTATUS](#) çıkışında (ya da eşdeğerleri).

Uygulama sona erdiğinde ya da MQDISC çağrısı verildiğinde etiket geçerli olmaz.

İlgili başvurular

[“ConnTag \(MQBYTE128\) for MQCNO on IBM MQ for z/OS” sayfa 331](#)

Bir bağlantı etiketi kavramsal olarak bir bağlantı tanıtıcısına benzer, ancak bunları tek bir uygulama örneği olarak tanımlayan birden çok ilgili bağlantıya yayılabilir. IBM MQ for z/OS üzerinde bağlantı etiketi, uygulama tarafından sağlanan ve o uygulama eşgörünümünden bağlantıları diziselleştirmek için MQCNO_*_CONN_TAG seçenekleriyle birlikte kullanılan bir giriş alanıdır.

z/OS ConnTag (MQBYTE128) for MQCNO on IBM MQ for z/OS

Bir bağlantı etiketi kavramsal olarak bir bağlantı tanıtıcısına benzer, ancak bunları tek bir uygulama örneği olarak tanımlayan birden çok ilgili bağlantıya yayılabilir. IBM MQ for z/OS üzerinde bağlantı etiketi, uygulama tarafından sağlanan ve o uygulama eşgörünümünden bağlantıları diziselleştirmek için MQCNO_*_CONN_TAG seçenekleriyle birlikte kullanılan bir giriş alanıdır.

Bir uygulamanın eşzamanlı olarak bağlanması amaçlanan birden çok eşgörünümü varsa, her birinin bu alan için benzersiz bir değer sağlaması gerekir. Daha fazla ayrıntı için bu [bağlantı etiketi seçeneklerinin](#) açıklamalarına bakın.

Notlar:

- IBM MQ for z/OS üzerinde, yürütme sırasında bir uygulamayla ilişkili bağlantı etiketini denetimci olarak belirlemenin bir yolu yoktur.
- ASCII ya da EBCDIC ' de MQ ile başlayan bağlantı etiketi değerleri, IBM ürünleri tarafından kullanılmak üzere ayrılmıştır. Bu harflerle başlayan bağlantı etiketi değerlerini kullanmayın.

Etiket gerekmiyorsa aşağıdaki özel değeri kullanın:

MQCT_NONE

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQCT_NONE_ARRAY değişmezi de tanımlanır; bu değişmezin değeri MQCT_NONE ile aynı, ancak dizgi yerine bir karakter dizisi.

ConnTag alanı, bir z/OS kuyruk yöneticisine bağlanırken kullanılır.

Bu alanın uzunluğu MQ_CONN_TAG_LENGTH tarafından verilir. *Version* , MQCNO_VERSION_3' den küçükse bu alan yoksayılr.

Multi IBM MQ for Multiplatforms üzerindeki bağlantı etiketini kullanmaya ilişkin bilgi için bkz. [“ConnTag \(MQBYTE128\) for MQCNO on Multiplatforms” sayfa 331](#) .

MQCNO için SSLConfigPtr (PMQSCO)

SSLConfigPtr bir giriş alanıdır. İlk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümüyle boş değerli bir byte dizisidir.

SSLConfigPtr ve *SSLConfigOffset* komutlarını yalnızca MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI client olarak çalışıyorsa ve kanal protokolü TCP/IP ise kullanın. Uygulama IBM MQ istemcisi olarak çalışmıyorsa ya da kanal protokolü TCP/IP değilse, *SSLConfigPtr* ve *SSLConfigOffset* yoksayılr.

Uygulama, *SSLConfigPtr* ya da *SSLConfigOffset* ve *ClientConnPtr* ya da *ClientConnOffset* belirtilerek istemci bağlantısı için TLS kullanımını denetleyebilir. TLS bilgileri bu şekilde belirtildiğinde, MQSSLKEYR ve MQSSLCRYP ortam değişkenleri yoksayılr; istemci kanal tanımlama çizelgesindeki (CCDT) TLS ile ilgili bilgiler de yoksayılr.

TLS bilgileri yalnızca aşağıdakiler üzerinde belirtilebilir:

- İstemci işleminin ilk MQCONNX çağrısı ya da
- Kuyruk yöneticisine önceki tüm TLS bağlantıları MQDISC kullanılarak sona erdiğinde sonraki bir MQCONNX çağrısı.

Bunlar, süreç genelinde TLS ortamının başlatılabileceği tek durumlardır. TLS ortamı zaten var olduğunda TLS bilgilerini belirten bir MQCONNX çağrısı verilirse, çağrıdaki TLS bilgileri yoksayılr ve var olan TLS ortamı kullanılarak bağlantı kurulur; çağrı, bu durumda tamamlama kodu MQCC_WARNING ve neden kodu MQRC_SSL_ALREADY_KULLANIMA hazırlanır.

MQSCO yapısını, *SSLConfigPtr* içinde bir adres belirterek ya da *SSLConfigOffset* içinde bir görel konum belirterek MQCD yapısıyla aynı şekilde sağlayabilirsiniz; Bunun nasıl yapılacağına ilişkin ayrıntılar için *ClientConnPtr* açıklamasına bakın. Ancak, *SSLConfigPtr* ve *SSLConfigOffset* ürünlerinden en fazla birini kullanabilirsiniz; Çağrı MQRC_SSL_CONFIG_ERROR neden koduyla başarısız olur. Eğer ikisi de sıfır değilse.

MQCONNX çağrısı tamamlandıktan sonra, MQSCO yapısına yeniden gönderme yapılmaz.

Version , MQCNO_VERSION_4' den küçükse bu alan yoksayılr.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir.

SSLConfigOffset (MQCNO için MQLONG)

SSLConfigOffset , MQCNO yapısının başlangıcından itibaren bir MQSCO yapısının byte cinsinden görel konumudur. Görel konum pozitif ya da negatif olabilir. Bu alan, ilk değeri 0 olan bir giriş alanıdır.

SSLConfigOffset komutunu yalnızca MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI clientolarak çalıştığıında kullanın. Bu alanın nasıl kullanılacağına ilişkin bilgi için *SSLConfigPtr* alanının açıklamasına bakın.

Version , MQCNO_VERSION_4' den küçükse bu alan yoksayılr.

MQCNO için ConnectionId (MQBYTE24)

ConnectionId , IBM MQ ' in bir uygulamayı güvenilir bir şekilde tanımlamasına izin veren 24 baytlık benzersiz bir tanıtıcıdır. Bir uygulama PUT ve GET çağrılarında ilinti için bu tanıtıcıyı kullanabilir. Bu çıkış parametresi, tüm programlama dillerinde 24 boş bayt başlangıç değerine sahiptir.

Kuyruk yöneticisi tüm bağlantılara benzersiz bir tanıtıcı atar, ancak bunlar oluşturulur. Bir MQCONNX, sürüm 5 MQCNO ile bağlantı kurarsa, uygulama döndürülen MQCNO ' dan *ConnectionId* değerini saptayabilir. Atanan tanıtıcının, *CorrelId*, *MsgIDve* *GroupId* gibi IBM MQ tarafından oluşturulan diğer tüm tanıtıcılar arasında benzersiz olması garanti edilir.

PCF komutunu *Inquire Connection* ya da MQSC komutunu *DISPLAY CONN* kullanarak uzun süreli iş birimlerini tanımlamak için *ConnectionId* komutunu kullanın. MQSC komutları (CONN) tarafından kullanılan *ConnectionId* , burada döndürülen *ConnectionId* değerinden türetilir. PCF *Inquire and Stop Connection* komutları, burada döndürülen *ConnectionId* komutunu herhangi bir değişiklik yapmadan kullanabilir.

PCF komutunu Stop Connection (Bağlantıyı Durdur) ya da MQSC komutunu STOP CONN komutunu kullanarak ConnectionId değerini belirterek, uzun süreli bir iş biriminin sona ermesini zorlamak için ConnectionId komutunu kullanabilirsiniz. Bu komutların kullanılmasıyla ilgili ek bilgi için [Bağlantıyı Durdur](#) ve [STOP CONN](#) başlıklı konuya bakın.

Sürüm MQCNO_VERSION_5' den küçükse bu alan döndürülmez.

Bu alanın uzunluğu MQ_CONNECTION_ID_LENGTH tarafından verilir.

SecurityParmsMQCNO için görel konum (MQLONG)

SecurityParmsGörel Konumu, MQCNO yapısının başlangıcından itibaren MQCSP yapısının bayt cinsinden görel konumdur. Görel konum pozitif ya da negatif olabilir. Bu alan, ilk değeri 0 olan bir giriş alanıdır.

Sürüm MQCNO_VERSION_5' den küçükse bu alan yoksayılır.

MQCSP yapısı "[MQCSP-Güvenlik deęiřtirgeleri](#)" sayfa 334içinde tanımlanır.

SecurityParmsMQCNO için Ptr (PMQCSP)

SecurityParmsPtr, yetki hizmeti tarafından kimlik doęrulaması için bir kullanıcı kimlięi ve parola belirtmek üzere kullanılan MQCSP yapısının adresidir. Bu alan bir giriş alanıdır ve ilk değeri boş değeri bir gösterge ya da boş değeri byte 'tır.

Sürüm MQCNO_VERSION_5' den küçükse bu alan yoksayılır.

MQCSP yapısı "[MQCSP-Güvenlik deęiřtirgeleri](#)" sayfa 334içinde tanımlanır.

MQCNO için ayrılmıř (MQBYTE4)

Yapıyı 64 bitlik bir sınıra çıkarmak için ayrılmıř bir alan. Alanın ilk değeri, alanın uzunluęu için ikili sıfırdır.

Version , MQCNO_VERSION_6' dan küçükse bu alan yoksayılır.

MQCNO için CCDUrlLength (MQLONG)

CCDUrlLength , baęlantı için kullanılacak istemci baęlantı kanalı çizelgesinin yerini tanıtan bir URL içeren CCDUrlPtr ya da CCDUrlOffset ile tanıtılan dizginin uzunluęudur. Alanın ilk değeri sıfırdır.

CCDUrlLength komutunu, MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI clientolarak çalıştıęında kullanın.

Bu, [MQCHLLIB](#) ve [MQCHLTAB](#) ortam deęiřkenlerini ayarlamaya yönelik programlı bir alternatiftir.

Uygulama istemci olarak çalışmıyorsa, CCDUrlLength yoksayılır.

Version , MQCNO_VERSION_6' dan küçükse bu alan yoksayılır.

MQCNO için CCDUrlPtr (PMQCHAR)

CCDUrlPtr , baęlantı için kullanılacak istemci baęlantı kanalı çizelgesinin yerini saptamak için, URLiçeren bir dizginin isteęe baęlı göstergesidir. Bu alan, göstergeleri destekleyen programlama dillerinde boş değeri bir iřaretçinin ilk değeri ve tersi durumda tümü boş değeri bir byte dizgisi olan bir giriş alanıdır.

CCDUrlPtr komutunu, MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI clientolarak çalıştıęında kullanın.

Önemli: CCDUrlPtr ve CCDUrlOffsetürünlerinden yalnızca birini kullanabilirsiniz. Her iki alan da sıfır deęilse, çağrı MQRC_CCDT_URL_ERROR neden koduyla başarısız olur.

Bu, [MQCHLLIB](#) ve [MQCHLTAB](#) ortam deęiřkenlerini ayarlamaya yönelik programlı bir alternatiftir.

Uygulama istemci olarak çalışmıyorsa, CCDUrlPtr yoksayılır.

Version , MQCNO_VERSION_6' dan küçükse bu alan yoksayılır.

MQCNO için CCDUrlOffset (MQLONG)

CCDUrlOffset , MQCNO yapısının başlangıcından URL içeren bir dizgiye, baęlantı için kullanılacak istemci baęlantı kanalı çizelgesinin konumunu tanımlayan bayt cinsinden görel konumdur. Görel konum pozitif ya da negatif olabilir ve alanın ilk değeri sıfırdır.

CCDTU_r10ffset komutunu, MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI clientolarak çalıştığıında kullanın.

Önemli: CCDTU_r1Ptr ve CCDTU_r10ffsetürünlerinden yalnızca birini kullanabilirsiniz. Her iki alan da sıfır değilse, çağrı MQRC_CCDT_URL_ERROR neden koduyla başarısız olur.

Bu, MQCHLLIB ve MQCHLTAB ortam değişkenlerini ayarlamaya yönelik programlı bir alternatiftir.

Uygulama istemci olarak çalışmıyorsa, CCDTU_r10ffset yoksayılır.

Version , MQCNO_VERSION_6' dan küçükse bu alan yoksayılır.

MQCNO için ApplName (MQCHAR28)

Kuyruk yöneticisiyle bağlantıyı tanımlamak için uygulama tarafından ayarlanan ad. Alanın ilk değeri MQAN_NONE_ARRAY (boş karakterler).

Version , MQCNO_VERSION_7' den küçükse ya da değer boşluk olarak ayarlandıysa bu alan yoksayılır.

z/OS Bu alanı z/OSüzerinde ayarlayamazsınız. Bunu yapmayı denerseniz, MQRC_CNO_ERROR neden kodunu geri alırsınız.

MQCNO için Reserved2 (MQBYTE4)

Yapıyı 64 bitlik bir sınıra çıkarmak için ayrılmış bir alan. Alanın ilk değeri, alanın uzunluğu için ikili sıfırdır.

Version , MQCNO_VERSION_7' den küçükse bu alan yoksayılır.

V9.3.0 BalanceParmsMQCNO için Görelî Konum (MQLONG)

Uygulamanın dengeleme davranışına ilişkin bilgileri içeren MQBNO tipinde bir yapının bellek konumu. Uygulama bir istemci kanalı üzerinden bağlanmıyorsa, yapı tümüyle yoksayılır.

Version , MQCNO_VERSION_8' den küçükse bu alan yoksayılır.

Ek bilgi için [MQBNO](#) konusuna bakın.

Bu alanı sağlarsanız, “MQCNO için BalanceParmsPtr (MQPTR)” sayfa 334 alanını sağlayamazsınız. Her iki alanı da sağlama girişiminde bulunursanız, bir MQRC_CNO_ERROR alırsınız. Bu alan yalnızca istemci bağlantılarıyla ilgili olduğundan, bu alanın başka bir bağlantı tipinde sağlanması da MQRC_CNO_ERROR ile sonuçlanır.

V9.3.0 MQCNO için BalanceParmsPtr (MQPTR)

Uygulamanın dengeleme davranışına ilişkin bilgileri içeren MQBNO tipinde bir yapıya ilişkin bellek konumunu gösteren gösterge. Uygulama bir istemci kanalı üzerinden bağlanmıyorsa, yapı tümüyle yoksayılır.

Version , MQCNO_VERSION_8' den küçükse bu alan yoksayılır.

Ek bilgi için [MQBNO](#) konusuna bakın.

Bu alanı sağlarsanız, “BalanceParmsMQCNO için Görelî Konum (MQLONG)” sayfa 334 alanını sağlayamazsınız. Her iki alanı da sağlama girişiminde bulunursanız, bir MQRC_CNO_ERROR alırsınız. Bu alan yalnızca istemci bağlantılarıyla ilgili olduğundan, bu alanın başka bir bağlantı tipinde sağlanması da MQRC_CNO_ERROR ile sonuçlanır.

MQCSP-Güvenlik değıştirgeleri

IBM MQ bağlantı güvenliği parametre yapısı, kuyruk yöneticisine yönelik bir MQCONNX çağrısında kimlik doğrulama bilgilerini akışı için uygulamalar tarafından kullanılır. Bu anahtar, hassas verileri şifreleyen IBM MQ parola koruma sistemiyle kullanılan ilk anahtar sağlaması için de kullanabilirsiniz.

Sürüm 1 'deki kullanıcı kimliğini ve parolayı içermek için *AuthenticationType* değerini MQCSP_AUTH_USER_ID_AND_PWD olarak ayarlayın.

Sürüm 2 'de ilk anahtar bilgilerini sağladığınızda, *AuthenticationType* varsayılan olarak MQCSP_AUTH_NONEdeğerini alır.

V 9.3.4 IBM MQ 9.3.4' den kimlik doğrulama simgesi bilgilerini eklemek için *AuthenticationType* kullanın.

V 9.3.4 MQCSP_AUTH_USER_ID_AND_PWD ya da MQCSP_AUTH_ID_TOKEN işlevini kullanabilirsiniz, ancak ikisini birden kullanamazsınız.

Uyarı: Bazı durumlarda, bir istemci uygulamasına ilişkin MQCSP yapısındaki ya da kimlik doğrulama simgesi ağ üzerinden düz metin olarak gönderilir. İstemci uygulama parolalarının ve kimlik doğrulama simgelerinin uygun şekilde korunduğundan emin olmak için bkz. [MQCSP parola koruması](#).

Kullanılabilirlik

MQCSP yapısı desteklenen tüm IBM MQ altyapılarında kullanılabilir.

Sürüm

Desteklenen programlama dilleri için sağlanan üstbilgi, COPY ve INCLUDE dosyaları, MQCSP ' nin en son sürümünü içerir, ancak *Version* alanının ilk değeri MQCSP_VERSION_1 olarak ayarlanmıştır. version-1 yapısında var olmayan alanları kullanmak için uygulamanın *Version* alanını gerekli sürüm numarasına ayarlaması gerekir.

Karakter kümesi ve kodlama

MQCSP ' deki veriler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır; bunlar sırasıyla **CodedCharSetId** kuyruk yöneticisi özneliği ve MQENC_NATIVE ile verilir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQCSP_STRUC_ID	'CSP-'
<u>Sürüm</u> (yapı sürümü numarası)	MQCSP_VERSION_1	1
<u>AuthenticationType</u> (kimlik doğrulama tipi)	Yok	MQCSP_AUTH_NONE
<u>Reserved1</u> (IBM üzerinde işaretçi hizalaması için gereklidir)	Yok	Boş değerli dizilim ya da boşluklar
<u>CSPUserIdPtr</u> (kullanıcı kimliğinin adresi)	Yok	Boş değerli gösterge ya da boş değerli byte
<u>CSPUserIdGörelili Konum</u> (kullanıcı kimliğinin görelili konumu)	Yok	0
<u>CSPUserIdLength</u> (kullanıcı kimliği uzunluğu)	Yok	0
<u>Reserved2</u> (IBM üzerinde işaretçi hizalaması için gereklidir)	Yok	Boş değerli dizilim ya da boşluklar
<u>CSPPasswordPtr</u> (parola adresi)	Yok	Boş değerli gösterge ya da boş değerli byte
<u>CSPPasswordOffset</u> (parolanın görelili konumu)	Yok	0
<u>CSPPasswordLength</u> (parola uzunluğu)	Yok	0
Not: <i>Version</i> MQCSP_VERSION_2 değerinden küçükse, kalan alanlar yoksayılr.		

Çizelge 482. MQCSP için MQCSP içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
V 9.3.0 V 9.3.0 Reserved3 (IBM üzerinde işaretçi hizalaması için gereklidir)	Yok	Boş değerli dizilim ya da boşluklar
V 9.3.0 V 9.3.0 InitialKeyPtr	Yok	Boş değerli gösterge ya da boşluklar
V 9.3.0 V 9.3.0 InitialKeyOffset (parola koruma sistemi için ilk anahtarın görelî konumu)	Yok	0
V 9.3.0 V 9.3.0 InitialKeyLength (parola koruma sistemi için ilk anahtarın uzunluğu)	Yok	0
Not: <i>Version</i> MQCSP_VERSION_3değerinden küçükse, kalan alanlar yoksayılr.		
V 9.3.4 Reserved4 (IBM üzerinde işaretçi hizalaması için gereklidir)	Yok	Boş değerli dizilim ya da boşluklar
V 9.3.4 TokenPtr (kimlik doğrulama simgesinin adresi)	Yok	Boş değerli gösterge ya da boşluklar
V 9.3.4 TokenKeyKayma (kimlik doğrulama belirtecinin görelî konumu)	Yok	0
V 9.3.4 TokenLength (kimlik doğrulama simgesinin uzunluğu)	Yok	0
Notlar:		
1. – simgesi, tek bir boş karakteri temsil eder.		
2. C programlama dilinde, makro değışkeniMQCSP_DEFAULT, çizelgede listelenen değeri içerir. Yapıdaki alanlar için başlangıç değeri sağlamak üzere aşağıdaki şekilde kullanılabilir:		
<pre>MQCSP MyCSP = {MQCSP_DEFAULT};</pre>		

Dil bildirimleri

MQCSP için C bildirim

```
V 9.3.0 V 9.3.0
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 */
};

```

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 */
};

```

MQCSP için COBOL bildirimi

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 **

```

MQCSP için PL/I bildirimi

```

V 9.3.0  V 9.3.0
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 */

```

```

V 9.3.4
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 */

```

MQCSP için Visual Basic bildirimi

```

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

```

İlgili kavramlar

Kimlik doğrulama belirteçleriyle çalışma

MQCSP için StrucId (MQCHAR4)

Bu, güvenlik parametreleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQCSP_STRUC_ID.

Değer şöyle olmalıdır:

MQCSP_STRUC_ID

Güvenlik parametreleri yapısına ilişkin tanıtıcı.

C programlama dili için, değişmez MQCSP_STRUC_ID_ARRAY de tanımlanır. Bu değer MQCSP_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQCSP için Sürüm (MQLONG)

MQCSP yapısı sürüm numarası.

Değer şöyle olmalıdır:

MQCSP_VERSION_1

Version-1 güvenlik parametreleri yapısı. Sürüm 1 'de, kuyruk yöneticisiyle kimlik doğrulamak için MQCSP yapısına bir kullanıcı kimliği ve parola ekleyebilirsiniz.

MQCSP_VERSION_2

Version-2 güvenlik parametreleri yapısı. Sürüm 2 'de, kuyruk yöneticisiyle kimlik doğrulamak için bir kullanıcı kimliği ve parola ekleyebilir ve parolaları korumak için kullanılan ilk anahtar belirtebilirsiniz.

V 9.3.4 MQCSP_VERSION_3

Version-3 güvenlik parametreleri yapısı. Sürüm 3 'te, kuyruk yöneticisiyle kimlik doğrulaması yapmak için MQCSP yapısına bir kullanıcı kimliği ve parola ya da kimlik doğrulama simgesi ekleyebilirsiniz. Parolaları korumak için kullanılan ilk anahtarı da belirtebilirsiniz.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCSP_CURRENT_VERSION

Güvenlik parametreleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri: MQCSP_VERSION_3.

MQCSP için AuthenticationType (MQLONG)

AuthenticationType bir giriş alanıdır. İlk değeri MQCSP_AUTH_NONE.

Bu, gerçekleştirilecek kimlik doğrulama türüdür. Geçerli değerler şunlardır:

MQCSP_AUTH_NONE

Kullanıcı kimliği ve parola ya da kimlik doğrulama simgesi alanlarını kullanmayın.

MQCSP_AUTH_USER_ID_AND_PWD

MQCSP yapısında kullanıcı kimliğini ve parolayı kullanarak kimlik doğrulaması yapın.

V 9.3.4 MQCSP_AUTH_ID_TOKEN

MQCSP yapısındaki kimlik doğrulama simgesini kullanarak kimlik doğrulamasını yapın.

Varsayılan değer MQCSP_AUTH_NONE değeridir. Varsayılan ayarda parola koruması gerçekleştirilmez.

Kimlik doğrulaması gerekiyorsa, **MQCSP** ayarını tanımlamanız gerekir. **AuthenticationType** - MQCSP_AUTH_USER_ID_AND_PWD ya da MQCSP_AUTH_ID_TOKEN.

Daha fazla bilgi için bkz. [MQCSP parola koruması](#).

İlgili kavramlar

[Kimlik doğrulama belirteçleriyle çalışma](#)

MQCSP için Reserved1 (MQBYTE4)

IBM üzerinde işaretçi hizalaması için gerekli ayrılmış bir alan.

Bu alanın başlangıç değerinin tümü boş.

CSPUserIdMQCSP için Ptr (MQPTR)

Kimlik doğrulamasında kullanılacak kullanıcı kimliğinin adresi.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizisidir. *Version* , MQCNO_VERSION_5' den küçükse bu alan yoksayılr.

Bu alan, kuyruk yöneticisinin CONNAUTH alanında bir IDPWOS **AUTHTYPE** değeri adlandırıldığında bir işletim sistemi kullanıcı kimliği içerebilir.

Windows üzerinde bu, tam olarak nitelenmiş bir etki alanı kullanıcı kimliği olabilir.

Kuyruk yöneticisinin CONNAUTH alanında bir IDPWLDAP **AUTHTYPE** değeri adlandırıldığında bu alan bir LDAP Kullanıcı Kimliği içerebilir.

CSPUserIdMQCSP için görelî konum (MQLONG)

Kimlik doğrulamasında kullanılacak kullanıcı kimliğinin bayt cinsinden görelî konumu. Görelî konum pozitif ya da negatif olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

CSPUserIdMQCSP için uzunluk (MQLONG)

Kimlik doğrulamasında kullanılacak kullanıcı kimliğinin uzunluğu.

Kullanıcı kimliğinin uzunluk üst sınırı altyapıya bağlıdır, bkz. [Kullanıcı Kimlikleri](#).

V 9.3.0 **V 9.3.0** Kullanıcı kimliğinin uzunluğu izin verilen uzunluk üst sınırından fazlaysa, kimlik doğrulama isteği MQR_CSP_ERROR ile başarısız olur. Önceki IBM MQ sürümlerinde, döndürülen hata MQR_NOT_AUTHORIZED hatasıdır.

Bu alan bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

MQCSP için Reserved2 (MQBYTE8)

IBM üzerinde işaretçi hizalaması için gerekli ayrılmış bir alan.

Bu alanın başlangıç değerinin tümü boş.

CSPPasswordPtr (MQCSP için MQPTR)

Kimlik doğrulamasında kullanılacak parolanın adresi.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizgisidir. *Version* , MQCNO_VERSION_5' den küçükse bu alan yoksayılr.

Bu alan, kuruluşa bağlı olarak işletim sistemi ya da LDAP parola denetimi tarafından reddedilen, ancak kimlik doğrulama yöntemine geçirilmeden önce IBM MQ tarafından reddedilmeyen boş bir parola içerebilir.

MQCSP için CSPPasswordOffset (MQLONG)

Bu, kimlik doğrulamasında kullanılacak parolanın bayt cinsinden görel konumudur. Görel konum pozitif ya da negatif olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

CSPPasswordLength (MQCSP için MQLONG)

Kimlik doğrulamasında kullanılacak parolanın uzunluğu.

Parolanın uzunluk üst sınırı MQ_CSP_PASSWORD_LENGTH, 256 karakterdir. **V 9.3.0** **V 9.3.0** Parolanın uzunluğu izin verilen uzunluk üst sınırından fazlaysa, kimlik doğrulama isteği MQR_CSP_ERROR ile başarısız olur. Önceki IBM MQ sürümlerinde, döndürülen hata MQR_NOT_AUTHORIZED hatasıdır.

Bu alan bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

V 9.3.0 V 9.3.0 MQCSP için Reserved3 (MQBYTE8)

IBM üzerinde işaretçi hizalaması için gerekli ayrılmış bir alan.

Bu alanın başlangıç değerinin tümü boş.

V 9.3.0 V 9.3.0 Multi MQCSP için InitialKeyPtr (MQPTR)

Parola koruma sistemine ilişkin ilk anahtarın adresi.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizgisidir. *Version* , MQCSP_VERSION_2' den küçükse bu alan yoksayılr.

Bu alan yalnızca IBM i, AIX, Linux, and Windows sistemlerinde çalışan IBM MQ MQI clients için geçerlidir.

IBM MQ MQI clients , IBM MQ parola koruma sistemini kullanarak bazı alanlar için şifrelenmiş değerler sağlayabilir. MQCSO yapısında belirtilen anahtar havuzuna ilişkin parolayı şifrelemek için bir ilk anahtar kullandıysanız, aynı istemci uygulaması için MQCSP ' ye ilk anahtar alanlarını eklediğinizden emin olun.

IBM MQ MQI istemcileri, IBM MQ parola koruma sistemini kullanarak bazı alanlar için şifrelenmiş değerler sağlayabilir. MQCSO yapısında belirtilen anahtar havuzuna ilişkin parolayı şifrelemek için bir ilk anahtar

kullandıysanız, aynı istemci uygulamasına ilişkin MQCSP yapısına ilk anahtar alanlarını eklediğinizden emin olun.

İlk anahtar, şifreleme algoritması tarafından bu değerleri şifrelemek ve bu değerlerin şifresini çözmek için kullanılır. Bu alanların değerleri **runmqicred** yardımcı programı kullanılarak şifrelendiğinde bir başlangıç anahtarı sağlanırsa, istemci kuyruk yöneticisine bağlandığında aynı başlangıç anahtarı belirtilmelidir.

Bu alanı kullanarak belirtilen ilk anahtar, *MQS_MQI_KEYFILE* ortam değişkeni ya da istemci yapılandırma dosyasının Security (Güvenlik) kısmına ilişkin *MQIInitialKeyFile* özelliği kullanılarak belirtilen ilk anahtarı geçersiz kılar.

İlk anahtarı belirtmek için *InitialKeyOffset* ya da *InitialKeyPtr* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız.

İlgili görevler

[AIX, Linux ve Windows üzerinde IBM MQ MQI istemcisi için başlangıç anahtarı sağlanması](#)
[IBM MQ bileşeni yapılandırma dosyalarındaki parolaları koruma](#)

İlgili başvurular

[mqicred \(IBM MQ istemci parolalarını koru\)](#)

[“KeyRepoPasswordPtr \(MQSCO için MQPTR\)” sayfa 560](#)

Bu, TLS anahtar havuzu parolası bayt cinsinden adresidir.

[“MQCSP için InitialKeyGörelili Konumu \(MQLONG\)” sayfa 342](#)

MQCSP yapısının başlangıcından itibaren parola koruma sistemine ilişkin ilk anahtarın bayt cinsinden görelili konumu. Görelili konum pozitif ya da negatif olabilir.

V 9.3.0 **V 9.3.0** **Multi** **MQCSP için InitialKeyGörelili Konumu (MQLONG)**

MQCSP yapısının başlangıcından itibaren parola koruma sistemine ilişkin ilk anahtarın bayt cinsinden görelili konumu. Görelili konum pozitif ya da negatif olabilir.

İlk anahtarı belirtmek için *InitialKeyOffset* ya da *InitialKeyPtr* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız. Daha fazla bilgi için *InitialKeyPtr* alanının açıklamasına bakın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir. *Version* , MQCSP_VERSION_2' den küçükse bu alan yoksayıılır.

İlgili görevler

[IBM MQ bileşeni yapılandırma dosyalarındaki parolaları koruma](#)

[AIX, Linux ve Windows üzerinde IBM MQ MQI istemcisi için başlangıç anahtarı sağlanması](#)

İlgili başvurular

[“MQCSP için InitialKeyPtr \(MQPTR\)” sayfa 341](#)

Parola koruma sistemine ilişkin ilk anahtarın adresi.

V 9.3.0 **V 9.3.0** **Multi** **MQCSP için InitialKeyLength (MQLONG)**

Parola koruma sistemine ilişkin ilk anahtarın uzunluğu.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir. *Version* , MQCSP_VERSION_2' den küçükse bu alan yoksayıılır.

İlgili görevler

[IBM MQ bileşeni yapılandırma dosyalarındaki parolaları koruma](#)

[AIX, Linux ve Windows üzerinde IBM MQ MQI istemcisi için başlangıç anahtarı sağlanması](#)

V 9.3.4 **MQCSP için Reserved4 (MQBYTE8)**

IBM üzerinde işaretçi hizalaması için gerekli ayrılmış bir alan.

Bu alanın başlangıç değerinin tümü boş.

V 9.3.4 **MQCSP için TokenPtr (MQPTR)**

Kuyruk yöneticisiyle kimlik doğrulaması için kullanılan kimlik doğrulama simgesinin adresi.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizgisidir. *Version* , MQCSP_VERSION_3' den küçükse bu alan yoksayılr.

Bu alan, AIX ya da Linux sistemlerinde çalışan IBM MQ MQI clients IBM MQ kuyruk yöneticilerine bağlanma ile ilgilidir.

Kimlik doğrulama simgesini belirtmek için *TokenOffset* ya da *TokenPtr* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız.

Daha fazla bilgi için [Uygulamada kimlik doğrulama belirteçlerini kullanmabaşlıklı konuya](#) bakın.

İlgili kavramlar

[Kimlik doğrulama belirteçleriyle çalışma](#)

İlgili başvurular

“MQCSP için TokenOffset (MQLONG)” sayfa 343

Bu, MQCSP yapısının başlangıcından itibaren kimlik doğrulama simgesine ilişkin bayt cinsinden görel konumdur. Görel konum pozitif ya da negatif olabilir.

V 9.3.4 MQCSP için TokenOffset (MQLONG)

Bu, MQCSP yapısının başlangıcından itibaren kimlik doğrulama simgesine ilişkin bayt cinsinden görel konumdur. Görel konum pozitif ya da negatif olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQCSP_VERSION_3' den küçükse bu alan yoksayılr.

Simgeyi belirtmek için *TokenOffset* ya da *TokenPtr* kullanabilirsiniz, ancak ikisini birden kullanamazsınız. Daha fazla bilgi için *TokenPtr* alanının açıklamasına bakın.

İlgili kavramlar

[Kimlik doğrulama belirteçleriyle çalışma](#)

İlgili görevler

[Uygulamada kimlik doğrulama belirteçlerini kullanma](#)

İlgili başvurular

“MQCSP için TokenPtr (MQPTR)” sayfa 342

Kuyruk yöneticisiyle kimlik doğrulaması için kullanılan kimlik doğrulama simgesinin adresi.

V 9.3.4 MQCSP için TokenLength (MQLONG)

Bu, kuyruk yöneticisiyle kimlik doğrulaması için kullanılan kimlik doğrulama simgesinin uzunluğudur.

Kimlik doğrulama simgesinin uzunluk üst sınırı 8192 byte olan MQ_CSP_TOKEN_LENGTH değeridir. *TokenLength* izin verilen uzunluk üst sınırından fazlaysa, kimlik doğrulama isteği MQRC_CSP_ERROR ile başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQCSP_VERSION_3' den küçükse bu alan yoksayılr.

Daha fazla bilgi için [Uygulamada kimlik doğrulama belirteçlerini kullanmabaşlıklı konuya](#) bakın.

İlgili kavramlar

[Kimlik doğrulama belirteçleriyle çalışma](#)

MQCTLO-Denetim geri çağırma seçenekleri yapısı

MQCTLO yapısı, bir denetim geri çağırma işleviyle ilgili seçenekleri belirtmek için kullanılır. Yapı, MQCTL çağırısındaki bir giriş ve çıkış deęiştirgesidir.

Kullanılabilirlik

MQCTLO yapısı aşağıdaki altyapılarda bulunur:

- **AIX** AIX
- **IBM i** IBM i
- **Linux** Linux
- **Windows** Windows
- **z/OS** z/OS

ve bu sistemlere bağı IBM MQ MQI clients için.

Sürüm

MQCTLO ' nun yürürlükteki sürümü MQCTLO_VERSION_1.

Karakter kümesi ve kodlama

MQCTLO içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında olmalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 483. MQCTLO ' daki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucID</u> (yapı tanıtıcısı)	MQCTLO_STRUC_ID	' CTLO '
<u>Sürüm</u> (yapı sürümü numarası)	MQCTLO_VERSION_1	1
<u>Seçenekler</u> (seçenekler)	MQCTLO_NONE	Boş Değerler
<u>Seçenekler</u> (ayrılmış alan)	Ayrılmış alan	
ConnectionArea (geri çağırma işlevinin kullanılacağı alan)	Yok	Boş değerli gösterge ya da boş değerli byte
<p>Notlar:</p> <p>1. C programlama dilinde, makro değişkeniMQCTLO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:</p> <pre>MQCTLO MyCTLO = {MQCTLO_DEFAULT};</pre>		

Dil bildirimleri

MQCTLO için C bildirimini

```
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 */
};
```

MQCTLO için COBOL bildirimi

```
** 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
```

MQCTLO için PL/I bildirimi

```
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 */
```

MQCTLO için StrucId (MQCHAR4)

Bu, denetim seçenekleri yapısının yapı tanıtcısıdır. Her zaman bir giriş alanıdır. Değeri MQCTLO_STRUC_ID.

Değer şöyle olmalıdır:

MQCTLO_STRUC_ID

Denetim seçenekleri yapısına ilişkin tanıtcı.

C programlama dili için, değişmez MQCTLO_STRUC_ID_ARRAY da tanımlanır. Bu değer MQCTLO_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQCTLO için Sürüm (MQLONG)

Denetim seçenekleri yapısı-Sürüm alanı

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQCTLO_VERSION_1

Version-1 Denetim seçenekleri yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCTLO_CURRENT_VERSION

Denetim seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQCTLO_VERSION_1' dir.

MQCTLO için Seçenekler (MQLONG)

Denetim seçenekleri yapısı-Seçenekler alanı

MQCTL işlemini denetleyen seçenekler.

MQCTLO_FAIL_IF QUIESCING

Kuyruk yöneticisi ya da bağlantı susturma durumundaysa MQCTL çağırısı başarısız yapmaya zorlayın.

MQCB çağırısında geçirilen MQGMO seçeneklerinde, ileti tüketicilerine susturulurken bildirim gönderilmesini sağlamak için MQGMO_FAIL_IF QUIESCING belirtin.

MQCTLO_THREAD_AFFINITY

Bu seçenek, uygulamaya, aynı bağlantı için tüm ileti tüketicilerinin aynı iş parçacığında çağrılmasını gerektirdiğini bildirir. Bu iş parçacığı, bağlantı durduruluncaya kadar tüketicilerin tüm çağrıları için kullanılır.

Varsayılan seçenek: Açıklanan seçeneklerden herhangi birine gerek duymuyorsanız, aşağıdaki seçeneği kullanın:

MQCTLO_NONE

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar. MQCTLO_NONE, program belgelerine yardımcı olmak üzere tanımlanır; bu seçeneğin başka bir öğeyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu bir giriş alanıdır. *Options* alanının ilk değeri MQCTLO_NONE.

MQCTLO için ayrılmış (MQLONG)

Bu ayrılmış bir alandır. Değer sıfır olmalıdır.

MQCTLO için ConnectionArea (MQPTR)

Denetim seçenekleri yapısı- ConnectionArea alanı

Bu, geri çağırma işlevinin kullanabileceği bir alandır.

Kuyruk yöneticisi bu alanın içeriğine dayalı olarak herhangi bir karar vermez ve bu, geri çağırmanın giriş parametresi olan MQCBC yapısındaki ConnectionArea alanına değiştirilmeden geçirilir.

Bu alan, MQOP_START ve MQOP_START_WAIT dışındaki tüm işlemler için yoksayılr.

Bu, geri çağırma işlevine ilişkin bir giriş ve çıkış alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

MQDH-Dağıtım üstbilgisi

MQDH yapısı, ileti bir iletim kuyruğunda saklanan bir dağıtım listesi iletileri olduğunda iletide bulunan ek verileri tanımlar. Dağıtım listesi iletileri, birden çok hedef kuyruğuna gönderilen bir iletidir. Ek veriler, MQDH yapısının ardından bir MQOR kayıtları dizisi ve bir MQPMR kayıtları dizisinden oluşur. Bu yapı, iletileri doğrudan iletim kuyruklarına koyan ya da iletileri iletim kuyruklarından (örneğin, ileti kanalı araçları) kaldiren özel uygulamalar tarafından kullanılır. Dağıtım listelerine ileti koymak isteyen uygulamalar bu yapıyı kullanmamalıdır. Bunun yerine, dağıtım listesindeki hedefleri tanımlamak için MQOD yapısını ve ileti özelliklerini belirtmek için MQPMO yapısını kullanmalı ya da tek tek hedeflere gönderilen iletilere ilişkin bilgileri almalıdır.

Kullanılabilirlik

MQDH yapısı aşağıdaki altyapılarda kullanılabilir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

Biçim Adı

MQFMT_DIST_HEADER

Karakter kümesi ve kodlama

MQDH ' deki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında yer almalıdır.

MQDH ' nin karakter kümesini ve kodlamasını aşağıdaki alanlardaki *CodedCharSetId* ve *Encoding* alanlarına ayarlayın:

- MQMD (MQDH yapısı ileti verilerinin başındadır) ya da
- MQDH yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Kullanım

Bir uygulama dağıtım listesine bir ileti koyduğunda ve hedeflerin bazıları ya da tümü uzak olduğunda, kuyruk yöneticisi uygulama ileti verilerini MQXQH ve MQDH yapılarıyla önekler ve iletiyi ilgili iletim kuyruğuna yerleştirir. Bu nedenle, ileti bir iletim kuyruğunda olduğunda veriler aşağıdaki sırada gerçekleşir:

- MQXQH yapısı
- MQDH yapısı ve MQOR ve MQPMR kayıtlarının dizileri
- Uygulama ileti verileri

Hedeflere bağlı olarak, kuyruk yöneticisi bu tür birden çok ileti oluşturabilir ve bunu farklı iletim kuyruklarına yerleştirebilir. Bu durumda, bu iletilerdeki MQDH yapıları, uygulama tarafından açılan dağıtım listesi tarafından tanımlanan hedeflerin farklı alt kümelerini tanımlar.

Dağıtım listesi iletilerini doğrudan bir iletim kuyruğuna koyan bir uygulama, daha önce açıklanan sıraya uymalı ve MQDH yapısının doğru olduğundan emin olmalıdır. MQDH yapısı geçerli değilse, kuyruk yöneticisi MQPUT ya da MQPUT1 çağrısından MQRC_DH_ERROR neden koduyla başarısız olabilir.

İletileri bir kuyruktaki dağıtım listesi biçiminde saklayabilirsiniz; ancak, kuyruğu dağıtım listesi iletilerini destekleyebilecek biçimde tanımladıysanız. “Kuyruklara ilişkin öznitelikler” sayfa 826’de açıklanan **DistLists** kuyruk özniteliğine bakın. Bir uygulama dağıtım listesi iletilerini dağıtım listelerini desteklemeyen bir kuyruğa doğrudan koyarsa, kuyruk yöneticisi dağıtım listesi iletilerini tek tek iletilere böler ve bunun yerine kuyruktaki iletilerini yerleştirir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQDH_STRUC_ID	'DH--'
<u>Sürüm</u> (yapı sürümü numarası)	MQDH_VERSION_1	1
<u>StrucLength</u> (MQDH yapısının uzunluğu ve aşağıdaki kayıtlar)	Yok	0
<u>Kodlama</u> (MQPMR kayıtları dizisini izleyen verilerin sayısal kodlaması)	Yok	0
<u>CodedCharSetId</u> (MQPMR kayıtları dizisini izleyen verilerin karakter kümesi tanıtıcısı)	MQCCSI_AYRILMAMIŞ	0
<u>Biçim</u> (MQPMR kayıtları dizisini izleyen verilerin biçim adı)	MQFMT_NONE	Boşluklar
<u>İşaretler</u> (genel işaretler)	MQDHF_NONE	0

Çizelge 484. MQDH için MQDH ' deki Alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
PutMsgRecFields (hangi MQPMM alanlarının var olduğunu gösteren işaretler)	MQPMRF_NONE	0
RecsPresent (var olan nesne kaydı sayısı)	Yok	0
ObjectRecGörelî Konum (MQDH ' nin başlangıcındaki ilk nesne kaydının görelî konumu)	Yok	0
PutMsgRecOffset (MQDH ' nin başlangıcından itibaren ilk koyma iletisi kaydının görelî konumu)	Yok	0

Notlar:

1. – simgesi, tek bir boş karakteri temsil eder.
2. C programlama dilinde, makro değişkeniMQDH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQDH MyDH = {MQDH_DEFAULT};
```

Dil bildirimleri

MQDH için C bildirimini

```
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 MQPMM records */
    MQLONG   Encoding;       /* Numeric encoding of data that follows
                             the MQOR and MQPMM records */
    MQLONG   CodedCharSetId; /* Character set identifier of data that
                             follows the MQOR and MQPMM records */
    MQCHAR8  Format;         /* Format name of data that follows the
                             MQOR and MQPMM records */
    MQLONG   Flags;          /* General flags */
    MQLONG   PutMsgRecFields; /* Flags indicating which MQPMM 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 MQPMM record from start
                             of MQDH */
};
```

MQDH için COBOL bildirimini

```
** 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 MQPMM records
15 MQDH-STRUCLNGTH PIC S9(9) BINARY.
** Numeric encoding of data that follows the MQOR and MQPMM records
15 MQDH-ENCODING PIC S9(9) BINARY.
** Character set identifier of data that follows the MQOR and MQPMM
** records
15 MQDH-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows the MQOR and MQPMM 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-RECSPRESENT    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.

```

MQDH için PL/I bildirimi

```

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 */

```

MQDH için Visual Basic bildirimi

```

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

```

MQDH için StrucId (MQCHAR4)

Bu, dağıtım üstbilgisi yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQDH_STRUC_ID.

Değer şöyle olmalıdır:

MQDH_STRUC_ID

Dağıtım üstbilgisi yapısına ilişkin tanıtıcı.

C programlama dili için, sabit MQDH_STRUC_ID_ARRAY da tanımlanır. Bu değer MQDH_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQDH için Sürüm (MQLONG)

Değer şöyle olmalıdır:

MQDH_VERSION_1

Dağıtım üstbilgisi yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQDH_CURRENT_VERSION

Dağıtım üstbilgisi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri MQDH_VERSION_1' dir.

MQDH için StrucLength (MQLONG)

MQDH yapısının başlangıcından MQOR ve MQPMR kayıtlarının dizilerini izleyen ileti verilerinin başlangıçlarına kadar olan byte sayısı. Veriler aşağıdaki sırada oluşur:

- MQDH yapısı
- MQOR kayıtları dizisi
- MQPMR kayıtları dizisi
- İleti Verileri

MQOR ve MQPMR kayıtlarının dizileri, MQDH yapısında bulunan görelî konumlar tarafından ele alınır. Bu görelî konumlar bir ya da daha fazla MQDH yapısı, kayıt dizileri ve ileti verileri arasında kullanılmayan byte 'lar ile sonuçlandıysa, kullanılmayan byte 'ların *StrucLength* değerine eklenmesi gerekir, ancak bu byte 'ların içeriği kuyruk yöneticisi tarafından korunmaz. MQPMR kayıtları dizisinin MQOR kayıtları dizisinden önce gelmesi için geçerlidir.

Bu alanın ilk değeri 0 'dır.

MQDH için kodlama (MQLONG)

Bu, MQOR ve MQPMR kayıtlarının dizilerini izleyen verilerin sayısal kodlamasıdır; MQDH yapısındaki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri 0 'dır.

CodedCharSetId (MQDH için MQLONG)

Bu, MQOR ve MQPMR kayıtlarının dizilerini izleyen verilerin karakter kümesi tanıtıcısıdır; MQDH yapısının kendisindeki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değeri kullanabilirsiniz:

MQCCSI_INHERIT

Bu yapının karakter kümesi tanıtıcısını edin.

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmazsa, MQGET çağrısı MQCCSI_INHERIT değerini döndürmez.

MQMD 'deki *PutAppLType* alanının değeri MQAT_BROKER ise MQCSI_INHERIT kullanılamaz.

Bu değer aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

Bu alanın ilk değeri MQCCSI_UNDEFINED 'dir.

MQDH için biçim (MQCHAR8)

Bu, MQOD ve MQPMR kayıtlarının dizilerini izleyen verilerin biçim adıdır (hangisi sonsa).

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD 'deki *Format* alanıyla aynıdır.

Bu alanın ilk değeri MQFMT_NONE.

MQDH için İşaretler (MQLONG)

Aşağıdaki işareti belirtebilirsiniz:

MQDHF_NEW_MSG_IDS

Dağıtım listesindeki her hedef için yeni bir ileti tanıtıcısı oluşturur. Bunu yalnızca bir koyma iletisi kaydı yoksa ya da kayıtlar varsa, ancak *MsgId* alanını içermediğinde ayarlayın.

Bu işaretin kullanılması, dağıtım listesi iletisinin tek tek iletilere ayrıldığı ana kadar ileti tanıtıcılarının oluşturulmasını erteler. Bu, dağıtım listesi iletisiyle akması gereken denetim bilgisi miktarını en aza indirir.

Bir uygulama bir iletiyi dağıtım listesine koyduğunda, kuyruk yöneticisi MQDH 'de MQDHF_NEW_MSG_IDS' yi aşağıdaki deyimlerin her ikisi de doğru olduğunda oluşturur:

- Uygulama tarafından sağlanan bir koyma iletisi kaydı yok ya da sağlanan kayıtlar *MsgId* alanını içermiyor.
- MQMD 'deki *MsgId* alanı MQMI_NONE ya da MQPMO içindeki *Options* alanı MQPMO_NEW_MSG_ID' yi içeriyor

İşaret gerekmezse, aşağıdakileri belirtin:

MQDHF_NONE

Hiçbir işaret belirtilmedi. MQDHF_NONE, program belgelerine yardımcı olmak için tanımlandı. Bu değişimin başka bir değişimle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu alanın ilk değeri MQDHF_NONE.

PutMsgRecFields (MQDH için MQLONG)

Aşağıdaki işaretlerin hiçbirini ya da daha fazlasını belirtebilirsiniz:

MQPMRF_MSG_ID

İleti tanıtıcısı alanı var.

MQPMRF_CORREL_ID

İlinti tanıtıcısı alanı var.

MQPMRF_GROUP_ID

Grup tanıtıcısı alanı var.

MQPMRF_FEEDBACK

Geribildirim alanı var.

MQPMRF_ACCOUNTING_TOKEN

Muhasebe simgesi alanı var.

MQPMR alanı yoksa, aşağıdakileri belirtin:

MQPMRF_NONE

Bir koyma iletisi kayıt alanı yok. MQPMRF_NONE, program belgelerine yardımcı olmak için tanımlandı. Bu değişimin başka bir değişimle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu alanın ilk değeri MQPMRF_NONE.

MQDH için RecsPresent (MQLONG)

Bu, varış noktası sayısıdır. Bir dağıtım listesi her zaman en az bir hedef içermelidir; bu nedenle *RecsPresent* her zaman sıfırdan büyük olmalıdır.

Bu alanın ilk değeri 0 'dır.

ObjectRecMQDH için görelî konum (MQLONG)

Bu, hedef kuyrukların adlarını içeren MQOR nesne kayıtları dizisindeki ilk kaydın bayt cinsinden görelî konumunu verir. Bu dizide *RecsPresent* kayıt var. Bu kayıtlar (artı ilk nesne kaydı ile önceki alan arasında atlanan bayt), *StrucLength* alanı tarafından belirtilen uzunluğa eklenir.

Bir dağıtım listesi her zaman en az bir hedef içermelidir; bu nedenle *ObjectRecOffset* her zaman sıfırdan büyük olmalıdır.

Bu alanın ilk değeri 0 'dır.

PutMsgRecOffset (MQDH için MQLONG)

Bu, ileti özelliklerini içeren MQPMR koyma iletisi kayıtları dizisindeki ilk kaydın bayt cinsinden görelî konumunu verir. Varsa, bu dizide *RecsPresent* kayıt vardır. Bu kayıtlar (artı ilk koyma iletisi kaydı ile önceki alan arasında atlanan byte 'lar), *StrucLength* alanı tarafından belirtilen uzunluğa eklenir.

Koyma iletisi kayıtları isteğe bağlıdır; herhangi bir kayıt sağlanmazsa, *PutMsgRecOffset* sıfırdır ve *PutMsgRecFields* MQPMRF_NONE değerine sahiptir.

Bu alanın ilk değeri 0 'dır.

MQDLH-Harf üstbilgisi

MQDLH yapısı, teslim edilmeyen iletiler (teslim edilmeyen ileti) kuyruğundaki iletilerin uygulama iletisi verilerinin önekini oluşturan bilgileri tanımlar. Kuyruk yöneticisi ya da ileti kanalı aracısı iletiyi kuyruğa yeniden yönlendirdiği için ya da bir uygulama iletisi doğrudan kuyruğa koyduğundan, ileti gönderilmeyen ileti kuyruğuna gelebilir.

Biçim Adı

MQFMT_DEAD_LETTER_HEADER

Karakter kümesi ve kodlama

MQDLH yapısındaki alanlar, *CodedCharSetId* ve *Encoding* alanları tarafından verilen karakter takımındaki ve kodlamasındaki alanlardır. Bunlar, MQDLH ' den önce gelen üstbilgi yapısında ya da MQDLH uygulama ileti verilerinin başlangıcında ise MQMD yapısında belirtilir.

Karakter takımı, kuyruk adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır.

Java/JMS için IBM MQ sınıflarını kullanıyorsanız ve MQMD ' de tanımlanan kod sayfası Java sanal makinesi tarafından desteklenmiyorsa, MQDLH UTF-8 karakter kümesine yazılır.

Kullanım

İletileri doğrudan ileti kuyruğuna koyan uygulamalar, ileti verilerine bir MQDLH yapısıyla önek eklemeli ve alanları uygun değerlerle kullanıma hazırlamalıdır. Ancak, kuyruk yöneticisi bir MQDLH yapısının var olmasını ya da alanlar için geçerli değerlerin belirtilmesini gerektirmez.

Bir ileti gelmeyen iletiler kuyruğuna konamayacak kadar uzunsa, uygulamanın aşağıdakilerden birini yapması gerekir:

- İleti verilerini, gelmeyen iletiler kuyruğuna sığacak şekilde kısaltın.

- İletiyi yardımcı belleğe kaydedin ve gitmeyen iletiler kuyruğuna bunu gösteren bir kural dışı durum raporu iletileri yerleştirin.
- İletiyi atın ve oluşturan bir hata döndürün. İleti kritik bir iletiyse (ya da önemli bir ileti olabiliyorsa), bunu yalnızca oluşturanın iletinin bir kopyasına sahip olduğu biliniyorsa yapın; örneğin, bir iletişim kanalından bir ileti kanalı aracısı tarafından alınan bir ileti.

Önceki işlemlerden hangisi uygunsuzsa (varsa), uygulamanın tasarımına bağlıdır.

Kuyruk yöneticisi, kesim olan bir ileti öndeki MQLH yapısıyla yerleştirildiğinde özel işlem gerçekleştirir; daha fazla ayrıntı için MQMDE yapısının tanımına bakın.

İletileri gönderilmeyen iletiler kuyruğuna koyma

Teslim edilmeyen ileti kuyruğuna bir ileti bulunduğunda, MQPUT ya da MQPUT1 çağrısı için kullanılan MQMD yapısı, aşağıdaki kural dışı durumlar dışında, iletiyle ilişkilendirilmiş MQMD ile (genellikle MQGET çağrısı tarafından döndürülen MQMD) aynı olmalıdır:

- *CodedCharSetId* ve *Encoding* alanlarını, MQLH yapısındaki alanlar için kullanılan karakter kümesi ve kodlamaya ayarlayın.
- Verilerin bir MQLH yapısıyla başladığını belirtmek için *Format* alanını MQFMT_DEAD_LETTER_HEADER olarak ayarlayın.
- Koşullara uygun bir bağlam seçeneği kullanarak bağlam alanlarını (*AccountingToken*, *AppIdentityData*, *AppOriginData*, *PutAppName*, *PutAppType*, *PutDate*, *PutTime*, *UserIdentifier*) ayarlayın:
 - Gitmeyen ileti kuyruğuna, önceki iletilerle ilgili olmayan bir iletiyi koyan bir uygulamanın MQPMO_DEFAULT_CONTEXT seçeneğini kullanması gerekir; bu, kuyruk yöneticisinin ileti tanımlayıcısındaki tüm bağlam alanlarını varsayılan değerlerine ayarlamasına neden olur.
 - Az önce aldığı bir ileti kuyruğuna konan bir sunucu uygulaması, özgün bağlam bilgilerini korumak için MQPMO_PASS_ALL_CONTEXT seçeneğini kullanmalıdır.
 - Yeni aldığı bir iletiye *yanıt* gönderen bir sunucu uygulaması, MQPMO_PASS_IDENTITY_CONTEXT seçeneğini kullanmalıdır; bu, kimlik bilgilerini korur, ancak kaynak bilgilerini sunucu uygulamasının kaynağı olacak şekilde ayarlar.
 - Bir ileti kanalı aracısı, iletişim kanalından aldığı bir ileti kuyruğuna, özgün bağlam bilgilerini korumak için MQPMO_SET_ALL_CONTEXT seçeneğini kullanmalıdır.

MQLH yapısının kendisinde, alanları aşağıdaki gibi ayarlayın:

- *CodedCharSetId*, *Encoding* ve *Format* alanlarını MQLH yapısını izleyen verileri (genellikle özgün ileti tanımlayıcısındaki değerleri) tanımlayan değerlere ayarlayın.
- *PutAppType*, *PutAppName*, *PutDate* ve *PutTime* bağlam alanlarını, iletiyi gitmeyen ileti kuyruğuna koyan uygulamaya uygun değerlere ayarlayın; bu değerler özgün iletiyle ilgili değildir.
- Diğer alanları uygun şekilde ayarlayın.

Tüm alanların geçerli değerlere sahip olduğunu ve karakter alanlarının alanın tanımlı uzunluğuna kadar boşluklarla doldurulması gerektiğini doğrulayın; kuyruk yöneticisi, boş ve sonraki karakterleri MQLH yapısında boşluklara dönüştürmediği için, karakter verilerini boş bir karakter kullanarak zamanından önce sona erdirmeyin.

Gitmeyen iletiler kuyruğundan ileti alınıyor

Gitmeyen iletiler kuyruğundan ileti alan uygulamalar, iletilerin bir MQLH yapısıyla başladığını doğrulamalıdır. Uygulama, ileti tanımlayıcısı MQMD 'deki *Format* alanını inceleyerek bir MQLH yapısının olup olmadığını saptayabilir; alan MQFMT_DEAD_LETTER_HEADER değerine sahipse, ileti verileri bir MQLH yapısıyla başlar. Ayrıca, uygulamaların gitmeyen iletiler kuyruğundan alacakları iletilerin kuyruk için çok uzun olması durumunda kesilebileceğini de unutmayın.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 485. MQDLH için MQDLH 'deki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQDLH_STRUC_ID	'DLH↵'
Sürüm (yapı sürümü numarası)	MQDLH_VERSION_1	1
Neden (ileti gitmeyen iletiler kuyruğuna neden geldi)	MQRN_NONE	0
DestQName (özgün hedef kuyruğun adı)	Yok	Boş değerli dizilim ya da boşluklar
DestQMgrAd (özgün hedef kuyruk yöneticisinin adı)	Yok	Boş değerli dizilim ya da boşluklar
Kodlama (MQDLH 'yi izleyen verilerin sayısal kodlaması)	Yok	0
CodedCharSetId (MQDLH 'yi izleyen verilerin karakter kümesi tanıtıcısı)	MQCCSI_AYRILMAMIŞ	0
Biçim (MQDLH 'yi izleyen verilerin biçim adı)	MQFMT_NONE	Boşluklar
PutApplTip (iletiyi teslim almayan ileti kuyruğuna koyan uygulamanın tipi)	Yok	0
PutApplAdı (iletiyi gönderilmeyen iletiler kuyruğuna koyan uygulamanın adı)	Yok	Boş değerli dizilim ya da boşluklar
PutDate (iletinin ölü mektup kuyruğuna konma tarihi)	Yok	Boş değerli dizilim ya da boşluklar
PutTime (iletinin ileti gönderilmeyen iletiler kuyruğuna konma süresi)	Yok	Boş değerli dizilim ya da boşluklar
Notlar: 1. ↵ simgesi, tek bir boş karakteri temsil eder. 2. Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir. 3. C programlama dilinde, makro değişkeniMQDLH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın: <pre>MQDLH MyDLH = {MQDLH_DEFAULT};</pre>		

Dil bildirimleri

MQDLH için C bildirimi

```
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
MQLONG    CodedCharSetId; /* Character set identifier of data that
follows MQLDH */
MQCHAR8   Format;        /* Format name of data that follows
MQLDH */
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 */
};

```

MQLDH için COBOL bildirimi

```

** MQLDH structure
10 MQLDH.
** Structure identifier
15 MQLDH-STRUCID PIC X(4).
** Structure version number
15 MQLDH-VERSION PIC S9(9) BINARY.
** Reason message arrived on dead-letter (undelivered-message) queue
15 MQLDH-REASON PIC S9(9) BINARY.
** Name of original destination queue
15 MQLDH-DESTQNAME PIC X(48).
** Name of original destination queue manager
15 MQLDH-DESTQMGRNAME PIC X(48).
** Numeric encoding of data that follows MQLDH
15 MQLDH-ENCODING PIC S9(9) BINARY.
** Character set identifier of data that follows MQLDH
15 MQLDH-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows MQLDH
15 MQLDH-FORMAT PIC X(8).
** Type of application that put message on dead-letter
(undelivered-message) queue
15 MQLDH-PUTAPPLTYPE PIC S9(9) BINARY.
** Name of application that put message on dead-letter
(undelivered-message) queue
15 MQLDH-PUTAPPLNAME PIC X(28).
** Date when message was put on dead-letter (undelivered-message)
queue
15 MQLDH-PUTDATE PIC X(8).
** Time when message was put on the dead-letter (undelivered-message)
queue
15 MQLDH-PUTTIME PIC X(8).

```

MQLDH için PL/I bildirimi

```

dcl
1 MQLDH 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 MQLDH */
3 CodedCharSetId fixed bin(31), /* Character set identifier of data
that follows MQLDH */
3 Format char(8), /* Format name of data that follows
MQLDH */
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

```

```

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 */

```

MQDLH için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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)

```

MQDLH için Visual Basic bildirimi

```

Type MQDLH
  StrucId      As String*4  'Structure identifier'
  Version      As Long      'Structure version number'
  Reason       As Long      'Reason message arrived on dead-letter'
                '(undelivered-message) queue'
  DestQName    As String*48 'Name of original destination queue'
  DestQMgrName As String*48 'Name of original destination queue'
                'manager'
  Encoding     As Long      'Numeric encoding of data that follows'
                'MQDLH'
  CodedCharSetId As Long    'Character set identifier of data that'
                'follows MQDLH'
  Format       As String*8   'Format name of data that follows MQDLH'
  PutApplType  As Long      'Type of application that put message on'
                'dead-letter (undelivered-message) queue'
  PutApplName  As String*28 'Name of application that put message on'
                'dead-letter (undelivered-message) queue'
  PutDate      As String*8   'Date when message was put on dead-letter'
                '(undelivered-message) queue'
  PutTime      As String*8   'Time when message was put on the'
                'dead-letter (undelivered-message) queue'
End Type

```

MQDLH için StrucId (MQCHAR4)

Bu, gönderilmeyen harf üstbilgisi yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQDLH_STRUC_ID.

Değer şöyle olmalıdır:

MQDLH_STRUC_ID

Ölü harf üstbilgisi yapısının tanıtıcısı.

C programlama dili için, MQDLH_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQDLH_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQDLH için Sürüm (MQLONG)

Sürüm, yapı sürümü numarasıdır.

Değer şöyle olmalıdır:

MQDLH_VERSION_1

Gönderilmeyen harf üstbilgi yapısına ilişkin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQDLH_CURRENT_VERSION

Ölü harf üstbilgi yapısının geçerli sürümü.

Bu alanın ilk değeri MQDLH_VERSION_1' dir.

MQDLH için neden (MQLONG)

Neden alanı, iletinin özgün hedef kuyruk yerine, gönderilmeyen iletiler kuyruğuna yerleştirilme nedenini tanımlar.

Bu, iletinin özgün hedef kuyruk yerine, gönderilmeyen iletiler kuyruğuna yerleştirilme nedenini tanımlar. MQFB_ * ya da MQRC_ * değerlerinden biri olmalıdır (örneğin, MQRC_Q_FULL). Oluşabilecek ortak MQFB_ * değerlerinin ayrıntıları için [“MQMD-İleti tanımlayıcı” sayfa 420](#) içindeki *Feedback* alanının tanımına bakın.

Değer MQFB_IMS_LAST ile MQFB_IMS_FIRST aralığındaysa, gerçek IMS hata kodu, *Reason* alanının değerinden MQFB_IMS_ERROR çıkarılarak belirlenebilir.

Bazı MQFB_ * değerleri yalnızca bu alanda oluşur. Bunlar, gitmeyen iletiler kuyruğuna aktarılan havuz iletileri, tetikleyici iletiler ya da iletim kuyruğu iletileriyle ilişkilidir. Bunlar:

MQFB_APPL_CANNOT_BE_STARTED (X'00000109')

Bir tetikleyici iletisini işleyen bir uygulama, tetikleyici iletisinin *AppId* alanında adı belirtilen uygulamayı başlatamadı (bkz. [“MQTM-Tetikleyici iletisi” sayfa 593](#)).

z/OSüzerinde CKTI CICS işlemi, tetikleyici iletileri işleyen bir uygulama örneğidir.

MQFB_APPL_TYPE_ERROR (X'0000010B')

Tetikleyici iletisinin *AppType* alanı geçersiz olduğundan, bir tetikleyici iletisini işleyen bir uygulama uygulamayı başlatamadı (bkz. [“MQTM-Tetikleyici iletisi” sayfa 593](#)).

z/OSüzerinde CKTI CICS işlemi, tetikleyici iletileri işleyen bir uygulama örneğidir.

MQFB_BIND_OPEN_CLUSRCVR_DEL (X'00000119')

İleti SYSTEM.CLUSTER.TRANSMIT.QUEUE , MQOO_BIND_ON_OPEN seçeneğiyle açılan bir küme kuyruğuna yönelik olarak tasarlandı, ancak iletiyi hedef kuyruğa iletmek için kullanılacak uzak küme alıcı kanalı, ileti gönderilmeden önce silindi. MQOO_BIND_ON_OPEN belirtildiğinden, iletiyi iletmek için yalnızca kuyruk açıldığında seçilen kanal kullanılabilir. Bu kanal artık kullanılmadığı için, ileti gönderilmeyen iletiler kuyruğuna yerleştirilir.

MQFB_NOT_A_REPOSITORY_MSG (X'00000118')

İleti bir havuz iletisi değil.

MQFB_DURDURED_BY_CHAD_EXIT (X'00000115')

İleti, kanal otomatik tanımlama çıkışı tarafından durduruldu.

MQFB_STOPPED_BY_MSG_EXIT (X'0000010D')

İleti, kanal ileti çıkışı tarafından durduruldu.

MQFB_TM_ERROR (X'0000010A')

MQMD' deki *Format* alanı MQFMT_TRIGGER ögesini belirtiyor, ancak ileti geçerli bir MQTM yapısıyla başlamıyor. Örneğin, *StrucId* animsaticı geçerli olmayabilir, *Version* tanınmamış olabilir ya da tetikleyici iletisinin uzunluğu MQTM yapısını içermek için yetersiz olabilir.

z/OSüzerinde CKTI CICS işlemi, tetikleyici iletileri işleyen ve bu geribildirim kodunu oluşturabilen bir uygulama örneğidir.

MQFB_XMIT_Q_MSG_ERROR (X'0000010F ')

Bir ileti kanalı aracı, iletim kuyruğundaki bir iletinin doğru biçimde olmadığını buldu. İleti kanalı aracı, bu geribildirim kodunu kullanarak iletiyi gönderilmeyen ileti kuyruğuna koyar.

Yaygın bir neden, bir iletinin doğrudan iletim kuyruğuna konmuş olması, bu nedenle iletinin beklenen XQH üstbilgisine sahip olmamasıdır. Uygulama MQXQH üstbilgisini oluşturmadıkça, iletiler uzak kuyruk üzerinden bir iletim kuyruğuna yerleştirilmelidir.

Bu alanın ilk değeri MQRC_NONE.

MQDLH için DestQName (MQCHAR48)

DestQName , iletinin özgün hedefi olan ileti kuyruğunun adıdır.

Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQDLH için DestQMgrAdı (MQCHAR48)

DestQMgrAd, iletinin özgün hedefi olan kuyruk yöneticisinin adıdır.

Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQDLH için kodlama (MQLONG)

Kodlama, MQDLH yapısını izleyen verilerin sayısal kodlamasıdır (genellikle özgün iletideki veriler); MQDLH yapısının kendisindeki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri şudur: 0.

CodedCharSetId (MQDLH için MQLONG)

CodedCharSetId , MQDLH yapısından akan verilerin karakter kümesi tanıtcısıdır (genellikle özgün iletideki veriler); MQDLH yapısının kendisindeki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değer kullanılabilir:

MQCCSI_INHERIT

Bu yapıyı izleyen verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı MQCCSI_INHERIT değerini döndürmez.

MQMD 'deki *PutAppType* alanının değeri MQAT_BROKER ise MQCSI_INHERIT kullanılamaz.

Bu değer aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

Bu alanın ilk değeri MQCCSI_UNDEFINED 'dir.

MQDLH için biçim (MQCHAR8)

Biçim, MQDLH yapısını izleyen verilerin biçim adıdır (genellikle özgün iletideki veriler).

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *Format* alanını kodlamaya ilişkin kurallarla aynıdır.

Bu alanın uzunluğu MQ_FORMAT_LENGTH tarafından verilir. Bu alanın ilk değeri MQFMT_NONE.

PutApplMQDLH için tip (MQLONG)

PutApplTipi, iletii teslim edilmeyen (teslim edilmeyen) kuyruğa koyan uygulama tipidir.

Bu alan, ileti tanımlayıcısı MQMD ' deki *PutApplType* alanıyla aynı anlama sahiptir (ayrıntılar için bkz. “MQMD-İleti tanımlayıcı” sayfa 420).

Kuyruk yöneticisi iletii teslim mektubu kuyruğuna yeniden yönlendirirse, *PutApplType* ' in MQAT_QMGR değeri vardır.

Bu alanın ilk değeri şudur: 0.

PutApplMQDLH için ad (MQCHAR28)

PutApplAdı, iletii teslim edilmeyen ileti kuyruğuna koyan uygulamanın adıdır.

Adın biçimi *PutApplType* alanına bağlıdır. Biçim, yayın düzeyinden yayına kadar değişiklik gösterebilir. “MQMD-İleti tanımlayıcı” sayfa 420 içindeki *PutApplName* alanının açıklamasına bakın.

Kuyruk yöneticisi iletii teslim etmeyen iletiler kuyruğuna yeniden yönlendirirse, *PutApplName* kuyruk yöneticisi adının ilk 28 karakterini içerir ve gerekirse boşluklarla doldurulur.

Bu alanın uzunluğu MQ_PUT_APPL_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C ' deki boş dizgi ve diğer programlama dillerindeki 28 boş karakterdir.

MQDLH için PutDate (MQCHAR8)

PutDate , iletinin teslim edilmeyen ileti kuyruğuna konma tarihidir.

Bu alanın kuyruk yöneticisi tarafından oluşturulduğu tarih için kullanılan biçim:

- YYYYAAAGG

karakterlerin temsil ettiği yer:

YYYY

yıl (dört sayısal basamak)

MM

yılın ayı (01-12)

GG

ayın günü (01-31)

Greenwich Ortalama Saati (GMT), sistem saatinin GMT ' ye doğru şekilde ayarlanmasına bağlı olarak *PutDate* ve *PutTime* alanları için kullanılır.

Bu alanın uzunluğu MQ_PUT_DATE_LENGTH tarafından verilir. Bu alanın ilk değeri, C ' deki boş dizgi ve diğer programlama dillerindeki sekiz boş karakterdir.

MQDLH için PutTime (MQCHAR8)

PutTime , iletinin teslim edilmeyen ileti kuyruğuna yerleştirildiği zamandır.

Bu alanın kuyruk yöneticisi tarafından oluşturulduğu saat için kullanılan biçim:

- HHMMSSSTH

karakterlerin temsil ettiği yer:

SS

saat (00-23)

MM

dakika (00-59)

SS

saniye (00-59; nota bakın)

T

saniyenin onda biri (0-9)

H

saniyenin yüzde biri (0-9)

Not: Sistem saati çok doğru bir zaman standardıyla eşitlenirse, nadir durumlarda *PutTime* içindeki saniyeler için 60 ya da 61 geri döndürülebilir. Bu, artık saniyeler genel zaman standardına eklendiğinde olur.

Greenwich Ortalama Saati (GMT), sistem saatinin GMT ' ye doğru şekilde ayarlanmasına bağlı olarak *PutDate* ve *PutTime* alanları için kullanılır.

Bu alanın uzunluğu MQ_PUT_TIME_LENGTH tarafından verilir. Bu alanın ilk değeri, C ' deki boş dizgi ve diğer programlama dillerindeki sekiz boş karakterdir.

MQDMHO-İleti işleme seçeneklerini sil

MQDMHO yapısı, uygulamaların ileti tanıtıcılarının nasıl silineceğini denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, **MQDLTMH** çağrısındaki bir giriş parametresidir.

Karakter kümesi ve kodlama

MQDMHO içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında olmalıdır (**MQENC_NATIVE**).

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 486. MQDMHO içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQDMHO_STRUC_ID	'DMHO'
<u>Sürüm</u> (yapı sürümü numarası)	MQDMHO_VERSION_1	1
<u>Seçenekler</u> (seçenekler)	MQDMHO_NONE	0

Notlar:

1. C programlama dilinde, makro değişkeni **MQDMHO_DEFAULT**, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQDMHO MyDMHO = {MQDMHO_DEFAULT};
```

Dil bildirimleri

MQDMHO için C bildirimi

```
typedef struct tagMQDMHO;  
struct tagMQDMHO {  
    MQCHAR4 StrucId;          /* Structure identifier */  
    MQLONG  Version;         /* Structure version number */  
};
```



```

MQLONG Options; /* Options that control the action of MQDLTMH */
};

```

MQDMHO için COBOL bildirimi

```

** 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.

```

MQDMHO için PL/I bildirimi

```

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 */

```

MQDMHO için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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)

```

MQDMHO için StrucId (MQCHAR4)

Bu, ileti silme tanıtıcısı seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQDMHO_STRUC_ID.

Değer şöyle olmalıdır:

MQDMHO_STRUC_ID

İleti silme işleme seçenekleri yapısına ilişkin tanıtıcı.

C programlama dili için, **MQDMHO_STRUC_ID_ARRAY** sabiti de tanımlanır. Bu, **MQDMHO_STRUC_ID** ile aynı değere sahiptir, ancak bir dizgi yerine bir karakter dizisidir.

MQDMHO için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQDMHO_VERSION_1

Version-1 ileti tanıtıcısı seçeneklerini silme yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQDMHO_CURRENT_VERSION

İleti silme tanıtıcısı seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **MQDMHO_VERSION_1**.

MQDMHO için Seçenekler (MQLONG)

Değer şöyle olmalıdır:

MQDMHO_NONE

Seçenek belirtilmedi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **MQDMHO_NONE**.

MQDMPO-İleti silme özelliği seçenekleri

MQDMPO yapısı, uygulamaların iletilerin özelliklerinin nasıl silineceğini denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQDLTMP çağrısına ilişkin bir giriş değiştirgesidir.

Karakter kümesi ve kodlama

MQDMPO içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında (MQENC_NATIVE) olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 487. MQDMPO içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQDMPO_STRUC_ID	'DMP0'
Sürüm (yapı sürümü numarası)	MQDMPO_VERSION_1	1
Seçenekler (MQDMPO işlemi denetleyen seçenekler)	MQDLTMP işlemi denetleyen seçenekler	MQDMPO_NONE

Notlar:

1. C programlama dilinde, makro değişkeniMQDMPO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQDMPO MyDMP0 = {MQDMPO_DEFAULT};
```

Dil bildirimleri

MQDMPO için C bildirimini

```
typedef struct tagMQDMPO MQDMPO;
struct tagMQDMPO {
    MQCHAR4 StrucId;          /* Structure identifier */
    MQLONG  Version;         /* Structure version number */
    MQLONG  Options;        /* Options that control the action of
                             MQDLTMP */
};
```

MQDMPO için COBOL bildirimini

```
** 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.
```

MQDMPO için PL/I bildirimi

```
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 */
```

MQDMPO için High Level Assembler bildirimi

```
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)
```

MQDMPO için StrucId (MQCHAR4)

Bu, ileti silme özelliği seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQDMPO_STRUC_ID.

Değer şöyle olmalıdır:

MQDMPO_STRUC_ID

İleti silme özelliği seçenekleri yapısına ilişkin tanıtıcı.

C programlama dili için, değişmez MQDMPO_STRUC_ID_ARRAY da tanımlanır. Bu değer MQDMPO_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQDMPO için Sürüm (MQLONG)

İleti özellik seçenekleri yapısını sil-Sürüm alanı

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MQDMPO_VERSION_1

İleti silme özelliği seçenekleri yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQDMPO_CURRENT_VERSION

İleti silme özelliği seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQDMPO_VERSION_1' dir.

MQDMPO için Seçenekler (MQLONG)

İleti özelliği seçenekleri yapısını sil-Seçenekler alanı

Konum seçenekleri: Aşağıdaki seçenekler, özellik imleciyle karşılaştırıldığında özelliğin görel konumuyla ilgilidir.

MQDMPO_DEL_FIRST

Belirtilen adyla eşleşen ilk özelliği siler.

MQDMPO_DEL_PROP_UNDER_CURSOR

Özellik imleci tarafından gösterilen özelliği siler; bu, MQIMPO_INQ_FIRST ya da MQIMPO_INQ_NEXT seçeneği kullanılarak en son sorulmuş olan özelliiktir.

İleti tanıtıcısı yeniden kullanıldığında özellik imleci ilk durumuna getirilir. İleti tanıtıcısı bir MQGET çağrısındaki MQGMO yapısının *MsgHandle* alanında ya da bir MQPUT çağrısındaki MQPMO yapısında belirtildiğinde de ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında bu seçenek kullanılırsa, çağrı tamamlama kodu MQCC_FAILED ve neden MQRC_PROPERTY_NOT_ALLOWED ile başarısız olur. Özellik imlecinin gösterdiği özellik zaten silindiye, çağrı MQCC_FAILED tamamlama kodu ve MQRC_PROPERTY_NOT_ALLOWED nedeniyle de başarısız olur.

Bu seçeneklerden hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

MQDMPO_NONE

Seçenek belirtilmedi.

Bu alan her zaman bir giriş alanıdır. Bu alanın ilk değeri MQDMPO_DEL_FIRST.

MQEPH-Gömülü PCF üstbilgisi

MQEPH yapısı, ileti programlanabilir bir komut biçimi (PCF) iletisi olduğunda iletide bulunan ek verileri tanımlar. *PCFHeader* alanı, bu yapıyı izleyen PCF değiştirgelerini tanımlar ve bu, PCF ileti verilerini diğer üstbilgilerle izlemenizi sağlar.

Biçim Adı

MQFMT_EMBEDDED_PCF

Karakter kümesi ve kodlama

MQEPH içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında yer almalıdır.

MQEPH 'nin karakter kümesini ve kodlamasını MQMD' deki *CodedCharSetId* ve *Encoding* alanlarına (MQEPH yapısı ileti verilerinin başındadır) ya da MQEPH yapısından önce gelen üstbilgi yapısına (diğer tüm durumlar) ayarlayın.

Kullanım

Komut sunucusuna ya da başka bir kuyruk yöneticisi PCF kabul eden sunucuya komut göndermek için MQEPH yapılarını kullanamazsınız.

Benzer şekilde, komut sunucusu ya da başka bir kuyruk yöneticisi PCF kabul eden sunucu, MQEPH yapıları içeren yanıtlar ya da olaylar oluşturmaz.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

<i>Çizelge 488. MQEPH için MQEPH 'deki Alanlar</i>		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQEPH_STRUC_ID	' EPH- '
<u>Sürüm</u> (yapı sürümü numarası)	MQEPH_VERSION_1	1
<u>StrucLength</u> (MQEPH yapısının uzunluğu ve bunu izleyen MQCFH ve değiştirge yapıları)	MQEPH_STRUC_LENGT H_FIXED	68
<u>Kodlama</u> (son PCF parametre yapısını izleyen verilerin sayısal kodlaması)	Yok	0
<u>CodedCharSetId</u> (son PCF parametre yapısını izleyen verilerin karakter kümesi tanıtıcısı)	MQCCSI_AYRILMAMIŞ	0
<u>Biçim</u> (son PCF parametre yapısını izleyen verilerin biçim adı)	MQFMT_NONE	Boşluklar

Çizelge 488. MÇEPH için MÇEPH 'deki Alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
İşaretler (işaretler)	MÇEPH_NONE	0
PCFHeader (programlanabilir komut biçimi (PCF) üstbilgisi)	Çizelge 489 sayfa 368 içinde tanımlandığı şekilde adlar ve değerler	0

Notlar:

- simgesi, tek bir boş karakteri temsil eder.
- C programlama dilinde, makro değişkeniMÇEPH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MÇEPH MyEPH = {MÇEPH_DEFAULT};
```

Dil bildirimleri

MÇEPH için C bildirim

```
typedef struct tagMÇEPH MÇEPH;
struct tagMQDH {
    MQCHAR4  StrucId;          /* Structure identifier */
    MQLONG   Version;        /* Structure version number */
    MQLONG   StrucLength;    /* Total length of MÇEPH 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 */
};
```

MÇEPH için COBOL bildirim

```
** MÇEPH structure
10 MÇEPH.
** Structure identifier
15 MÇEPH-STRUCID PIC X(4).
** Structure version number
15 MÇEPH-VERSION PIC S9(9) BINARY.
** Total length of MÇEPH structure including the MQCFH
** and parameter structures that follow it
15 MÇEPH-STRUCLNGTH PIC S9(9) BINARY.
** Numeric encoding of data that follows last
** PCF structure
15 MÇEPH-ENCODING PIC S9(9) BINARY.
** Character set identifier of data that
** follows last PCF parameter structure
15 MÇEPH-CODEDCHARSETID PIC S9(9) BINARY.
** Format name of data that follows last PCF
** parameter structure
15 MÇEPH-FORMAT PIC X(8).
** Flags
15 MÇEPH-FLAGS PIC S9(9) BINARY.
** Programmable command format header
15 MÇEPH-PCFHEADER.
** Structure type
20 MÇEPH-PCFHEADER-TYPE PIC S9(9) BINARY.
** Structure length
20 MÇEPH-PCFHEADER-STRUCLNGTH PIC S9(9) BINARY.
** Structure version number
20 MÇEPH-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.

```

MQEPH için PL/I bildirimi

```

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 */

```

MQEPH için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

MQEPH          DSECT
MQEPH_STRUCID  DS   CL4  Structure identifier
MQEPH_VERSION  DS   F    Structure version number
MQEPH_STRUCLNGTH  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   0F  Force fullword alignment
MQEPH_PCFHEADER_TYPE DS   F    Structure type
MQEPH_PCFHEADER_STRUCLNGTH 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)

```

MQEPH için Visual Basic bildirimi

```
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'
  Encoding     As Long      'Numeric encoding of data that follows last'
  CodedCharSetId As Long      'PCF parameter structure'
  Format       As Long      'Character set identifier of data that'
  Flags       As Long      'follows last PCF parameter structure'
  PCFHeader   As String*8  'Format name of data that follows last PCF'
  End Type    'parameter structure'
  Flags       As Long      'Flags'
  PCFHeader   As MQCFH     'Programmable command format header'
End Type

Global MQEPH_DEFAULT As MQEPH
```

MQEPH için StrucId (MQCHAR4)

Bu, yerleşik PCF üstbilgi yapısının yapı tanıtcısıdır. Her zaman bir giriş alanıdır. Değeri MQEPH_STRUC_ID.

Değer şöyle olmalıdır:

MQEPH_STRUC_ID

Yerleşik PCF üstbilgi yapısına ilişkin tanıtcı.

C programlama dili için, MQEPH_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQEPH_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQEPH için Sürüm (MQLONG)

Değer şöyle olmalıdır:

MQEPH_VERSION_1

Yerleşik PCF üstbilgi yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCFH_VERSION_3

Gömülü PCF üstbilgi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri MQEPH_VERSION_1' dir.

MQEPH için StrucLength (MQLONG)

Bu, sonraki üstbilgi yapısından önceki veri miktarıdır. Aşağıdakileri içerir:

- MQEPH üstbilgisinin uzunluğu
- Üstbilgiyi izleyen tüm PCF değiştirgelerinin uzunluğu
- Bu parametreleri izleyen herhangi bir boş dolgu

StrucLength , 4 'ün katı olmalıdır.

Yapının değişmez uzunluklu kısmı MQEPH_STRUC_LENGTH_FIXED tarafından tanımlandı.

Bu alanın ilk değeri 68 'dir.

MQEPH için kodlama (MQLONG)

Bu, MQEPH yapısını ve ilişkili PCF değiştirgelerini izleyen verilerin sayısal kodlamasıdır; MQEPH yapısındaki karakter verileri için geçerli değildir.

Bu alanın ilk değeri 0 'dir.

MQEPH için CodedCharSetId (MQLONG)

Bu, MQEPH yapısını ve ilişkili PCF deęiřtirgelerini izleyen verilerin karakter kümesi tanıtıcısıdır; MQEPH yapısının kendisindeki karakter verileri için geçerli deęildir.

Bu alanın ilk deęeri MQCCSI_UNDEFINED 'dir.

MQEPH için biçim (MQCHAR8)

MQEPH yapısını ve ilişkili PCF deęiřtirgelerini izleyen verilerin biçim adıdır.

Bu alanın ilk deęeri MQFMT_NONE.

MQEPH için işaretler (MQLONG)

Ařaęıdaki deęerler kullanılabilir:

MQEPH_NONE

Hiçbir işaret belirtilmedi. MQEPH_NONE, program belgelerine yardımcı olmak için tanımlanır. Bu deęişmezin başka bir deęişmeze birlikte kullanılması amaçlanmamıştır, ancak deęeri sıfır olduęu için bu tür bir kullanım algılanamaz.

MQEPH_CCsid_EMBEDDED

Karakter verileri içeren parametrelerin karakter kümesi, her bir yapıdaki CodedCharSetId alanında tek tek belirtilir. StrucId ve Format alanlarının karakter kümesi, MQEPH yapısından önce gelen üstbilgi yapısındaki CodedCharSetId alanıyla ya da MQEPH iletinin başlangıcında ise MQMD 'deki CodedCharSetId alanıyla tanımlanır.

Bu alanın ilk deęeri MQEPH_NONE.

MQEPH için PCFHeader (MQCFH)

Bu, MQEPH yapısını izleyen PCF deęiřtirgelerini tanımlayan programlanabilir komut biçimi (PCF) üstbilgisidir. Bu, PCF ileti verilerini dięer üstbilgilerle izlemenizi saęlar.

PCF üstbilgisi başlangıçta ařaęıdaki deęerlerle tanımlanır:

<i>Çizelge 489. MQCFH 'deki Alanlar</i>		
Alan adı	Deęişmezin adı	Deęişmezin deęeri
<i>Type</i>	MQCFT_NONE	0
<i>StrucLength</i>	MQCFH_STRUC_LENGTH	36
<i>Version</i>	MQCFH_VERSION_3	3
<i>StrucLength</i>	Yok	0
<i>Command</i>	MQCMD_NONE	0
<i>MsgSeqNumber</i>	Yok	1
<i>Control</i>	MQCFC_LAST	1
<i>CompCode</i>	MQCC_OK	0
<i>Reason</i>	MQRC_NONE	0
<i>ParameterCount</i>	Yok	0

Uygulamanın Type 'i MQCFT_NONE' dan, yerleşik PCF üstbilgisini kullanması için geçerli bir yapı tipine çevirmesi gerekir.

MQGMO-Get-message seçenekleri

MQGMO yapısı, uygulamanın iletilerin kuyruklardan nasıl kaldırılacağını denetlemesini saęlar. Yapı, MQGET çağırısına ilişkin bir giriş/çıkış deęiřtirgesidir.

Sürüm

MQGMO ' nun yürürlükteki sürümü: MQGMO_VERSION_4. Belirli alanlar yalnızca MQGMO ' nun belirli sürümlerinde kullanılabilir. Birden çok ortam arasında uygulamaları bağlamanız gerekiyorsa, MQGMO sürümünün tüm ortamlarda tutarlı olduğundan emin olmanız gerekir. Yalnızca yapının belirli sürümlerinde var olan alanlar, “MQGMO-Get-message seçenekleri” sayfa 368 içinde ve alan açıklamalarında olduğu gibi tanımlanır.

Desteklenen programlama dilleri için sağlanan üstbilgi, COPY ve INCLUDE dosyaları, ortam tarafından desteklenen ancak *Version* alanının ilk değeri MQGMO_VERSION_1olarak ayarlanmış MQGMO ' nun en son sürümünü içerir. version-1 yapısında var olmayan alanları kullanmak için, *Version* alanını gerekli sürümün sürüm numarasına ayarlayın.

Karakter kümesi ve kodlama

MQGMO ' daki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında yer almalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 490. MQGMO için MQGMO içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQGMO_STRUC_ID	'GMO↵'
<u>Sürüm</u> (yapı sürümü numarası)	MQGMO_VERSION_1	1
<u>MQGMO-Options alanı</u> (MQGET işlemi denetleyen seçenekler)	MQGMO_NO_WAIT	0
<u>WaitInterval</u> (bekleme aralığı)	Yok	0
<u>Signal1</u> (sinyal)	Yok	z/OS üzerinde boş değerli gösterge; 0 tersi durumda
<u>Signal2</u> (sinyal tanıtıcısı)	Yok	0
<u>ResolvedQName</u> (hedef kuyruğun çözülmüş adı)	Yok	Boş değerli dizilim ya da boşluklar
Not: <i>Version</i> , MQGMO_VERSION_2' den küçükse kalan alanlar yoksayılr.		
<u>MatchOptions</u> (MQGET için kullanılan seçim ölçütlerini denetleyen seçenekler)	MQMO_MATCH_MSG_ID + MQMO_MATCH_CORREL_ID	3
<u>GroupStatus</u> (alınan iletinin bir grupta olup olmadığını gösteren işaret)	MQGS_NOT_IN_GROUP	'↵'
<u>SegmentStatus</u> (alınan iletinin bir mantıksal iletinin parçası olup olmadığını gösteren işaret)	MQSS_NOT_A_SEGMENT	'↵'
<u>Bölümleme</u> (alınan ileti için daha fazla bölümlemeye izin verilip verilmediğini gösteren işaret)	MQSEG_ENGELLEME	'↵'

Çizelge 490. MQGMO için MQGMO içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
Reserved1 (ayrılmış)	Yok	'¬'
Not: <i>Version</i> , MQGMO_VERSION_3' den küçükse kalan alanlar yoksayılr.		
MsgToken (ileti simgesi)	MQMTOK_NONE	Boş Değerler
ReturnedLength (döndürülen ileti verilerinin bayt cinsinden uzunluğu)	MQRL_UNDEFINED	-1
Not: <i>Version</i> , MQGMO_VERSION_4' den küçükse kalan alanlar yoksayılr.		
Reserved2 (ayrılmış)	Yok	'¬'
MsgHandle (kuyruktan alınmakta olan iletinin özellikleriyle doldurulacak bir ileti tanıtıcısı)	MQHM_NONE	0
<p>Notlar:</p> <ol style="list-style-type: none"> ¬ simgesi, tek bir boş karakteri temsil eder. Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir. C programlama dilinde, makro değişkeniMQGMO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir: <pre style="background-color: #f0f0f0; padding: 10px;">MQGMO MyGMO = {MQGMO_DEFAULT};</pre>		

Dil bildirimleri

MQGMO için C bildirimi

```
typedef struct tagMQGMO MQGMO;
struct tagMQGMO {
    MQCHAR4    StructId;        /* 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 */
};
```

Not: z/OSsistemlerinde *Signal1* alanı PMQLONGolarak bildirilir.

MQGMO için COBOL bildirimini

```
** 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.
```

Not: z/OSsistemlerinde *Signal1* alanı POINTERolarak bildirilir.

MQGMO için PL/I bildirimini

```
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 */
```

Not: z/OSsistemlerinde *Signal1* alanı pointerolarak bildirilir.

MQGMO için High Level Assembler bildirimi

```
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)
```

MQGMO için High Level Assembler bildirimi

```
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'
              'used for MQGET'
GroupStatus   As String*1  'Flag indicating whether message'
              'retrieved is in a group'
SegmentStatus As String*1  'Flag indicating whether message'
              'retrieved is a segment of a logical'
              'message'
Segmentation  As String*1  'Flag indicating whether further'
              'segmentation is allowed for the message'
              'retrieved'
Reserved1     As String*1  'Reserved'
MsgToken      As MQBYTE16  'Message token'
ReturnedLength As Long     'Length of message data returned (bytes)'
End Type
```

PROPCTL kanal seçenekleri MQGMO

Daha önceki bir IBM MQ sürümünden bir ortak kuyruk yöneticisine IBM MQ 9.3 kuyruk yöneticisinden gönderilen bir iletiye hangi ileti özelliklerinin dahil edildiğini denetlemek için **PROPCTL** kanal özneliğini kullanın.

Çizelge 491. Kanal iletişi özellik özneliği ayarları	
PROPCTL	Açıklama
TÜMÜ	<p>Daha önceki bir sürümden iş ortağı kuyruk yöneticisine bağı uygulamalar, bir IBM MQ 9.3 uygulaması tarafından iletiye yerleştirilen özellikleri işleyebiliyorsa bu seçeneği kullanın.</p> <p>Tüm özellikler, MQRFH2içine yerleştirilen ad-değer çiftlerine ek olarak ortak kuyruk yöneticisine gönderilir.</p> <p>İki uygulama tasarımı sorununu göz önünde bulundurmanız gerekir:</p> <ol style="list-style-type: none"> 1. Ortak kuyruk yöneticisine bağı bir uygulama, IBM MQ 9.3 kuyruk yöneticisinde oluşturulan MQRFH2 üstbilgilerini içeren iletileri işleyebilmelidir. 2. Ortak kuyruk yöneticisine bağı uygulamanın, MQPD_SUPPORT_REQUIRED ile doğru şekilde işaretlenmiş yeni ileti özelliklerini işlemesi gerekir. <p>ALL kanal seçeneği ayarlandığında, JMS uygulamaları kanalı kullanarak IBM MQ 9.3 ile daha önceki bir sürüm arasında birlikte çalışabilir. İleti özelliklerini kullanan yeni IBM MQ 9.3 uygulamaları, önceki sürüm uygulamasının MQRFH2 üstbilgilerini nasıl işlediğine bağı olarak daha önceki bir sürümdeki uygulamalarla birlikte çalışabilir.</p>
DERLEME	<p>Bazı durumlarda daha önceki bir sürüm iş ortağı kuyruk yöneticisine bağı uygulamalara ileti özellikleri göndermek için bu seçeneği kullanın, ancak tümünü göndermeyin. İleti özellikleri yalnızca iki koşul karşılandığında gönderilir:</p> <ol style="list-style-type: none"> 1. Hiçbir özellik, ileti özelliği işlenmesi gerektiren olarak işaretlenmemelidir. 2. İleti özelliklerinden en az biri "ayrılmış" bir klasörde olmalıdır; bkz. <u>Not</u>. <p>COMPAT kanal seçeneği ayarlandığında, JMS uygulamaları kanalı kullanarak IBM MQ 9.3 ile önceki bir sürüm arasında birlikte çalışabilir.</p> <p>Kanal, ileti özelliklerini kullanan her uygulama tarafından kullanılamaz, yalnızca ayrılmış klasörleri kullanan uygulamalar tarafından kullanılabilir. İletinin ya da özelliğin gönderilip gönderilmediğine ilişkin kurallar şunlardır:</p> <ol style="list-style-type: none"> 1. İletinin özellikleri varsa, ancak özelliklerin hiçbiri "ayrılmış" bir klasörle ilişkilendirilmediyse, ileti özellikleri gönderilmez. 2. "Ayrılmış" bir özellik klasöründe herhangi bir ileti özelliği yaratıldıysa, iletiyle ilişkili tüm ileti özellikleri gönderilir. Ancak: <ol style="list-style-type: none"> a. İleti özelliklerinden herhangi biri gerekli destek olarak işaretliyse (MQPD_SUPPORT_REQUIRED ya da MQPD_SUPPORT_REQUIRED_IF_LOCAL), iletinin tamamı reddedilir. Rapor seçeneklerinin değerine göre geri döndürülür, atılır ya da teslim edilmeyen iletiler kuyruğuna gönderilir. b. Herhangi bir ileti özelliği gerekli destek olarak işaretlenmezse, tek bir özellik gönderilmeyebilir. İleti özelliği tanımlayıcı alanlarından herhangi biri varsayılan olmayan değerlere ayarlanırsa, tek tek özellik gönderilmez. İleti gönderilmeye devam ediyor. Varsayılan olmayan bir özellik tanımlayıcı alan değeri örneği: MQPD_USER_CONTEXT. <p>Not: "Ayrılmış" klasör adları mcd . , jms . , usx . ya da mqext . ile başlar. Bu klasörler, JMS arabirimini kullanan uygulamalar için oluşturulur. IBM MQ 9.3 ' de bu klasörlere yerleştirilen ad-değer çiftleri ileti özellikleri olarak kabul edilir.</p> <p>İleti özellikleri, MQRFH2 üstbilgisine yerleştirilen ad-değer çiftlerine ek olarak bir MQRFH2 üstbilgisinde gönderilir. MQRFH2 üstbilgisine yerleştirilen ad-değer çiftleri, ileti reddedilmediği sürece gönderilir.</p>

Çizelge 491. Kanal iletisi özellik özneliği ayarları (devamı var)	
PROPCTL	Açıklama
YOK	<p>Daha önceki bir sürüm iş ortağı kuyruk yöneticisine bağlı uygulamalara ileti özelliklerinin gönderilmesini önlemek için bu seçeneği kullanın. Ad-değer çiftlerini ve ileti özelliklerini içeren bir MQRFH2 yine de gönderilir, ancak yalnızca ad-değer çiftleriyle birlikte gönderilir.</p> <p>NONE kanal seçeneği ayarlandığında, bir JMS iletisi, herhangi bir JMS ileti özelliği olmayan bir JMSTextMessage ya da JMSBytesMessage olarak gönderilir. Önceki bir sürüm uygulamasının IBM MQ 9.3 uygulamasında ayarlanan tüm özellikleri yoksaması mümkünse, bu uygulamayla birlikte çalışabilir.</p>

PROPCTL MQGMO için kuyruk seçenekleri

İleti özelliklerinin, herhangi bir **MQGMO** ileti özelliği seçeneği ayarlanmadan **MQGET** çağırılan bir uygulamaya nasıl döndürüleceğini denetlemek için **PROPCTL** kuyruk özneliğini kullanın.

Çizelge 492. Kuyruk iletisi özellik özneliği ayarları	
PROPCTL	Açıklama
TÜMÜ	<p>Aynı kuyruktan ileti okuyan farklı uygulamaların iletiyi farklı şekillerde işleyebilmesi için ALL seçeneğini kullanın.</p> <ul style="list-style-type: none"> Daha önceki bir sürümden değiştirilmeden geçirilen bir uygulama, MQRFH2 ögesini doğrudan okumaya devam edebilir. MQRFH2 üstbilgisinde özelliklere doğrudan erişilebilir. <p>Yeni özellikleri ve yeni özellik özneliklerini işlemek için uygulamayı değiştirmeniz gerekir. Uygulama, MQRFH2 üstbilgilerinin yerleşim düzenindeki ve sayısındaki değişikliklerden etkilenebilir. Bazı klasör öznelikleri kaldırılabilir ya da IBM MQ , önceki bir sürümde yoksaydığı MQRFH2 üstbilgisinin yerleşim düzeninde bir hata bildiriyor olabilir.</p> <ul style="list-style-type: none"> Yeni ya da değiştirilen bir uygulama, ileti özelliklerini sorgulamak ve MQRFH2 üstbilgisinde ad-değer çiftlerini doğrudan okumak için MQI ileti özelliğini kullanabilir. <p>İletideki tüm özellikler uygulamaya döndürülür.</p> <ul style="list-style-type: none"> Uygulama bir ileti tanıtıcısı oluşturmak için MQCRTMH ' i çağırırsa, ileti özelliklerini MQINQMPkullanarak sorgulamalıdır. İleti özellikleri olmayan ad-değer çiftleri, ileti özelliklerinin soyulduğu MQRFH2içinde kalır. Uygulama bir ileti tanıtıcısı oluşturmazsa, tüm ileti özellikleri ve ad-değer çiftleri MQRFH2içinde kalır. <p>ALL bu etkiyi yalnızca, alan uygulama bir MQGMO_PROPERTIES seçeneği ayarlı değilse ya da MQGMO_PROPERTIES_AS_Q_DEFolarak ayarlıysa gösterir.</p>

Çizelge 492. Kuyruk iletisi özellik özneliği ayarları (devamı var)

PROPCTL	Açıklama
COMPAT (varsayılan)	<p>COMPAT varsayılan seçenektir. GMO_PROPERTIES_* ayarlanmazsa, önceki bir sürümdeki değiştirilmemiş bir uygulamada olduğu gibi, COMPAT varsayılr. Belirtik olarak MQRFH2 yaratmayan daha önceki bir sürüm uygulaması olan COMPAT seçeneği varsayılan olarak IBM MQ 9.3 üzerinde değişiklik yapılmadan çalışır.</p> <p>JMS iletilerini okumak için daha önceki bir sürüm uygulaması MQI uygulaması yazdıysanız bu seçeneği kullanın.</p> <ul style="list-style-type: none">MQRFH2 üstbilgisinde saklanan JMS özellikleri, mcd., jms., us1. ya da mqextile başlayan adlara sahip klasörlerdeki MQRFH2 üstbilgisinde uygulamaya döndürülür.İletide JMS klasörleri varsa ve bir IBM MQ 9.3 uygulaması iletiye yeni özellik klasörleri eklerse, bu özellikler MQRFH2 içinde de döndürülür. Sonuç olarak, yeni özellikleri ve yeni özellik özneliklerini işlemek için uygulamayı değiştirmeniz gerekir. Değiştirilmemiş bir uygulama, MQRFH2 üstbilgilerinin yerleşim düzenindeki ve sayısındaki değişikliklerden etkilenebilir. Bazı klasör öznelikleri kaldırılmış olabilir ya da IBM MQ, önceki bir sürümde yoksaydığı MQRFH2 üstbilgisinin yerleşim düzeninde hatalar bulabilir. <p>Not: Bu senaryoda, uygulamanın önceki bir sürümüne ya da IBM MQ 9.3 kuyruk yöneticisine bağlı olması fark etmez, uygulamanın davranışı aynıdır. Kanal PROPCTL özneliği COMPAT ya da ALL olarak ayarlanırsa, iletide önceki sürüm ortak kuyruk yöneticisine yeni ileti özellikleri gönderilir.</p> <ul style="list-style-type: none">İleti bir JMS iletisi değilse, ancak diğer özellikleri içeriyorsa, bu özellikler bir MQRFH2 üstbilgisinde uygulamaya döndürülmez.¹Bu seçenek, birçok durumda, belirtik olarak bir MQRFH2 yaratan önceki sürüm uygulamalarının doğru bir şekilde çalışmasını da sağlar. Örneğin, MQRFH2 İçeren JMS ileti özelliklerini yaratan bir MQI programı doğru çalışmaya devam eder. JMS ileti özellikleri olmadan, ancak diğer bazı MQRFH2 klasörleriyle bir ileti oluşturulursa, klasörler uygulamaya döndürülür. Yalnızca klasörler ileti özelliği klasörleriye, bu belirli klasörler MQRFH2' den kaldırılır. İleti özelliği klasörleri content= 'properties' adlı yeni klasör özneliğine sahip olarak tanıtılır ya da Tanımlı özellik klasörü adı ya da Gruplanmamış özellik klasörü adlistesinde yer alan klasörlerdir.Uygulama bir ileti tanıtıcısı oluşturmak için MQCRTMH ' i çağırırsa, ileti özelliklerini MQINQMP kullanarak sorgulamalıdır. İleti özellikleri MQRFH2 üstbilgilerinden kaldırılır. İleti özellikleri olmayan ad-değer çiftleri MQRFH2 içinde kalır.Uygulama bir ileti tanıtıcısı oluşturmak için MQCRTMH ögesini çağırırsa, iletinin JMS klasörleri olup olmadığına bakılmaksızın tüm ileti özelliklerini sorgulayabilir.Uygulama bir ileti tanıtıcısı oluşturmazsa, tüm ileti özellikleri ve ad-değer çiftleri MQRFH2 içinde kalır. <p>Bir ileti yeni kullanıcı özelliği klasörleri içeriyorsa, iletinin yeni ya da değiştirilmiş bir IBM MQ 9.3 uygulaması tarafından oluşturulduğunu çıkarabilirsiniz. Alan uygulama bu yeni özellikleri doğrudan bir MQRFH2 içinde işlemek için kullanıyorsa, uygulamayı ALL seçeneğini kullanacak şekilde değiştirmeniz gerekir. Varsayılan COMPAT seçeneği ayarlandığında, değiştirilmemiş bir uygulama, IBM MQ 9.3 özellikleri olmadan MQRFH2' in geri kalanını işlemeye devam eder.</p> <p>PROPCTL arabiriminin amacı, MQRFH2 klasörlerini okuyan eski uygulamaları ve ileti özelliği arabirimini kullanan yeni ve değiştirilmiş uygulamaları desteklemektir. Yeni uygulamaların tüm kullanıcı iletisi özellikleri için ileti özelliği arabirimini kullanmasını ve MQRFH2 üstbilgilerini doğrudan okumaktan ve yazmaktan kaçınmasını hedefleyin.</p> <p>COMPAT yalnızca, alan uygulama bir MQGMO_PROPERTIES seçeneği ayarlı değilse ya da MQGMO_PROPERTIES_AS_Q_DEF olarak ayarladıysa bu etkiyi gösterir.</p>

Çizelge 492. Kuyruk iletisi özellik özneliği ayarları (devamı var)

PROPCTL	Açıklama
ZORLA	<p>FORCE seçeneği, tüm ileti özelliklerini MQRFH2 üstbilgilerine yerleştirir. MQRFH2 üstbilgilerindeki tüm ileti özellikleri ve ad-değer çiftleri iletide kalır. İleti özellikleri MQRFH2' den kaldırılmaz ve bir ileti tanıtıcısı aracılığıyla kullanılabilir kıılır. FORCE seçeneğinin belirlenmesinin etkisi, yeni geçirilen bir uygulamanın MQRFH2 üstbilgilerinden ileti özelliklerini okumasını sağlamaktır.</p> <p>Bir uygulamayı IBM MQ 9.3 ileti özelliklerini işleyecek şekilde değiştirdiğinizi, ancak daha önce olduğu gibi MQRFH2 üstbilgileriyle doğrudan çalışma yeteneğini koruduğunu varsayın. Başlangıçta PROPCTL kuyruk özneliğini FORCEolarak ayarlayarak, uygulamanın ileti özelliklerini kullanmaya ne zaman geçeceğine karar verebilirsiniz. İleti özelliklerini kullanmaya başlamaya hazır olduğunuzda PROPCTL kuyruk özneliğini başka bir değere ayarlayın. Uygulamadaki yeni işlev beklediğiniz gibi davranmazsa, PROPCTL seçeneğini FORCEolarak ayarlayın.</p> <p>FORCE yalnızca, alan uygulama bir MQGMO_PROPERTIES seçeneği belirlemediyse ya da bunu MQGMO_PROPERTIES_AS_Q_DEFolarak ayarladıysa bu etkiyi gösterir.</p>
YOK	<p>Var olan bir uygulamanın bir iletiyi işleyebilmesi, tüm ileti özelliklerini yoksayması ve yeni ya da değiştirilen bir uygulamanın ileti özelliklerini sorgulaması için NONE seçeneğini kullanın.</p> <ul style="list-style-type: none">• Uygulama bir ileti tanıtıcısı oluşturmak için MQCRTMH ' i çağırırsa, ileti özelliklerini MQINQMPkullanarak sorgulamalıdır. İleti özellikleri olmayan ad-değer çiftleri, ileti özelliklerinin soyulduğu MQRFH2içinde kalır.• Uygulama bir ileti tanıtıcısı yaratmazsa, tüm ileti özellikleri MQRFH2' den kaldırılır. MQRFH2 üstbilgilerindeki ad-değer çiftleri iletide kalır. <p>NONE değeri yalnızca, alan uygulama bir MQGMO_PROPERTIES seçeneği ayarlı değilse ya da MQGMO_PROPERTIES_AS_Q_DEFdeğerine ayarlıysa bu etkiyi gösterir.</p>
V6COMPAT	<p>Gönderildiği biçimde bir MQRFH2 almak için bu seçeneği kullanın. Gönderen uygulama ya da kuyruk yöneticisi ek ileti özellikleri yaratırsa, bunlar ileti tanıtıcısında döndürülür.</p> <p>Bu seçenek, hem gönderen hem de alan kuyruklarda ve araya giren iletim kuyruklarında belirlenmeli. Kuyruk adı çözüm yolundaki kuyruk tanımlamalarında ayarlanan diğer PROPCTL seçeneklerini geçersiz kılar.</p> <p>V6COMPAT seçeneğini yalnızca istisnai durumlarda kullanın. Örneğin, uygulamaları önceki bir sürümden IBM MQ 9.3sürümüne geçiriyorsanız, önceki sürümün davranışını koruduğu için bu seçenek değerlidir. Bu seçeneğin ileti verimi üzerinde bir etkisi olabilir. Ayrıca, yönetmek daha zordur; seçeneğin gönderen, alıcı ve müdahale eden iletim kuyruklarında ayarlandığından emin olmanız gerekir.</p> <p>V6COMPAT yalnızca, alan uygulama bir MQGMO_PROPERTIES seçeneği ayarlı değilse ya da MQGMO_PROPERTIES_AS_Q_DEFolarak ayarladıysa bu etkiyi yaratır.</p>

İleti özellikleri ve ad-değer çiftleri hakkında daha fazla bilgi için bkz. "[NameValue MQRFH2 için veri \(MQCHARn\)](#)" sayfa 528.

MQGMO için ileti özelliği seçenekleri

İleti özelliklerinin bir uygulamaya nasıl döndürüleceğini denetlemek için **MQGMO** ileti özelliği seçeneklerini kullanın.

¹ IBM MQ classes for JMS tarafından oluşturulan belirli özellik klasörlerinin varlığı bir JMS iletisini gösterir. Özellik klasörleri: mcd., jms., usr. ya da mqext.

Çizelge 493. MQGMO ileti özelliği seçeneği ayarları

MQGMO Seçenek	Açıklama
MQGMO_PROPERTIES_AS_Q_DEF	<p>Aynı kuyruktan okuyan ve GMO_PROPERTIES_* ayarını tanımlamayan IBM MQ uygulamaları, ileti özelliklerini farklı şekilde alır. İleti tanıtıcısı yaratmayan IBM MQ uygulamaları, kuyruk PROPCTL özniteliği tarafından denetlenir. IBM MQ uygulaması, MQRFH2 içindeki ileti özelliklerini almayı seçebilir ya da bir ileti tanıtıcısı yaratıp ileti özelliklerini sorgulayabilir. Uygulama bir ileti tanıtıcısı oluşturursa, özellikler MQRFH2' den kaldırılır.</p> <ul style="list-style-type: none">• GMO_PROPERTIES_* ayarını tanımlamayan ya da MQGMO_PROPERTIES_AS_Q_DEF olarak ayarlayan yeni ya da değiştirilmiş bir IBM MQ uygulaması, ileti özelliklerini sorgulamayı seçebilir. MQINQMP MQI çağrısını kullanarak ileti tanıtıcısı ve sorgu iletisi özellikleri yaratmak için MQCRTMH ayarlanmalıdır.• Yeni ya da değiştirilen bir uygulama ileti tanıtıcısı yaratmazsa, doğrudan MQRFH2 üstbilgilerinden aldığı ileti özelliklerini okuması gerekir.• PROPCTL kuyruk özniteliği FORCE olarak ayarlanırsa, ileti tanıtıcısında hiçbir özellik döndürülmez. Tüm özellikler MQRFH2 üstbilgilerinde döndürülür.• PROPCTL kuyruk özniteliği NONE ya da COMPAT olarak ayarlanırsa, ileti tanıtıcısı yaratan bir IBM MQ uygulaması tüm ileti özelliklerini alır.
MQGMO_PROPERTIES_IN_HANDLE	<p>Bir uygulamayı ileti özelliklerini kullanmaya zorlayın. Değiştirilen bir uygulamanın ileti tanıtıcısı yaratmakta başarısız olup olmadığını saptamak için bu seçeneği kullanın. Uygulama, ileti özelliklerini MQINQMPyerine doğrudan bir MQRFH2 içinden okumaya çalışıyor olabilir.</p>
MQGMO_NO_PROPERTIES	<ul style="list-style-type: none">• Tüm özellikler kaldırılır. JMS özellikleri gibi kuyruk yöneticisi tarafından oluşturulan özellikler kaldırılır.• Bir ileti tanıtıcısı yaratıldığında bile özellikler kaldırılır. Diğer MQRFH2 klasörlerindeki ad-değer çiftleri, ileti verilerinde bulunur.
MQGMO_PROPERTIES_FORCE_MQRFH2	<p>Bir ileti tanıtıcısı oluşturulsa da, özellikler MQRFH2 üstbilgilerinde döndürülür.</p> <ul style="list-style-type: none">• MQINQMP , bir ileti tanıtıcısı yaratıldığında bile ileti özellikleri döndürmez. Bir özellik sorulursa MQRC_PROPERTY_NOT_AVAILABLE döndürülür.

Çizelge 493. MQGMO ileti özelliği seçeneği ayarları (devamı var)

MQGMO Seçenek	Açıklama
MQGMO_PROPERTIES_COMPATIBILITY	<p>İleti bir JMS istemcisiyse, MQRFH2 üstbilgilerinde JMS özellikleri döndürülür. İleti tanıtıcısı yaratan yeni ya da değiştirilmiş IBM MQ uygulamaları farklı davranır.</p> <ul style="list-style-type: none">• İleti bir mcd . , jms . , usx . ya da mqext klasörü içeriyorsa, herhangi bir ileti özelliği klasöründeki tüm özellikler döndürülür.• İleti bir mcd . , jms . , usx . ya da mqext klasörü değil, özellik klasörleri içeriyorsa, bir MQRFH2 içinde ileti özellikleri döndürülmez.• Yeni ya da değiştirilmiş bir IBM MQ uygulaması bir ileti tanıtıcısı yaratırsa, ileti özelliklerini MQINQMP MQI çağrısını kullanarak sorgulayın. Tüm ileti özellikleri MQRFH2'iden kaldırılır.• Yeni ya da değiştirilmiş bir IBM MQ uygulaması bir ileti tanıtıcısı yaratırsa, iletideki tüm özellikler sorgulanabilir. İleti bir mcd . , jms . , usx . ya da mqext klasörü içermese bile, tüm ileti özellikleri sorgulanabilir.

İlgili başvurular

PROPCTL

2471 (09A7) (RC2471): MQRC_PROPERTY_NOT_ALLOWED

MQGMO için StrucId (MQCHAR4)

Bu, ileti alma seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQGMO_STRUC_ID.

Değer şöyle olmalıdır:

MQGMO_STRUC_ID

İleti alma seçenekleri yapısına ilişkin tanıtıcı.

C programlama dili için, MQGMO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQGMO_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQGMO için Sürüm (MQLONG)

Sürüm, yapı sürümü numarasıdır.

Değer aşağıdakilerden biri olmalıdır:

MQGMO_VERSION_1

Version-1 get-message seçenekleri yapısı.

Bu sürüm tüm ortamlarda desteklenir.

MQGMO_VERSION_2

Version-2 get-message seçenekleri yapısı.

Bu sürüm tüm ortamlarda desteklenir.

MQGMO_VERSION_3

Version-3 get-message seçenekleri yapısı.

Bu sürüm tüm ortamlarda desteklenir.

MQGMO_VERSION_4

Version-4 ileti alma seçenekleri yapısı.

Bu sürüm tüm ortamlarda desteklenir.

Yalnızca yapının daha yeni sürümlerinde bulunan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQGMO_CURRENT_VERSION

get-message seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQGMO_VERSION_1' dir.

MQGMO için Seçenekler (MQLONG)

MQGMO seçenekleri, MQGETeylemine denetler. Seçeneklerden sıfır ya da daha fazlasını belirtebilirsiniz. Birden fazla isteğe bağlı değere gereksinim duyarsanız:

- Değerleri ekleyin (aynı değişmezi bir kereden fazla eklemeyin) ya da
- Bitsel OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliorsa).

Geçerli olmayan seçenek birleşimleri belirtilir; diğer tüm birleşimler geçerlidir.

Bekleme seçenekleri

Aşağıdaki seçenekler, kuyruktaki iletilerin gelmesini bekleme işlemiyle ilgilidir:

MQGMO_WAIT

Uygulama, uygun bir ileti gelinceye kadar bekler. Uygulamanın bekleme süresi üst sınırı *WaitInterval* içinde belirtilir.

Önemli: Uygun bir ileti hemen varsa, bekleme ya da gecikme yoktur.

MQGET istekleri engellenirse ya da MQGET istekleri bekleme sırasında engellenirse, bekleme iptal edilir. Arama, kuyrukta uygun ileti olup olmadığına bakılmaksızın MQCC_FAILED ve neden kodu MQRC_GET_INHIBITED ile tamamlanır.

MQGMO_WAIT komutunu MQGMO_BROWSE_FIRST ya da MQGMO_BROWSE_NEXT seçenekleriyle kullanabilirsiniz.

Birden çok uygulama aynı paylaşılan kuyrukta bekliyorsa, uygun bir ileti geldiğinde hangi uygulamanın etkinleştirildiğini aşağıdaki kurallar belirler:

<i>Çizelge 494. Paylaşılan bir kuyrukta MQGET çağrılarının etkinleştirilmesine ilişkin kurallar.</i>		
Etkinleştirilmeyi bekleyen MQGET çağrılarının sayısı		Sonuç
BROWSE (Göz At) seçeneğiyle	BROWSE seçeneği olmadan²	
Yok	bir ya da daha fazla	BROWSE (Göz At) seçeneği olmayan bir MQGET çağrısı etkinleştirilir.
bir ya da daha fazla	Yok	BROWSE (Göz At) seçeneğiyle yapılan tüm MQGET çağrıları etkinleştirilir.
bir ya da daha fazla	bir ya da daha fazla	BROWSE (Göz At) seçeneği olmayan bir MQGET çağrısı etkinleştirilir. Etkinleştirilen bir BROWSE seçeneğiyle yapılan MQGET çağrılarının sayısı öngörülemez.

BROWSE seçeneği olmadan birden çok MQGET çağrısı aynı kuyrukta bekliyorsa, yalnızca bir çağrı etkinleştirilir. Kuyruk yöneticisi, çağrıları bekleme sırasına aşağıdaki sırayla öncelik vermeyi dener:

1. Yalnızca belirli iletiler (örneğin, belirli bir MsgId ya da CorrelId (ya da her ikisi) ile karşılanabilen belirli get-wait istekleri.
2. Herhangi bir ileti tarafından karşılanabilen genel bekleme alma istekleri.

² MQGMO_LOCK seçeneğini belirten bir MQGET çağrısı, göz atmayan bir çağrı olarak kabul edilir.

Not:

- İlk kategori içinde, daha belirli bekleme alma isteklerine ek öncelik verilmez. Örneğin, hem MsgId hem de CorrelId belirten istekler.
- Her iki kategori içinde hangi uygulamanın seçildiği tahmin edilemez. Özellikle, en uzun süre bekleyen uygulama mutlaka seçilen uygulama değildir.
- İşletim sisteminin yol uzunluğu ve öncelik zamanlaması ile ilgili önemli noktalar, beklenenden daha düşük işletim sistemi önceliğine sahip bir bekleme uygulamasının iletiyi aldığı anlamına gelebilir.
- Beklemeyen bir uygulama, iletiyi olduğu gibi almayı tercih eden bir uygulama da olabilir.

z/OS z/OS üzerinde aşağıdaki noktalar geçerlidir:

- İletin gelmesini beklerken uygulamanın başka bir işle devam etmesini istiyorsanız, bunun yerine sinyal seçeneğini (MQGMO_SET_SIGNAL) kullanmayı düşünün. Ancak, sinyal seçeneği ortama özgüdür; farklı ortamlar arasında bağlantı oluşturacağınız uygulamalar bunu kullanmamalıdır.
- Aynı iletiyi bekleyen birden çok MQGET çağrısı varsa, bekleme ve sinyal seçeneklerinin karışımıyla birlikte, her bir bekleyen çağrı eşit olarak değerlendirilir. Bu, MQGMO_WAIT ile MQGMO_SET_SIGNAL belirtilirse bir hatadır. İşaret bekleyen bir kuyruk tanıtıcısı ile bu seçeneği belirlemek de bir hatadır.
- IndexType / MQIT_MSG_TOKEN içeren bir kuyruk için MQGMO_WAIT ya da MQGMO_SET_SIGNAL belirtirseniz, seçim ölçütlerine izin verilmez. Bu şu anlama gelir:
 - version-1 MQGMO kullanıyorsanız, MQGET çağrısında MQMI_NONE ve MQCI_NONE olarak belirtilen MQMD 'deki MsgId ve CorrelId alanlarını ayarlayın.
 - version-2 ya da üstü MQGMO kullanıyorsanız, MatchOptions alanını MQMO_NONE olarak ayarlayın.
- Paylaşılan bir kuyruktaki MQGET çağrısı için ve çağrı bir göz atma isteği ya da bir grup iletisinin yıkıcı bir alma isteğidir ve ne MsgId ne de CorrelId eşleştirilmez, sinyal ECB 'niz 200 milisaniyeden sonra MQEC_MSG_HAS olarak gönderilir.

Kuyruk MQEC_WAIT_INTERVAL_EXPIRED ile gönderilinceye kadar bekleme aralığı sona erinceye kadar, kuyruğa uygun bir ileti gelmemiş olsa da bu durum oluşur. MQEC_MSG_RETRIEVE gönderildiğinde, varsa iletiyi almak için ikinci bir MQGET çağrısı yayınlamanız gerekir.

Bu teknik, ileti gelişinin zamanında bildirilmesini sağlamak için kullanılır, ancak paylaşılmayan bir kuyruktaki benzer bir çağrı sırası ile karşılaştırıldığında beklenmeyen bir işleme ek yükü olarak görünebilir.

MQGMO_WAIT , MQGMO_BROWSE_MSG_UNDER_CURSOR ya da MQGMO_MSG_UNDER_CURSOR ile belirtildiyse yoksayılar; hata oluşmadı.

MQGMO_NO_WAIT

Uygun bir ileti yoksa uygulama beklemeyebilir. MQGMO_NO_WAIT , MQGMO_WAIT' in tersidir. MQGMO_NO_WAIT , program belgelerine yardımcı olmak için tanımlanır. İkisi de belirtilmezse, varsayılan değer budur.

MQGMO_SET_SIGNAL

Bu seçeneği Signal1 ve Signal2 alanlarıyla kullanın. Bir iletinin gelmesini beklerken uygulamaların diğer işlemlerle devam etmesini sağlar. Ayrıca, (uygun işletim sistemi olanakları varsa) uygulamaların birden fazla kuyruğa gelen iletileri beklemesine izin verir.

Not: MQGMO_SET_SIGNAL seçeneği, ortama özgüdür; bu seçeneği, kapı oluşturmak istediğiniz uygulamalar için kullanmayın.

İki durumda, arama, bu seçenek belirtilmemiş gibi tamamlanır:

1. Şu anda kullanılabilir bir ileti, ileti tanımlayıcısında belirtilen ölçütleri karşılıyorsa.
2. Bir parametre hatası ya da başka bir zaman uyumlu hata algılanırsa.

İleti tanımlayıcıda belirtilen ölçütlere uyan bir ileti yoksa, denetim bir iletinin gelmesini beklemeyen uygulamaya döner. **CompCode** ve **Reason** parametreleri MQCC_WARNING ve MQRC_SIGNAL_REQUEST_ACCEPTED olarak ayarlanır. İleti tanımlayıcıdaki diğer çıkış alanları ve

MQGET çağrısının çıkış parametreleri ayarlanmadı. Daha sonra uygun bir mesaj geldiğinde, sinyal ECB 'nin gönderilmesiyle teslim edilir.

Daha sonra, çağırana iletiyi almak için MQGET çağrısını yeniden yayınlamalıdır. Uygulama, işletim sistemi tarafından sağlanan işlevleri kullanarak bu sinyali bekleyebilir.

İşletim sistemi çoklu bir bekleme mekanizması sağlıyorsa, bu mekanizmayı kullanarak birkaç kuyruklardan herhangi birine ileti gelmesini bekleyebilirsiniz.

Sıfır dışında bir `WaitInterval` belirtilirse, bekleme aralığı sona erdikten sonra sinyal teslim edilir. Kuyruk yöneticisi de bekleme işlemini iptal edebilir; bu durumda sinyal teslim edilir.

Birden çok MQGET çağrısı aynı ileti için bir sinyal ayarlayabilir. Uygulamaların etkinleştirildiği sıra, MQGMO_WAIT için açıklanan sıralamayla aynıdır.

Aynı iletiyi birden çok MQGET çağrısı bekliyorsa, bekleyen her çağrı eşit olarak değerlendirilir. Çağrılar, bekleme ve sinyal seçeneklerinin bir karışımını içerebilir.

Belirli koşullar altında MQGET çağrısı bir iletiyi alabilir ve aynı iletinin gelmesinden kaynaklanan bir sinyal teslim edilebilir. Bir sinyal gönderildiğinde, herhangi bir iletinin kullanılabilmesi için bir uygulama hazırlanmalıdır.

Bir kuyruk tanıtıcısı, bekleyen birden fazla sinyal isteği olamaz.

Bu seçenek, aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- MQGMO_UNLOCK
- MQGMO_WAIT

Paylaşılan bir kuyruktaki MQGET çağrısı için ve çağrı bir göz atma isteği ya da bir grup iletisinin yıkıcı bir alma isteğidir ve `MsgId` ya da `CorrelId` eşleşmez, kullanıcının sinyal ECB 'si 200 milisaniye sonra MQEC_MSG_ARRIVED gönderilir.

Bu durum, kuyruğa uygun bir ileti gelmemiş olsa da, bekleme süresi doluncaya kadar, kuyruk MQEC_WAIT_INTERVAL_EXPIRE ile gönderildiğinde ortaya çıkar. MQEC_MSG_ARRIVED gönderildiğinde, varsa iletiyi almak için ikinci bir MQGET çağrısı yayınlamanız gerekir.

Bu teknik, ileti gelişinin zamanında bildirilmesini sağlamak için kullanılır, ancak paylaşılmayan bir kuyruktaki benzer bir çağrı sırası ile karşılaştırıldığında beklenmeyen bir işleme ek yükü olarak görünebilir.

İletiler seyrek olarak eklendiğinde bu, etkin bir ileti alma yöntemi değildir. Göz atma vakasına ilişkin bu ek yükü önlemek için, MQGET çağrısında `MsgId` (`MsgId` tarafından izinlenmediyse ya da izinlenmediyse) ya da `CorrelId` (`CorrelId` tarafından izinlendiyse) belirtin.

z/OS Bu seçenek yalnızca z/OS üzerinde desteklenir.

MQGMO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturma durumundaysa, MQGET çağrısına başarısız olması için zorlayın.

z/OS z/OS işletim sistemlerinde bu seçenek, bağlantı (bir CICS ya da IMS uygulaması için) susturma durumundaysa MQGET çağrısına da başarısız olması için zorlar.

Bu seçenek MQGMO_WAIT ya da MQGMO_SET_SIGNAL ile belirtilirse ve kuyruk yöneticisi susturma durumuna girdiğinde bekleme ya da sinyal bekletmemişse:

- Bekleme iptal edilir ve çağrı MQCC_FAILED tamamlanma kodunu MQRC_Q_MGR QUIESCING ya da MQRC_CONNECTION QUIESCING neden koduyla döndürür.
- Sinyal, ortama özgü bir sinyal tamamlama koduyla iptal edilir.

z/OS z/OS sistemlerinde sinyal, MQEC_Q_MGR QUIESCING ya da MQEC_CONNECTION QUIESCING kolay tamamlama koduyla tamamlanır.

MQGMO_FAIL_IF QUIESCING belirtilmezse ve kuyruk yöneticisi ya da bağlantı susturma durumuna girerse, bekleme ya da sinyal iptal edilmez.

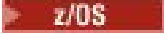
Eşitleme noktası seçenekleri

Aşağıdaki seçenekler, bir iş birimi içindeki MQGET aramasının katılımıyla ilgilidir:

MQGMO_SYNCPOINT

İstek, normal iş birimi protokolleri içinde çalışmaktadır. İleti, diğer uygulamalar tarafından kullanılamıyor olarak işaretlenir, ancak yalnızca iş birimi kesinleştirildiğinde kuyruktan silinir. İş birimi geriletildiyse, ileti yeniden kullanılır kılır.

MQGMO_SYNCPOINT ve MQGMO_NO_SYNCPOINT ayarını belirsiz bırakabilirsiniz. Bu durumda, iş birimi protokollerine alma isteğinin dahil edilmesi, kuyruk yöneticisini çalıştıran ortam tarafından belirlenir. Uygulamayı çalıştıran ortam tarafından belirlenmez.

-  z/OS' da, alma isteği bir iş birimi içindedir.
- z/OS dışındaki tüm ortamlarda alma isteği bir iş birimi içinde değildir.

Bu farklılıklar nedeniyle, bağlantı noktası olarak kullanmak istediğiniz bir uygulama bu seçeneğin varsayılan değerine ayarlanmasına izin vermemelidir; MQGMO_SYNCPOINT ya da MQGMO_NO_SYNCPOINT belirtik olarak belirtilmelidir.

Bu seçenek, aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- 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

İstek, normal iş birimi iletişim kuralları içinde çalışmaktadır, ancak yalnızca alınan ileti kalıcı ise. Kalıcı bir ileti, MQMD' deki Persistence alanında MQPER_PERSISTENT değerini içeriyor.

- İleti kalıcıysa, kuyruk yöneticisi çağırısı uygulamanın MQGMO_SYNCPOINT belirttiği gibi işler.
- İleti kalıcı değilse, kuyruk yöneticisi çağırısı uygulamanın MQGMO_NO_SYNCPOINT belirttiği gibi işler.

Bu seçenek, aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- 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

Bu seçenek aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  z/OS

ve bu sistemlere bağlı IBM MQ MQI clients için.

MQGMO_NO_SYNCPOINT

İstek, normal iş birimi protokollerinin dışında çalışmaktır. Göz atma seçeneği olmayan bir ileti alırsanız, ileti kuyruktan hemen silinir. İş birimi yedeklenerek ileti yeniden kullanılabilir kılınmaz.

MQGMO_BROWSE_FIRST ya da MQGMO_BROWSE_NEXT belirtirseniz bu seçenek kabul edilir.

MQGMO_SYNCPOINT ve MQGMO_NO_SYNCPOINT ayarını belirsiz bırakabilirsiniz. Bu durumda, iş birimi protokollerine alma isteğinin dahil edilmesi, kuyruk yöneticisini çalıştıran ortam tarafından belirlenir. Uygulamayı çalıştıran ortam tarafından belirlenmez.

- **z/OS** z/OS' da, alma isteği bir iş birimi içindedir.
- z/OS'dışındaki tüm ortamlarda alma isteği bir iş birimi içinde değildir.

Bu farklılıklar nedeniyle, bağlantı vermek istediğiniz bir uygulama bu seçeneğin varsayılan değerine ayarlanmasına izin vermemelidir; MQGMO_SYNCPOINT ya da MQGMO_NO_SYNCPOINT belirttik olarak belirtilmelidir.

Bu seçenek, aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- MQGMO_MARK_SKIP_BACKOUT
- MQGMO_SYNCPOINT
- MQGMO_SYNCPOINT_IF_PERSISTENT

z/OS MQGMO_MARK_SKIP_BACKOUT

Kuyrukta bu seçenekle işaretli iletiyi yeniden belirtmeden bir iş birimini geri çıkarın.

Bu seçenek yalnızca z/OS üzerinde desteklenir.

Bu seçenek belirtilirse, MQGMO_SYNCPOINT da belirtilmelidir. MQGMO_MARK_SKIP_BACKOUT , aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- MQGMO_BROWSE_FIRST
- MQGMO_BROWSE_MSG_UNDER_CURSOR
- MQGMO_BROWSE_NEXT
- MQGMO_LOCK
- MQGMO_NO_SYNCPOINT
- MQGMO_SYNCPOINT_IF_PERSISTENT
- MQGMO_UNLOCK

Not: IMS ve CICS sistemlerinde, MQGMO_MARK_SKIP_BACKOUT ile işaretli bir iletiyi içeren bir iş birimini geri çektikten sonra bir ekstras IBM MQ çağrısı yayınlamanız gerekebilir. İşaretli iletiyi içeren yeni iş birimini kesinleştirmeden önce bir IBM MQ çağrısı yayınlamanız gerekir. Arama, istediğiniz herhangi bir IBM MQ araması olabilir.

1. IMS işletim sistemlerinde IMS APAR PN60855 uygulamadıysanız ve bir IMS MPP ya da BMP uygulaması çalıştırıyorsanız.
2. CICS üzerinde, herhangi bir uygulama çalıştırıyorsanız.

Her iki durumda da, geriletiyi içeren yeni iş birimini kesinleştirmeden önce herhangi bir IBM MQ çağrısı yayınlayın.

Not: Bir iş birimi içinde, yalnızca bir alma isteği atlanıyor olarak işaretlenebilir, ayrıca yok ya da birkaç işaretlenmemiş alma isteği de olabilir.

Bir uygulama bir iş biriminden geri çekilirse, MQGMO_MARK_SKIP_BACKOUT kullanılarak alınan bir ileti önceki durumuna geri yüklenmez. Diğer kaynak güncellemeleri geriletildi. İleti, geriletme isteği tarafından başlatılan yeni bir iş biriminde alınmış gibi işlenir. İleti, MQGMO_MARK_SKIP_BACKOUT seçeneği olmadan alınır.

Bazı kaynaklar değiştirildikten sonra, iş biriminin başarıyla tamamlanamadığı anlaşılırsa MQGMO_MARK_SKIP_BACKOUT yararlı olur. Bu seçeneği atlarsanız, iş biriminin geri çekilmesi,

kuyruktaki iletiyi yeniden yürürlüğe alır. İleti bir sonraki alındığında, aynı olay sırası yeniden ortaya çıkar.

Ancak, özgün MQGET çağrısında MQGMO_MARK_SKIP_BACKOUT değerini belirlerseniz, iş biriminin yedeklenmesi diğer kaynaklara ilişkin güncellemeleri geri alır. İleti, yeni bir iş birimi altında alınmış gibi işlenir. Uygulama uygun hata işlemeyi gerçekleştirebilir. Özgün iletiyi gönderene bir rapor iletilisi gönderebilir ya da özgün iletiyi gitmeyen ileti kuyruğuna yerleştirebilir. Daha sonra yeni iş birimini kesinleştirebilir. Yeni iş birimi kesinleştirildiğinde, ileti özgün kuyruktan kalıcı olarak kaldırılır.

MQGMO_MARK_SKIP_BACKOUT , tek bir fiziksel iletiyi işaretler. İleti bir ileti grubuna aitse, gruptaki diğer iletiler işaretlenmez. Benzer şekilde, işaretli ileti mantıksal iletinin bir parçası ise, mantıksal iletideki diğer bölümler işaretlenmez.

Bir gruptaki herhangi bir ileti işaretlenebilir, ancak iletiler MQGMO_LOGICAL_ORDERkullanılarak alınır, gruptaki ilk iletiyi işaretlemek avantajlıdır. İş birimi geriletildiyse, ilk (işaretli) ileti yeni iş birimine taşınır. Gruptaki ikinci ve sonraki iletiler kuyrukta yeniden kullanılır. Kuyrukta kalan iletiler, MQGMO_LOGICAL_ORDERkullanılarak başka bir uygulama tarafından alınamıyor. Gruptaki ilk ileti artık kuyrukta değil. Ancak, çalışma birimini destekleyen uygulama, MQGMO_LOGICAL_ORDER seçeneğini kullanarak yeni iş birimine ikinci ve sonraki iletileri alabilir. İlk ileti zaten alındı.

Ara sıra yeni iş birimini geri almanız gerekebilir. Örneğin, gitmeyen iletiler kuyruğu dolu olduğundan ve iletinin atılması gerekmediğinden. Yeni iş biriminin geri çekilmesi, özgün kuyruktaki iletiyi yeniden yürürlüğe koyar ve iletinin kaybolmasını önler. Ancak, bu durumda işleme devam edemez. Yeni iş birimini geri çektikten sonra, uygulamanın işletmene ya da yöneticiye kurtarılamaz bir hata olduğunu bildirmesi ve daha sonra bitirmesi gerekir.

MQGMO_MARK_SKIP_BACKOUT yalnızca, alma isteğini içeren iş birimi, uygulama bu isteği geri çekerek kesintiye uğratılırsa çalışır. Alma isteğini içeren iş birimi, hareket ya da sistem başarısız olduğu için geri çekilirse, MQGMO_MARK_SKIP_BACKOUT yoksayılır. Bu seçenek kullanılarak alınan iletiler, bu seçenek olmadan alınan iletiyle aynı şekilde kuyrukta yeniden kullanılır.

Göz Atma Seçenekleri

Aşağıdaki seçenekler, kuyruktaki iletilere göz atma ile ilgilidir:

MQGMO_BROWSE_FIRST

MQOO_BROWSE seçeneğiyle bir kuyruk açıldığında, mantıksal olarak kuyruktaki ilk iletiden önce konumlandırılmış bir göz atma imleci oluşturulur. Daha sonra, kuyruktan iletileri yıkıcı olmayan bir şekilde almak için MQGMO_BROWSE_FIRST, MQGMO_BROWSE_NEXTya da MQGMO_BROWSE_MSG_UNDER_CURSOR seçeneğini belirten MQGET çağrılarını kullanabilirsiniz. Göz atma imleci, kuyruktaki iletilerin içindeki konumu işaretler. Bu konumdan sonraki MQGET çağrısı MQGMO_BROWSE_NEXT uygun bir iletiyi arar.

MQGMO_BROWSE_FIRST , aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- MQGMO_BROWSE_MSG_UNDER_CURSOR
- MQGMO_BROWSE_NEXT
- MQGMO_MARK_SKIP_BACKOUT
- MQGMO_MSG_UNDER_CURSOR
- MQGMO_SYNCPOINT
- MQGMO_SYNCPOINT_IF_PERSISTENT
- MQGMO_UNLOCK

Kuyruk göz atmak için açılmamışsa da bu bir hatadır.

MQGMO_BROWSE_FIRST içeren bir MQGET çağrısı, göz atma imlecinin önceki konumunu yoksayar. Kuyruktaki, ileti tanımlayıcısında belirtilen koşullara uyan ilk ileti alınır. İleti kuyrukta kalır ve göz atma imleci bu iletide konumlandırılır.

Bu çağrıdan sonra, göz atma imleci döndürülen iletide konumlandırılır. İleti, MQGMO_BROWSE_NEXT ile bir sonraki MQGET çağrısı yayınlanmadan önce kuyruktan kaldırılabilir. Bu durumda, göz atma imleci, artık boş olsa da, iletinin bulunduğu kuyruktaki konumda kalır.

İletiyi kuyruktan kaldırmak için, MQGMO_MSG_UNDER_CURSOR seçeneğini göz atmayan bir MQGET çağrısıyla kullanın.

Göz atma imleci, aynı *Hobj* tanıtıcısı kullanılsa bile, göz atma özelliği olmayan bir MQGET çağrısıyla taşınmaz. MQCC_FAILEDtamamlanma kodunu ya da MQRC_TRUNCATED_MSG_FAILEDneden kodunu döndüren bir göz atma MQGET çağrısıyla da taşınmaz.

Göz atılan iletiyi kilitlemek için bu seçenekle MQGMO_LOCK seçeneğini belirleyin.

MQGMO_BROWSE_FIRST seçeneğini, iletilerin gruplar ve mantıksal ileti kesimleri halinde işlenmesini denetleyen MQGMO_* ve MQMO_* seçeneklerinin geçerli bir birleşimiyle belirleyebilirsiniz.

MQGMO_LOGICAL_ORDERbelirtirseniz, iletilere mantıksal sırayla göz atılır. Bu seçeneği atlarsanız, iletilere fiziksel sırayla göz atılır. MQGMO_BROWSE_FIRSTbelirtirseniz, mantıksal sıra ile fiziksel sıra arasında geçiş yapabilirsiniz. MQGMO_BROWSE_NEXT kullanan sonraki MQGET çağrıları, kuyruk tanıtıcısı için belirlenen en son çağrıyla aynı sırada kuyruğa göz atın MQGMO_BROWSE_FIRST .

Kuyruk yöneticisi, MQGET çağrıları için iki grup ve bölüm bilgisi kümesini tutar. Göz atma çağrılarına ilişkin grup ve bölüm bilgileri, kuyruktan ileti kaldran çağrılara ilişkin bilgilerden ayrı olarak tutulur. MQGMO_BROWSE_FIRSTbelirtirseniz, kuyruk yöneticisi göz atmak için grup ve bölüm bilgilerini yoksayar. Kuyruğu, yürürlükteki grup ve yürürlükteki mantıksal ileti yokmuş gibi tarar. MQGET çağrısı başarılı olursa, tamamlama kodu MQCC_OK ya da MQCC_WARNING, göz atmaya ilişkin grup ve bölüm bilgileri, döndürülen iletinin grup ve bölüm bilgilerine ayarlanır. Arama başarısız olursa, grup ve bölüm bilgileri çağrıdan önceki bilgilerle aynı kalır.

MQGMO_BROWSE_NEXT

Göz atma imlecini, MQGET çağrısında belirtilen seçim ölçütlerini karşılayan kuyruktaki sonraki iletiye ilerleyin. İleti uygulamaya döndürülür, ancak kuyrukta kalır.

MQGMO_BROWSE_NEXT , aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- MQGMO_BROWSE_FIRST
- MQGMO_BROWSE_MSG_UNDER_CURSOR
- MQGMO_MARK_SKIP_BACKOUT
- MQGMO_MSG_UNDER_CURSOR
- MQGMO_SYNCPOINT
- MQGMO_SYNCPOINT_IF_PERSISTENT
- MQGMO_UNLOCK

Kuyruk göz atmak için açılmamışsa da bu bir hatadır.

MQGMO_BROWSE_NEXT , kuyruk göz atmak üzere açıldıktan sonra kuyruğa göz atmak için ilk çağrıya, MQGMO_BROWSE_FIRSTile aynı şekilde davranır.

İmleç altındaki ileti, MQGMO_BROWSE_NEXT ile sonraki MQGET çağrısı yayınlanmadan önce kuyruktan kaldırılabilir. Göz atma imleci, artık boş olsa da, iletinin bulunduğu kuyrukta mantıksal olarak kalır.

İletiler şu iki yoldan biriyle kuyrukta saklanır:

- Öncelik içinde FIFO (MQMDS_PRIORITY) ya da
- Önceliğe bakılmaksızın FIFO (MQMDS_FIFO)

MsgDeliverySequence kuyruk özneliği hangi yöntemin geçerli olduğunu gösterir (ayrıntılar için bkz. “Kuyruklara ilişkin öznelikler” sayfa 826).

Bir kuyrukta MsgDeliverySequence / MQMDS_PRIORITYolabilir. Kuyruğa, göz atma imlecinin gösterdiğinden daha yüksek önceliğe sahip bir ileti gelir. Bu durumda, MQGMO_BROWSE_NEXTkullanılarak kuyruğun yürürlükteki taraması sırasında yüksek öncelikli ileti

bulunmaz. Yalnızca göz atma imleci MQGMO_BROWSE_FIRST ile ilk durumuna getirildikten sonra ya da kuyruk yeniden açılarak bulunabilir.

Gerekliyse, MQGMO_MSG_UNDER_CURSOR seçeneği, iletiyi kuyruktan kaldırmak için göz atma dışı bir MQGET çağrısıyla birlikte kullanılabilir.

Göz atma imleci, aynı Hobj tanıtıcısı kullanılarak MQGET çağrılarında göre taşınmaz.

Göz atılan iletiyi kilitlemek için bu seçenekle MQGMO_LOCK seçeneğini belirleyin.

MQGMO_BROWSE_NEXT seçeneğini, iletilerin gruplar ve mantıksal ileti kesimleri halinde işlenmesini denetleyen MQGMO_* ve MQMO_* seçeneklerinin geçerli bir birleşimiyle belirleyebilirsiniz.

MQGMO_LOGICAL_ORDER belirtirseniz, iletilere mantıksal sırayla göz atılır. Bu seçeneği atlarsanız, iletilere fiziksel sırayla göz atılır. MQGMO_BROWSE_FIRST belirtirseniz, mantıksal sıra ile fiziksel sıra arasında geçiş yapabilirsiniz. MQGMO_BROWSE_NEXT kullanan sonraki MQGET çağrılarını, kuyruk tanıtıcısı için belirlenen en son çağrıyla aynı sırada kuyruğa göz atın MQGMO_BROWSE_FIRST . Bu koşul karşılanmazsa, çağrı MQRC_INCONSISTENT_BROWSE neden koduyla başarısız olur.

Not: MQGMO_LOGICAL_ORDER belirtilmemişse, bir ileti grubunun sonunun ötesine göz atmak için MQGET çağrısını kullanırken dikkatli olun. Örneğin, gruptaki son iletinin kuyruktaki gruptaki ilk iletiden önce olduğunu varsayın. Grubun sonunun ötesine göz atmak için MQGMO_BROWSE_NEXT seçeneğinin kullanılması, MQMO_MATCH_MSG_SEQ_NUMBER with MsgSeqNumber set to 1 seçeneğinin belirlenmesi, grupta göz atılan ilk iletiyi döndürür. Bu sonuç hemen oluşabilir ya da araya giren gruplar varsa, daha sonra bir dizi MQGET çağrısı yapılabilir. Aynı değer, bir grupta olmayan mantıksal bir ileti için de geçerlidir.

Göz atma çağrılarında ilişkin grup ve bölüm bilgileri, kuyruktan ileti kaldıran çağrılara ilişkin bilgilerden ayrı olarak tutulur.

MQGMO_BROWSE_MSG_UNDER_CURSOR

MQGMO' nun MatchOptions alanında belirtilen MQMO_* seçeneklerinden bağımsız olarak, göz atma imlecinin gösterdiği iletiyi alın.

MQGMO_BROWSE_MSG_UNDER_CURSOR , aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- MQGMO_BROWSE_FIRST
- MQGMO_BROWSE_NEXT
- MQGMO_MARK_SKIP_BACKOUT
- MQGMO_MSG_UNDER_CURSOR
- MQGMO_SYNCPOINT
- MQGMO_SYNCPOINT_IF_PERSISTENT
- MQGMO_UNLOCK

Kuyruk göz atmak için açılmamışsa da bu bir hatadır.

Göz atma imlecinin gösterdiği ileti, MQGMO_BROWSE_FIRST ya da MQGMO_BROWSE_NEXT seçeneği kullanılarak en son alınan iletidir. Açıldığından bu yana bu kuyruk için bu çağrılarının hiçbirisi yayınlanmamışsa çağrı başarısız olur. Göz atma imlecinin altındaki ileti yıkıcı bir şekilde alındıysa da çağrı başarısız olur.

Göz atma imlecinin konumu bu çağrı tarafından değiştirilmez.

MQGMO_MSG_UNDER_CURSOR seçeneği, iletiyi kuyruktan kaldırmak için göz atmayan bir MQGET çağrısıyla kullanılabilir.

Göz atma imleci, aynı Hobj tanıtıcısı kullanılsa bile, göz atma özelliği olmayan bir MQGET çağrısıyla taşınmaz. MQCC_FAILED tamamlanma kodunu ya da MQRC_TRUNCATED_MSG_FAILED neden kodunu döndüren bir göz atma MQGET çağrısıyla da taşınmaz.

MQGMO_BROWSE_MSG_UNDER_CURSOR , MQGMO_LOCK ile belirtilirse:

- Kilitli bir ileti varsa, bu iletinin imlecinin altındaki ileti olması gerekir; böylece, kilidi açılmadan ve kilitlenmeden ileti döndürülür. İleti kilitli kalır.

- Kilitli ileti yoksa ve göz atma imlecinin altında bir ileti varsa, ileti kilitlenir ve uygulamaya döndürülür. Göz atma imlecinin altında ileti yoksa, arama başarısız olur.

MQGM0_BROWSE_MSG_UNDER_CURSOR , MQGM0_LOCK olmadan belirtilirse:

- Kilitli bir ileti varsa, imlecinin altındaki ileti olmalıdır. İleti uygulamaya döndürülür ve kilidi açılır. İletin kilidi açıldığından, iletiye yeniden göz atılabileceğinin ya da aynı uygulama tarafından yıkıcı bir şekilde alınabileceğinin garantisi yoktur. Kuyruktan ileti alan başka bir uygulama tarafından yıkıcı bir şekilde alınmış olabilir.
- Kilitli ileti yoksa ve göz atma imlecinin altında bir ileti varsa, bu ileti uygulamaya döndürülür. Göz atma imlecinin altında ileti yoksa, çağrı başarısız olur.

MQGM0_COMPLETE_MSG MQGM0_BROWSE_MSG_UNDER_CURSOR ile belirtilirse, göz atma imleci MQMD 'deki Offset alanı sıfır olan bir iletiyi tanımlamalıdır. Bu koşul karşılanmazsa, arama MQRC_INVALID_MSG_UNDER_CURSOR neden koduyla başarısız olur.

Göz atma çağrılarında ilişkin grup ve bölüm bilgileri, kuyruktan ileti kaldıran çağrılara ilişkin bilgilerden ayrı olarak tutulur.

MQGM0_MSG_UNDER_CURSOR

MQGM0' nun MatchOptions alanında belirtilen MQMO_* seçeneklerinden bağımsız olarak, göz atma imlecinin gösterdiği iletiyi alın. İleti kuyruktan kaldırılır.

Göz atma imlecinin gösterdiği ileti, MQGM0_BROWSE_FIRST ya da MQGM0_BROWSE_NEXT seçeneği kullanılarak en son alınan ilettir.

MQGM0_COMPLETE_MSG MQGM0_MSG_UNDER_CURSOR ile belirtilirse, göz atma imleci MQMD 'deki Offset alanı sıfır olan bir iletiyi tanımlamalıdır. Bu koşul karşılanmazsa, arama MQRC_INVALID_MSG_UNDER_CURSOR neden koduyla başarısız olur.

Bu seçenek, aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- MQGM0_BROWSE_FIRST
- MQGM0_BROWSE_MSG_UNDER_CURSOR
- MQGM0_BROWSE_NEXT
- MQGM0_UNLOCK

Kuyruk hem göz atma hem de giriş için açılmamışsa da bu bir hatadır. Göz atma imleci şu anda alınabilir bir iletiyi göstermezse, MQGET çağrısı bir hata döndürdü.

MQGM0_MARK_BROWSE_HANDLE

Başarılı bir MQGET tarafından döndürülen ya da döndürülen MsgToken ile tanımlanan ileti işaretli. İşaret, aramada kullanılan nesne tanıtıcısına özgüdür.

İleti kuyruktan kaldırılmadı.

MQGM0_MARK_BROWSE_HANDLE yalnızca aşağıdaki seçeneklerden biri de belirtildiyse geçerlidir:

- MQGM0_BROWSE_FIRST
- MQGM0_BROWSE_MSG_UNDER_CURSOR
- MQGM0_BROWSE_NEXT

MQGM0_MARK_BROWSE_HANDLE , aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- MQGM0_ALL_MSGS_AVAILABLE
- MQGM0_ALL_SEGMENTS_AVAILABLE
- MQGM0_COMPLETE_MSG
- MQGM0_LOCK
- MQGM0_LOGICAL_ORDER
- MQGM0_UNLOCK

Aşağıdaki olaylardan biri gerçekleşinceye kadar ileti bu durumda kalır:

- İlgili nesne tanıtıcısı, olağan ya da başka bir şekilde kapatılır.
- İleti, MQGMO_UNMARK_BROWSE_HANDLE seçeneğiyle MQGET adresine yapılan bir çağrı tarafından bu tanıtıcı için işaretlenmemiş.
- İleti, MQCC_OK ya da MQCC_WARNING ile tamamlanan yıkıcı MQGET çağrısından döndürülür. MQGET daha sonra geriye işlense de ileti durumu değişmeden kalır.
- İletinin süresi dolacak.

MQGMO_MARK_BROWSE_CO_OP

Başarılı bir MQGET tarafından döndürülen ya da döndürülen *MsgToken* tarafından tanımlanan ileti, işbirliği kümesindeki tüm tutamaçlar için işaretlenir.

İşbirliği düzeyi işareti, ayarlanmış olabilecek herhangi bir tanıtıcı düzeyi işaretine ek olarak.

İleti kuyruktan kaldırılmaz.

MQGMO_MARK_BROWSE_CO_OP yalnızca, kullanılan nesne tanıtıcısı MQOPEN belirtilen MQOO_CO_OP çağrısıyla döndürüldüyse geçerlidir. Aşağıdaki MQGMO seçeneklerinden birini de belirtmeniz gerekir:

- MQGMO_BROWSE_FIRST
- MQGMO_BROWSE_MSG_UNDER_CURSOR
- MQGMO_BROWSE_NEXT

Bu seçenek, aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- MQGMO_ALL_MSGS_AVAILABLE
- MQGMO_ALL_SEGMENTS_AVAILABLE
- MQGMO_COMPLETE_MSG
- MQGMO_LOCK
- MQGMO_LOGICAL_ORDER
- MQGMO_UNLOCK

İleti önceden işaretliyse ve MQGMO_UNMARKED_BROWSE_MSG seçeneği belirtilmediyse, arama MQCC_FAILED ve neden kodu MQRC_MSG_MARKED_BROWSE_CO_OP ile başarısız olur.

Aşağıdaki olaylardan biri gerçekleşinceye kadar ileti bu durumda kalır:

- Birlikte çalışma kümesindeki tüm nesne tutamaçları kapalıdır.
- İleti, işbirliği içinde çalışan tarayıcılar için MQGMO_UNMARK_BROWSE_CO_OP seçeneğiyle MQGET adresine yapılan bir aramayla işaretlenmemiş.
- İleti, kuyruk yöneticisi tarafından otomatik olarak işaretlenir.
- İleti, göz atmayan bir MQGET çağrısından döndürülür. MQGET daha sonra geriye işlense de ileti durumu değişmeden kalır.
- İletinin süresi dolacak.

MQGMO_UNMARKED_BROWSE_MSG

MQGMO_UNMARKED_BROWSE_MSG belirten bir MQGET çağrısı, tanıtıcısı için işaretlenmemiş olarak kabul edilen bir ileti döndürür. İleti, tanıtıcısı için işaretlenmişse bir ileti döndürmez. Kuyruk, MQOO_CO_OP seçeneğiyle MQOPEN adresine yapılan bir çağrı tarafından açıldıysa ve ileti, işbirliği kümesinin bir üyesi tarafından işaretlendiyse, iletiyi döndürmez.

Bu seçenek, aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- MQGMO_ALL_MSGS_AVAILABLE
- MQGMO_ALL_SEGMENTS_AVAILABLE
- MQGMO_COMPLETE_MSG
- MQGMO_LOCK
- MQGMO_LOGICAL_ORDER

- MQGMO_UNLOCK

MQGMO_UNMARK_BROWSE_CO_OP

Bu seçeneği belirten bir MQGET çağrısından sonra, ileti artık işbirliği kümesi için işaretlenecek işbirliği tutamaçları kümesindeki açık tutamaçlar tarafından dikkate alınmaz. İleti, bu çağrıdan önce tanıtıcı düzeyinde işaretlenmişse, ileti yine de tanıtıcı düzeyinde işaretlenmiş olarak kabul edilir.

MQGMO_UNMARK_BROWSE_CO_OP seçeneğinin kullanılması yalnızca, MQOO_CO_OP seçeneğiyle MQOPEN ' e yapılan başarılı bir çağrı tarafından döndürülen bir tanıtıcı ile geçerlidir. MQGET , iletinin işbirliği içinde çalışan tutamaçlar kümesi tarafından işaretlenmiş olarak kabul edilmese de başarılı olur.

MQGMO_UNMARK_BROWSE_CO_OP , göz atmayan bir MQGET çağrısında ya da aşağıdaki seçeneklerden herhangi birinde geçerli değildir:

- 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

Bu seçeneği belirten bir MQGET çağrısından sonra, bulunan ileti artık bu tanıtıcı tarafından işaretlenmiş olarak kabul edilmez.

İleti bu tanıtıcı için işaretlenmese de arama başarılı olur.

Bu seçenek, göz atmayan bir MQGET çağrısında ya da aşağıdaki seçeneklerden biriyle geçerli değildir:

- 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

Kilit seçenekleri

Aşağıdaki seçenekler, kuyruktaki iletilerin kilitlenmesiyle ilgilidir:

MQGMO_KİLİK

Göz atılan iletiyi kilitleyin; böylece ileti, kuyruk için açık olan başka bir tutamaç tarafından görünmez olur. Bu seçenek yalnızca aşağıdaki seçeneklerden biri de belirtilirse belirtilebilir:

- MQGMO_BROWSE_FIRST
- MQGMO_BROWSE_NEXT
- MQGMO_BROWSE_MSG_UNDER_CURSOR

Her kuyruk tanıtıcısı için yalnızca bir ileti kilitlenebilir. İleti mantıksal bir ileti ya da fiziksel bir ileti olabilir:

- MQGMO_COMPLETE_MSG değerini belirlerseniz, mantıksal iletiyi oluşturan tüm ileti bölümleri kuyruk tanıtıcısı için kilitlenir. İletilerin tümü kuyruktaki bulunmalı ve alınabilmelidir.

- MQGMO_COMPLETE_MSG seçeneğini atlarsanız, kuyruk tanıtıcısına yalnızca tek bir fiziksel ileti kilitlenir. Bu ileti mantıksal iletinin bir parçası olursa, kilitli kısım, mantıksal iletiyi almak ya da göz atmak için MQGMO_COMPLETE_MSG komutunu kullanan diğer uygulamaları engeller.

Kilitli ileti her zaman göz atma imlecinin altındaki iletidir. İleti, MQGMO_MSG_UNDER_CURSOR seçeneğini belirten daha sonraki bir MQGET çağrısı tarafından kuyruktan kaldırılabilir. Kuyruk tanıtıcısını kullanan diğer MQGET çağrıları da iletiyi kaldırabilir (örneğin, kilitli iletinin ileti tanıtıcısını belirten bir çağrı).

Arama, MQCC_FAILED ya da MQCC_WARNING neden koduyla MQRC_TRUNCATED_MSG_FAILED tamamlanma kodunu döndürürse, hiçbir ileti kilitlenmez.

Uygulama iletiyi kuyruktan kaldırmazsa, kilit aşağıdaki işlemlerden biriyle serbest bırakılır:

- MQGMO_BROWSE_FIRST ya da MQGMO_BROWSE_NEXT belirtilerek bu tanıtıcı için başka bir MQGET çağrısı yayınlanıyor. Arama MQCC_OK ya da MQCC_WARNING ile tamamlanırsa kilit serbest bırakılır. Arama MQCC_FAILED ile tamamlanırsa ileti kilitli kalır. Ancak, aşağıdaki özel durumlar geçerlidir:
 - MQCC_WARNING , MQRC_TRUNCATED_MSG_FAILED ile döndürülürse iletinin kilidi açılmaz.
 - MQCC_FAILED , MQRC_NO_MSG_AVAILABLE ile döndürülürse iletinin kilidi açılır.

MQGMO_LOCK belirtilirse, döndürülen ileti kilitlenir. MQGMO_LOCK seçeneğini atlarsanız, aramadan sonra kilitli bir ileti olmaz.

MQGMO_WAIT belirtirseniz ve hemen kullanılabilir bir ileti yoksa, bekleme başlamadan önce özgün iletinin kilidi açılır.

- MQGMO_LOCK olmadan MQGMO_BROWSE_MSG_UNDER_CURSOR ile bu tanıtıcı için başka bir MQGET çağrısı yayınlanıyor. Arama MQCC_OK ya da MQCC_WARNING ile tamamlanırsa kilit serbest bırakılır. Arama MQCC_FAILED ile tamamlanırsa ileti kilitli kalır. Ancak, aşağıdaki kural dışı durum geçerlidir:
 - MQCC_WARNING , MQRC_TRUNCATED_MSG_FAILED ile döndürülürse iletinin kilidi açılmaz.
- MQGMO_UNLOCK ile bu tanıtıcı için başka bir MQGET çağrısı yayınlanıyor.
- Tutamacı kullanarak bir MQCLOSE araması yayınlanıyor. Uygulamanın sona ermesi nedeniyle MQCLOSE örtük olabilir.

Beraberindeki bir göz atma seçeneğini belirtmek için gerekli olan MQOO_BROWSE dışında MQGMO_LOCK seçeneğini belirtmek için özel bir MQOPEN seçeneği gerekmez.

MQGMO_LOCK , aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- MQGMO_MARK_SKIP_BACKOUT
- MQGMO_SYNCPOINT
- MQGMO_SYNCPOINT_IF_PERSISTENT
- MQGMO_UNLOCK

MQGMO_UNLOCK

Kilidinin açılacak ileti, daha önce MQGMO_LOCK seçeneğiyle bir MQGET çağrısı tarafından kilitlenmiş olmalıdır. Bu tanıtıcı için kilitli bir ileti yoksa, arama MQCC_WARNING ve MQRC_NO_MSG_LOCKED ile tamamlanır.

MQGMO_UNLOCK belirtirseniz, **MsgDesc**, **BufferLength**, **Buffer** ve **DataLength** parametreleri denetlenmez ya da değiştirilmez. *Buffer* içinde herhangi bir ileti döndürülmez.

MQGMO_UNLOCK değerini belirtmek için özel bir açma seçeneği gerekmez (ancak, kilit isteğini ilk sırada yayınlamak için MQOO_BROWSE gereklidir).

Bu seçenek, aşağıdakiler dışında hiçbir seçenikle geçerli değildir:

- MQGMO_NO_WAIT
- MQGMO_NO_SYNCPOINT

Bu seçeneklerin her ikisi de belirtilmiş ya da belirtilmemiş olarak kabul edilir.

İleti-veri seçenekleri

Aşağıdaki seçenekler, ileti kuyruktan okunduğunda ileti verilerinin işlenmesiyle ilgilidir:

MQMO_ACCEPT_TRUNCATED_MSG

İleti arabelleği tam iletiyi tutamayacak kadar küçükse, MQGET çağrısına arabelleği doldurmasına izin verin. MQGET , arabelleği alabildiği kadar iletiyle doldurur. Bir uyarı tamamlama kodu yayınlar ve işlemlerini tamamlar. Bu şu anlama gelir:

- İletilere göz atarken, göz atma imleci döndürülen iletiye ilerler.
- İletiler kaldırılırken, döndürülen ileti kuyruktan kaldırılır.
- Başka bir hata oluşmazsa, neden kodu MQRC_TRUNCATED_MSG_ACCEPTED döndürülür.

Bu seçenek olmadan, arabellek, tutabildiği kadar iletiyle dolu olur. Bir uyarı tamamlama kodu yayınlandı, ancak işleme tamamlanmadı. Bu şu anlama gelir:

- İletilere göz atarken, göz atma imleci gelişmiş değildir.
- İletiler kaldırılırken, ileti kuyruktan kaldırılmaz.
- Başka bir hata oluşmazsa, neden kodu MQRC_TRUNCATED_MSG_FAILED döndürülür.

MQMO_CONVERT

Bu seçenek, iletideki uygulama verilerini, MQGET çağrısında **MsgDesc** parametresinde belirtilen CodedCharSetId ve Encoding değerlerine uygun olacak şekilde dönüştürür. Veriler, **Buffer** değiştirgesine kopyalanmadan önce dönüştürülür.

İletin konması sırasında belirtilen Format alanı, iletideki verilerin türünü tanımlamak için dönüştürme işlemi tarafından kabul edilir. İleti verileri, yerleşik biçimler için kuyruk yöneticisi tarafından ve diğer biçimler için kullanıcı tarafından yazılan bir çıkış tarafından dönüştürülür. Veri dönüştürme çıkışının ayrıntıları için bkz. "[Veri dönüştürme çıkışı](#)" sayfa 895 .

- Dönüştürme başarılı olursa, MQGET çağrısından dönüşte **MsgDesc** parametresinde belirtilen CodedCharSetId ve Encoding alanları değişmez.
- Yalnızca dönüştürme başarısız olursa, ileti verileri dönüştürülmeden döndürülür. MsgDesc içindeki CodedCharSetId ve Encoding alanları, dönüştürülmemiş iletinin değerlerine ayarlanır. Bu durumda tamamlama kodu MQCC_WARNING olur.

Her iki durumda da, bu alanlar **Buffer** değiştirgesinde döndürülen ileti verilerinin karakter kümesi tanıtıcısını ve kodlamasını açıklar.

Kuyruk yöneticisinin dönüştürmeyi gerçekleştirdiği biçim adlarının bir listesi için "[MQMD-İleti tanımlayıcı](#)" sayfa 420 içinde açıklanan *Format* alanına bakın.

Grup ve bölüm seçenekleri

Aşağıdaki seçenekler, mantıksal ileti grupları ve kesimleri içindeki iletilerin işlenmesiyle ilgilidir. Seçenek açıklamalarından önce, önemli terimlerin bazı tanımları aşağıda verilmiştir:

Fiziksel ileti

Fiziksel ileti, bir kuyruğa yerleştirilebilecek ya da kuyruktan kaldırılabilir en küçük bilgi birimidir. Genellikle tek bir MQPUT, MQPUT1 ya da MQGET çağrısında belirtilen ya da alınan bilgilere karşılık gelir. Her fiziksel iletinin kendi ileti tanımlayıcısı (MQMD) vardır. Genellikle, fiziksel iletiler ileti tanıtıcısı için farklı değerlerle, MQMD' deki MsgId alanıyla ayırt edilir. Kuyruk yöneticisi farklı değerler uygulamaz.

Mantıksal ileti

Mantıksal ileti, tek bir uygulama bilgisi birimidir. Sistem kısıtlamaları olmadığında, mantıksal bir ileti fiziksel bir iletiyle aynıdır. Mantıksal iletiler büyükse, sistem kısıtlamaları, mantıksal bir iletiyi bölümler adı verilen iki ya da daha fazla fiziksel iletiye bölmeyi uygun ya da gerekli kılabilir.

Kesimlere ayrılmış bir mantıksal ileti, MQMD' de aynı boş olmayan grup tanıtıcısına (GroupId) sahip iki ya da daha fazla fiziksel iletiden oluşur. MQMD' deki MsgSeqNumber alanı, aynı ileti sıra numarasına sahiptir. Kesimler, MQMD' deki Offset alanı, kesim görelî konumu için farklı değerlerle ayırt edilir. Kesim görelî konumu, fiziksel iletideki verilerin mantıksal iletideki verilerin başlangıcından başlayarak

görelî konumudur. Her bir bölüm fiziksel bir ileti olduđu için, mantıksal bir iletideki bölümler genellikle farklı ileti tanıtıcılarına sahiptir.

Bölümlenmemiş, ancak bölümlenmeye gönderen uygulama tarafından izin verilen mantıksal bir ileti, boş olmayan bir grup tanıtıcısına da sahiptir. Bu durumda, mantıksal ileti bir ileti grubuna ait değilse, o grup tanıtıcısına sahip yalnızca bir fiziksel ileti vardır. Bölümlenme işlemi gönderen uygulama tarafından engellenen mantıksal iletilerin, mantıksal ileti bir ileti grubuna ait değilse, boş değerli bir grup tanıtıcısı (MQGI_NONE) vardır.

İleti grubu

İleti grubu, aynı boş olmayan grup tanıtıcısına sahip bir ya da daha fazla mantıksal iletiden oluşan bir kümedir. Gruptaki mantıksal iletiler, ileti sıra numarası için farklı değerlerle ayırt edilir. Sıra numarası, 1-n aralığındaki bir tamsayıdır; burada n, gruptaki mantıksal iletilerin sayısıdır. Mantıksal iletilerden biri ya da daha fazlası bölümlenmişse, grupta n'den fazla fiziksel ileti vardır.

MQGMO_LOGICAL_ORDER

MQGMO_LOGICAL_ORDER, kuyruk tanıtıcısı için ardışık MQGET çağrılarını tarafından iletilerin döndürüleceği sırayı denetler. Bu seçenek her çağrıda belirlenmelidir.

Aynı kuyruk tanıtıcısı için ardışık MQGET çağrılarını için MQGMO_LOGICAL_ORDER belirtilirse, gruptaki iletiler ileti sıra numaralarına göre döndürülür. Mantıksal ileti kesimleri, kesim görelî konumlarına göre verilen sırayla döndürülür. Bu sıra, bu ileti ve bölümlerin kuyruktaki oluşma sırasından farklı olabilir.

Not: MQGMO_LOGICAL_ORDER değerinin belirtilmesi, grupta ait olmayan ve bölüm olmayan iletiler üzerinde olumsuz sonuçlar doğurmaz. Gerçekte, bu tür iletilerin her biri yalnızca bir iletiden oluşan bir ileti grubuna aitmiş gibi ele alınır. Gruplarda, ileti bölümlerinde ve grupta olmayan bölümlenmemiş iletilerde ileti karışımı içeren kuyruklardan ileti alınırken MQGMO_LOGICAL_ORDER belirtilebilecek kadar güvenlidir.

İletileri istenen sırayla döndürmek için kuyruk yöneticisi, ardışık MQGET çağrılarını arasındaki grup ve bölüm bilgilerini saklar. Grup ve bölüm bilgileri, kuyruk tanıtıcısı için yürürlükteki ileti grubunu ve yürürlükteki mantıksal iletiyi tanıtır. Ayrıca, grup ve mantıksal ileti içindeki geçerli konumu ve iletilerin bir iş birimi içinde alınıp alınmadığını da tanımlar. Kuyruk yöneticisi bu bilgileri sakladığı için, uygulamanın her MQGET çağrısına başlamadan önce grup ve bölüm bilgilerini ayarlamasına gerek yoktur. Bu, uygulamanın MQMD'de GroupId, MsgSeqNumber ve Offset alanlarını ayarlamasına gerek olmadığı anlamına gelir. Ancak, uygulamanın her çağrıda MQGMO_SYNCPOINT ya da MQGMO_NO_SYNCPOINT seçeneğini doğru olarak ayarlaması gerekir.

Kuyruk açıldığında, yürürlükteki ileti grubu ve yürürlükteki mantıksal ileti yok. MQGET çağrısı MQMF_MSG_IN_GROUP işaretine sahip bir iletiyi döndürdüğünde, ileti grubu geçerli ileti grubu olur. MQGMO_LOGICAL_ORDER ardışık çağrılarda belirtildiğinde, aşağıdaki özelliklere sahip bir ileti döndürülünceye kadar bu grup geçerli grup olarak kalır:

- MQMF_LAST_MSG_IN_GROUP onsuz MQMF_SEGMENT (yani, gruptaki son mantıksal ileti bölümlenmez) ya da
- MQMF_LAST_MSG_IN_GROUP with MQMF_LAST_SEGMENT (yani, döndürülen ileti, gruptaki son mantıksal iletinin son bölümüdür).

Böyle bir ileti döndürüldüğünde, ileti grubu sonlandırılır ve MQGET çağrısı başarıyla tamamlandığında artık geçerli bir grup olmaz. Benzer bir şekilde, MQMF_SEGMENT işaretine sahip bir ileti MQGET çağrısı tarafından döndürüldüğünde mantıksal bir ileti geçerli mantıksal ileti olur. MQMF_LAST_SEGMENT işaretine sahip ileti döndürüldüğünde mantıksal ileti sonlandırılır.

Seçim ölçütü belirtilmezse, ardışık MQGET çağrılarını, kuyruktaki ilk ileti grubuna ilişkin iletileri doğru sırayla döndürür. Daha sonra, ikinci ileti grubuna ilişkin iletileri döndürür ve başka ileti kalmıncaya kadar devam eder. MatchOptions alanında aşağıdaki seçeneklerden birini ya da birkaçını belirterek, döndürülen ileti grupları seçilebilir:

- MQMO_MATCH_MSG_ID
- MQMO_MATCH_CORREL_ID
- MQMO_MATCH_GROUP_ID

Ancak, bu seçenekler yalnızca yürürlükteki ileti grubu ya da mantıksal ileti olmadığında geçerlidir. Daha fazla ayrıntı için “MQGMO-Get-message seçenekleri” sayfa 368 içinde açıklanan MatchOptions alanına bakın.

Çizelge 495 sayfa 393 içinde, kuyruk yöneticisinin MQGET çağrısında döndürülecek bir ileti bulmaya çalışırken aradığı MsgId, CorrelId, GroupId, MsgSeqNumber ve Offset alanlarının değerleri gösterilmektedir. Kurallar, kuyruktaki iletilerin kaldırılmasına ve kuyruktaki iletilere göz atılmasına uygulanır. Tabloda, Evet ya da Hayır anlamına gelir:

LOG ORD

Çağrıda MQGMO_LOGICAL_ORDER seçeneğinin belirtilip belirtilmediğini belirtir.

Cur grp

Aramadan önce geçerli bir ileti grubunun var olup olmadığını belirtir.

Cur log msg

Çağrıdan önce geçerli bir mantıksal iletinin bulunup bulunmadığını belirtir.

Diğer sütunlar

Kuyruk yöneticisinin aradığı değerleri gösterin. Önceki, kuyruk tanıtıcısı için önceki iletide alan için döndürülen değeri belirtir.

Çizelge 495. Mantıksal ileti grupları ve bölümlerindeki iletilerle ilgili MQGET seçenekleri							
Belirled iğiniz seçene- kler	Çağrıdan önce grup ve günlük- ileti durumu		Kuyruk yöneticisinin aradığı değerler				
	LOG ORD	Cur grp	Cur log msg	MsgId	CorrelId	GroupId	MsgSeqNumbe r
Evet	Hayır	Hayır	MatchOption s tarafından denetlenir	MatchOption s tarafından denetlenir	MatchOption s tarafından denetlenir	1	0
Evet	Hayır	Evet	Herhangi bir ileti tanıtıcısı	Herhangi bir ilinti tanıtıcısı	Önceki grup tanıtıcısı	1	Önceki görel konum + önceki kesim uzunluğu
Evet	Evet	Hayır	Herhangi bir ileti tanıtıcısı	Herhangi bir ilinti tanıtıcısı	Önceki grup tanıtıcısı	Önceki sıra numarası + 1	0
Evet	Evet	Evet	Herhangi bir ileti tanıtıcısı	Herhangi bir ilinti tanıtıcısı	Önceki grup tanıtıcısı	Önceki sıra numarası	Önceki görel konum + önceki kesim uzunluğu
Hayır	Herhan gi biri	Herhan gi biri	MatchOption s tarafından denetlenir	MatchOption s tarafından denetlenir	MatchOption s tarafından denetlenir	MatchOption s tarafından denetlenir	MatchOption s tarafından denetlenir

Kuyrukta birden çok ileti grubu varsa ve döndürülmeye uygun ise, gruplar, her gruptaki ilk mantıksal iletinin ilk bölümünün kuyruğundaki konumuyla belirlenen sırayla döndürülür. Yani, ileti sıra numaraları 1 ve görel konumlar 0 olan fiziksel iletiler, uygun grupların döndürüleceği sırayı belirler.

MQGMO_LOGICAL_ORDER seçeneği, iş birimlerini aşağıdaki gibi etkiler:

- Bir gruptaki ilk mantıksal ileti ya da bölüm bir iş birimi içinde alındıysa, aynı kuyruk tanıtıcısı kullanılırsa, gruptaki diğer tüm mantıksal iletiler ve kesimler bir iş birimi içinde alınmalıdır. Ancak, aynı iş birimi içinde alınmaları gerekmez. Bu, birçok fiziksel iletiden oluşan bir ileti grubunun, kuyruk tanıtıcısı için iki ya da daha fazla ardışık iş birimine bölünmesini sağlar.

- Bir gruptaki ilk mantıksal ileti ya da bölüm bir iş birimi içinde alınmazsa ve aynı kuyruk tanıtıcısı kullanılırsa, gruptaki diğer mantıksal iletilerin ve bölümlerin hiçbiri bir iş birimi içinde alınmaz.

Bu koşullar karşılanmazsa, MQGET çağrısı MQRC_INCONSISTENT_UOWneden koduyla başarısız olur.

MQGMO_LOGICAL_ORDER belirtildiğinde, MQGET çağrısında belirtilen MQGMO MQGMO_VERSION_2değerinden küçük olmamalıdır ve MQMD değeri MQMD_VERSION_2değerinden küçük olmamalıdır. Bu koşul karşılanmazsa, arama MQRC_WRONG_GMO_VERSION ya da MQRC_WRONG_MD_VERSIONneden koduyla başarısız olur.

Kuyruk tanıtıcısı için ardışık MQGET çağrıları için MQGMO_LOGICAL_ORDER belirtilmezse, iletiler ileti gruplarına ait olup olmadıklarına ya da mantıksal ileti kesimleri olup olmadıklarına bakılmaksızın döndürülür. Bu, belirli bir grup ya da mantıksal iletiden gelen iletilerin ya da bölümlerin sıra dışı olarak döndürülebileceği ya da diğer gruplardan ya da mantıksal iletilerden gelen iletilerle ya da gruplarda olmayan ve bölüm olmayan iletilerle karışabileceği anlamına gelir. In this situation, the particular messages that are returned by successive MQGET calls is controlled by the MQMO_* options specified on those calls (see the *MatchOptions* field described in “MQGMO-Get-message seçenekleri” sayfa 368 for details of these options).

Bu, bir sistem arızası oluştuğundan sonra ortada bir ileti grubunu ya da mantıksal iletiyi yeniden başlatmak için kullanılabilecek bir tekniktir. Sistem yeniden başlatıldığında, uygulama GroupId, MsgSeqNumber, Offsetve MatchOptions alanlarını uygun değerlere ayarlayabilir ve daha sonra, MQGET call with MQGMO_SYNCPOINT ya da MQGMO_NO_SYNCPOINT kümesini MQGMO_LOGICAL_ORDERbelirtmeden çağrısında bulunabilir. Bu çağrı başarılı olursa, kuyruk yöneticisi grup ve bölüm bilgilerini saklar ve bu kuyruk tanıtıcısını kullanan sonraki MQGET çağrıları MQGMO_LOGICAL_ORDER değerini normal olarak belirtebilir.

Kuyruk yöneticisinin MQGET çağrısı için sakladığı grup ve bölüm bilgileri, MQPUT çağrısı için sakladığı grup ve bölüm bilgilerinden farklıdır. Ayrıca, kuyruk yöneticisi aşağıdakilere ilişkin ayrı bilgileri de saklar:

- Kuyruktan ileti kaldran MQGET ıaşrmalar.
- MQGET , kuyruktaki iletilere göz atmanızı sağlar.

Belirli bir kuyruk tanıtıcısı için uygulama, MQGMO_LOGICAL_ORDER değerini belirten MQGET çağrılarını MQGET olmayan çağrılarla karıştırabilir. Ancak, aşağıdaki noktalara dikkat edin:

- MQGMO_LOGICAL_ORDERdeğerini atlarsanız, her başarılı MQGET çağrısı, kuyruk yöneticisinin kaydedilen grup ve bölüm bilgilerini, döndürülen iletiye karşılık gelen değerlere ayarlamasına neden olur; bu, kuyruk tanıtıcısı için kuyruk yöneticisi tarafından tutulan var olan grup ve bölüm bilgilerinin yerine geçer. Yalnızca çağrı işlemine uygun bilgiler (göz at ya da kaldır) değiştirilir.
- MQGMO_LOGICAL_ORDERdeğerini atlarsanız, geçerli bir ileti grubu ya da mantıksal ileti varsa arama başarısız olmaz; arama MQCC_WARNING tamamlama koduyla başarılı olabilir. Çizelge 496 sayfa 394 , ortaya çıkabilecek çeşitli durumları gösterir. Bu durumlarda, tamamlama kodu MQCC_OKdeğilse, neden kodu aşağıdakilerden biridir (uygun olduğu şekilde):

- MQRC_INCOMPLETE_GROUP
- MQRC_INCOMPLETE_MSG
- MQRC_INCONSISTENT_UOW

Not: Kuyruk yöneticisi, bir kuyruğa göz atarken ya da giriş için açılmış olan bir kuyruğu kapatırken grup ve bölüm bilgilerini denetlemez; bu durumlarda, tamamlama kodu her zaman MQCC_OK olur (başka hata olmadığı varsayılarak).

Çizelge 496. MQGET ya da MQCLOSE çağrısı grup ve bölüm bilgileriyle tutarlı olmadığına sonuç		
Geçerli arama:	Önceki arama MQGMO_LOGICAL_ORDER ile MQGET idi	Önceki arama MQGET onuz MQGMO_LOGICAL_ORDER idi
MQGET şu ürünü geçir MQGMO_LOGICAL_ORDER	MQCC_FAILED	MQCC_FAILED

Çizelge 496. MQGET ya da MQCLOSE çağrısı grup ve bölüm bilgileriyle tutarlı olmadığına sonuç (devamı var)

Geçerli arama:	Önceki arama MQGMO_LOGICAL_ORDER ile MQGET idi	Önceki arama MQGET onsuz MQGMO_LOGICAL_ORDER idi
MQGET olmadan MQGMO_LOGICAL_ORDER	MQCC_WARNING	MQCC_OK
Sonlandırılmamış grup ya da mantıksal ileti ile MQCLOSE	MQCC_WARNING	MQCC_OK

İletileri ve bölümleri mantıksal sırayla almak isteyen uygulamaların MQGMO_LOGICAL_ORDER belirtmesi önerilir; bu, en basit seçenektir. Kuyruk yöneticisi bu bilgileri yönettiği için, bu seçenek grup ve bölüm bilgilerini yönetme gereksinimini ortadan kaldırır. Ancak, özelleştirilmiş uygulamaların MQGMO_LOGICAL_ORDER seçeneği tarafından sağlanandan daha fazla denetime gereksinimi olabilir ve bu, bu seçenek belirtilmeden gerçekleştirilebilir. Daha sonra uygulama, MQMD'deki MsgId, CorrelId, GroupId, MsgSeqNumber ve Offset alanlarının ve MQGMO'daki MatchOptions içindeki MQMO_* seçeneklerinin her MQGET çağrısından önce doğru olarak ayarlandığından emin olmalıdır.

Örneğin, aldığı fiziksel iletileri iletmek isteyen bir uygulama, bu iletilerin gruplar halinde mi, yoksa mantıksal iletilerin bölümlerinde mi olduğuna bakılmaksızın MQGMO_LOGICAL_ORDER belirtmemelidir. Gönderme ve alma kuyruk yöneticileri arasında birden çok yolu olan karmaşık bir ağda, fiziksel iletiler sıradışı gelebilir. MQPUT çağrısında MQGMO_LOGICAL_ORDER ya da ilgili MQPMO_LOGICAL_ORDER belirtimini belirtmeyerek, ileme uygulaması her fiziksel iletiyi gelir gelmez, mantıksal olarak bir sonrakinin gelmesini beklemek zorunda kalmadan alabilir ve iletebilir.

MQGMO_LOGICAL_ORDER seçeneğini diğer MQGMO_* seçenekleriyle ve uygun koşullarda MQMO_* seçenekleriyle birlikte belirtebilirsiniz (önceki bölüme bakın).

- **z/OS** z/OS işletim sisteminde bu seçenek özel ve paylaşılan kuyruklar için desteklenir, ancak kuyruğun izin tipi MQIT_GROUP_ID olmalıdır. Paylaşılan kuyruklar için, kuyruğun eşlediği CFSTRUCT nesnesinin CFLEVEL (3) ya da üstü olması gerekir.
- Bu seçenek, aşağıdaki altyapılara ilişkin tüm yerel kuyruklar için desteklenir:

- **AIX** AIX
- **Linux** Linux
- **IBM i** IBM i
- **Windows** Windows

ve bu sistemlere bağlı IBM MQ MQI clients için,

MQGMO_COMPLETE_MSG

MQGET çağrısıyla yalnızca tam bir mantıksal ileti döndürülebilir. Mantıksal ileti bölümlenmişse, kuyruk yöneticisi kesimleri yeniden birleştirir ve uygulamaya tüm mantıksal iletiyi döndürür; mantıksal iletinin bölümlenmiş olması, bunu alan uygulama tarafından anlaşılır değildir.

Not: Bu, kuyruk yöneticisinin ileti bölümlerini yeniden birleştirmesine neden olan tek seçenektir. Belirtilmezse, kuyrukta varsa (ve MQGET çağrısında belirtilen diğer seçim ölçütlerine uyuyorsa), bölümler uygulamaya tek tek döndürülür. Tek tek kesimleri almak istemeyen uygulamalar her zaman MQGMO_COMPLETE_MSG belirtmelidir.

Bu seçeneği kullanmak için, uygulamanın tüm iletiyi alacak büyüklükte bir arabellek sağlaması ya da MQGMO_ACCEPT_TRUNCATED_MSG seçeneğini belirtmesi gerekir.

Kuyruk, bazı kesimleri eksik (belki de ağda gecikmiş oldukları ve henüz gelmedikleri için) bölümlenmiş iletiler içeriyorsa, MQGMO_COMPLETE_MSG belirtilmesi, eksik mantıksal iletilere ait bölümlerin alınmasını önler. Ancak, bu ileti bölümleri **CurrentQDepth** kuyruk özniteliğinin değerine katkıda

bulunmaya devam eder; bu, *CurrentQDepth* sıfırdan büyük olsa da, alınabilir mantıksal ileti olmayabileceği anlamına gelir.

Kalıcı iletiler için, kuyruk yöneticisi bölümleri yalnızca bir iş birimi içinde yeniden birleştirebilir:

- MQGET çağrısı kullanıcı tanımlı bir iş birimi içinde çalışıyorsa, o iş birimi kullanılır. Yeniden çevirme işlemi sırasında çağrı başarısız olursa, kuyruk yöneticisi, yeniden çevirme sırasında kaldırılan bölümleri kuyrukta yeniden gösterir. Ancak, başarısızlık iş biriminin başarıyla kesinleştirilmesini engellemez.
- Çağrı kullanıcı tanımlı bir iş birimi dışında çalışıyorsa ve kullanıcı tanımlı bir iş birimi yoksa, kuyruk yöneticisi çağrı süresi boyunca bir iş birimi yaratır. Çağrı başarılı olursa, kuyruk yöneticisi iş birimini otomatik olarak kesinleştirir (uygulamanın bunu yapması gerekmez). Çağrı başarısız olursa, kuyruk yöneticisi iş birimini geri alır.
- Çağrı kullanıcı tanımlı bir iş biriminin dışında çalışıyorsa, ancak kullanıcı tanımlı bir iş birimi varsa, kuyruk yöneticisi yeniden birleştirilemez. İleti yeniden çevirme gerektirmiyorsa, arama yine de başarılı olabilir. Ancak, ileti yeniden çevirme gerektiriyorsa, arama MQRC_UOW_NOT_AVAILABLEneden koduyla başarısız olur.

Kalıcı olmayan iletiler için kuyruk yöneticisi, yeniden çevirme gerçekleştirmek için bir iş biriminin kullanılabilir olmasını gerektirmez.

Bölüm olan her fiziksel iletinin kendi ileti tanımlayıcısı vardır. Tek bir mantıksal iletiyi oluşturan kesimler için, ileti tanımlayıcıdaki alanların çoğu mantıksal iletideki tüm kesimler için aynıdır; genellikle mantıksal iletideki kesimler arasında farklılık gösteren yalnızca *MsgId*, *Offset* ve *MsgFlags* alanlarıdır. Ancak, bir bölüm bir ara kuyruk yöneticisindeki bir teslim edilmeyen iletiler kuyruğuna yerleştirilirse, DLQ işleyicisi MQGMO_CONVERT seçeneğini belirten iletiyi alır ve bu, kesimin karakter kümesinin ya da kodlamasının değiştirilmesiyle sonuçlanabilir. DLQ işleyicisi kesimi başarılı bir şekilde gönderirse, kesim hedef kuyruk yöneticisine geldiğinde mantıksal iletideki diğer bölümlerden farklı bir karakter kümesi ya da kodlama olabilir.

CodedCharSetId ve *Encoding* alanlarının farklı olduğu bölümlerden oluşan mantıksal ileti, kuyruk yöneticisi tarafından tek bir mantıksal iletide yeniden birleştirilemez. Kuyruk yöneticisi, aynı karakter kümesi tanıtıcılarına ve kodlamalarına sahip mantıksal iletinin başında ilk birkaç ardışık bölümü yeniden birleştirir ve döndürür; MQGET çağrısı, tamamlanma kodu MQCC_WARNING ve neden kodu MQRC_INCONSISTENT_CCIDS ya da MQRC_INCONSISTENT_ENCODINGSile tamamlanır. Bu, MQGMO_CONVERT belirtilip belirtilmediğine bakılmaksızın gerçekleşir. Geri kalan bölümleri almak için uygulamanın MQGET çağrısı MQGMO_COMPLETE_MSG seçeneği olmadan yeniden yayınlayarak bölümleri birer birer alması gerekir. MQGMO_LOGICAL_ORDER , kalan bölümleri sırayla almak için kullanılabilir.


Bölümleri yerleştiren bir uygulama, ileti tanımlayıcısındaki diğer alanları bölümler arasında farklılık gösteren değerlere de ayarlayabilir. Ancak, alan uygulama mantıksal iletiyi almak için MQGMO_COMPLETE_MSG kullanıyorsa, bunu yapmanın bir yararı yoktur. Kuyruk yöneticisi bir mantıksal iletiyi yeniden birleştirdiğinde, ileti tanımlayıcısında ilk kesime ilişkin ileti tanımlayıcısının değerlerini döndürür; tek kural dışı durum, kuyruk yöneticisinin yeniden birleştirilen iletinin tek bölüm olduğunu belirtmek için ayarladığı *MsgFlags* alanıdır.

Bir rapor iletisi için MQGMO_COMPLETE_MSG belirtilirse, kuyruk yöneticisi özel işleme gerçekleştirir. Kuyruk yöneticisi, mantıksal iletideki farklı bölümlerle ilgili rapor tipindeki tüm rapor iletilerinin kuyrukta olup olmadığını kontrol eder. Varsa, bunlar MQGMO_COMPLETE_MSG belirtilerek tek bir ileti olarak alınabilir. Bunun mümkün olması için, rapor iletileri bölümlenmeyi destekleyen bir kuyruk yöneticisi ya da MCA tarafından oluşturulmalıdır ya da kaynak uygulama en az 100 baytlık ileti verisi istemelidir (yani, uygun MQRO_*_WITH_DATA ya da MQRO_*_WITH_FULL_DATA seçenekleri belirtilmelidir). Bir kesim için uygulama verilerinin tam miktarından az bir miktar varsa, eksik baytlar, döndürülen rapor iletisinde boş değerlerle değiştirilir.

MQGMO_COMPLETE_MSG MQGMO_MSG_UNDER_CURSOR ya da MQGMO_BROWSE_MSG_UNDER_CURSOR ile belirtilirse, göz atma imleci MQMD ' deki *Offset* alanı 0 değerine sahip bir iletide konumlandırılmalıdır. Bu koşul karşılanmazsa, arama MQRC_INVALID_MSG_UNDER_CURSORneden koduyla başarısız olur.

MQGMO_COMPLETE_MSG , belirtilmesi gerekmeyen MQGMO_ALL_SEGMENTS_AVAILABLEögesini belirtir.

MQGMO_COMPLETE_MSG , MQGMO_SYNCPOINT_IF_PERSISTENT dışındaki diğer MQGMO_* seçenekleriyle ve MQMO_MATCH_OFFSET dışındaki MQMO_* seçeneklerinden herhangi biriyle belirtilebilir.

-  z/OS işletim sisteminde bu seçenek özel ve paylaşılan kuyruklar için desteklenir, ancak kuyruğun izin tipi MQIT_GROUP_ID olmalıdır. Paylaşılan kuyruklar için, kuyruk eşlemindeki CFSTRUCT nesnesinin CFLEVEL (3) ya da üstü olması gerekir.

• Aşağıdaki platformlarda:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için bu seçenek tüm yerel kuyruklar için desteklenir.

MQGMO_ALL_MSGS_VAR

Bir gruptaki iletiler yalnızca gruptaki tüm iletiler kullanılabilir olduğunda alınabilir. Kuyruk, bazı iletilerin eksik olduğu ileti grupları içeriyorsa (ağda gecikmiş oldukları ve henüz gelmedikleri için olabilir), MQGMO_ALL_MSGS_AVAILABLE belirtilmesi tamamlanmamış gruplara ait iletilerin alınmasını önler. Ancak, bu iletiler **CurrentQDepth** kuyruk özniteliğinin değerine katkıda bulunmaya devam eder; bu, CurrentQDepth sıfırdan büyük olsa da, alınabilecek ileti grupları olmayabileceği anlamına gelir. Alınabilecek başka ileti yoksa, belirtilen bekleme aralığı (varsa) sona erdikten sonra MQRC_NO_MSG_AVAILABLE neden kodu döndürülür.

MQGMO_ALL_MSGS_AVAILABLE işleminin işlenmesi, MQGMO_LOGICAL_ORDER ' un da belirtilip belirtilmediğine bağlıdır:


- Her iki seçenek de belirtilirse, MQGMO_ALL_MSGS_AVAILABLE yalnızca geçerli bir grup ya da mantıksal ileti olmadığında etkili olur. Geçerli bir grup ya da mantıksal ileti varsa, MQGMO_ALL_MSGS_AVAILABLE yoksayılır. Bu, iletiler mantıksal sırada işlenirken MQGMO_ALL_MSGS_AVAILABLE ' in açık kalabileceği anlamına gelir.
- MQGMO_ALL_MSGS_AVAILABLE , MQGMO_LOGICAL_ORDER olmadan belirtilirse, MQGMO_ALL_MSGS_AVAILABLE her zaman bir etkiye sahiptir. Bu, gruptaki kalan iletileri kaldırabilmek için, gruptaki ilk ileti kuyruktan kaldırıldıktan sonra seçeneğin kapatılması gerektiği anlamına gelir.

MQGET Belirten MQGMO_ALL_MSGS_AVAILABLE çağrısının başarıyla tamamlanması, MQGET çağrısının yayınlandığı sırada gruptaki tüm iletilerin kuyruksa olduğu anlamına gelir. Ancak, diğer uygulamaların yine de gruptan ileti kaldırabileceğini unutmayın (grup, gruptaki ilk iletiyi alan uygulamaya kilitlemez).



Bu seçeneği atlarsanız, grup eksik olduğunda bile gruplara ait iletiler alınabilir.

MQGMO_ALL_MSGS_AVAILABLE , belirtilmesi gerekmeyen MQGMO_ALL_SEGMENTS_AVAILABLE ögesini belirtir.

MQGMO_ALL_MSGS_AVAILABLE , diğer MQGMO_* seçeneklerinden herhangi biriyle ve MQMO_* seçeneklerinden herhangi biriyle belirtilebilir.

-  z/OS işletim sisteminde bu seçenek özel ve paylaşılan kuyruklar için desteklenir, ancak kuyruğun izin tipi MQIT_GROUP_ID olmalıdır. Paylaşılan kuyruklar için, kuyruk eşlemindeki CFSTRUCT nesnesinin CFLEVEL (3) ya da üstü olması gerekir.

• Aşağıdaki platformlarda:

-  AIX
-  IBM i

-  Linux
-  Windows

ve bu sistemlere bağı IBM MQ MQI clients için bu seçenek tüm yerel kuyruklar için desteklenir.

MQGMO_ALL_SEGMENTS_VAR

Mantıksal iletideki bölümler, yalnızca mantıksal iletideki tüm bölümler kullanılabilir olduğunda alınmak üzere kullanılabilir. Kuyruk, eksik bölümlerin bazılarını içeren bölümlenmiş iletiler içeriyorsa (ağda geciktikleri ve henüz gelmedikleri için olabilir), MQGMO_ALL_SEGMENTS_AVAILABLE değerinin belirlenmesi, tamamlanmamış mantıksal iletilere ait bölümlerin alınmasını önler. Ancak, bu kesimler **CurrentQDepth** kuyruk özniteliğinin değerine katkıda bulunmaya devam eder; bu, CurrentQDepth sıfırdan büyük olsa da, alınabilir mantıksal ileti olmayabileceği anlamına gelir. Alınabilecek başka ileti yoksa, belirtilen bekleme aralığı (varsa) sona erdikten sonra MQRC_NO_MSG_AVAILABLE neden kodu döndürülür.

MQGMO_ALL_SEGMENTS_AVAILABLE işleminin işlenmesi, MQGMO_LOGICAL_ORDER ' un da belirtilip belirtilmediğine bağlıdır:

- Her iki seçenek de belirtilirse, MQGMO_ALL_SEGMENTS_AVAILABLE yalnızca geçerli bir mantıksal ileti olmadığında etkili olur. Geçerli bir mantıksal ileti varsa, MQGMO_ALL_SEGMENTS_AVAILABLE yoksayılır. Bu, iletiler mantıksal sırada işlenirken MQGMO_ALL_SEGMENTS_AVAILABLE ' in açık kalabileceği anlamına gelir.
- MQGMO_ALL_SEGMENTS_AVAILABLE , MQGMO_LOGICAL_ORDER olmadan belirtilirse, MQGMO_ALL_SEGMENTS_AVAILABLE her zaman bir etkiye sahiptir. Bu, mantıksal iletideki kalan bölümleri kaldırabilmek için mantıksal iletideki ilk bölüm kaldırıldıktan sonra seçeneğin kapatılması gerektiği anlamına gelir.

Bu seçenek belirtilmezse, mantıksal ileti eksik olduğunda bile ileti bölümleri alınabilir.

Hem MQGMO_COMPLETE_MSG hem de MQGMO_ALL_SEGMENTS_AVAILABLE , herhangi birinin alınabilmesi için önce tüm bölümlerin kullanılabilir olmasını gerektiriyorken, ilki tüm iletiyi döndürür, ikincisi ise bölümlerin birer birer alınmasına izin verir.

Bir rapor iletisi için MQGMO_ALL_SEGMENTS_AVAILABLE belirtilirse, kuyruk yöneticisi, tüm mantıksal iletiyi oluşturan her bir bölüm için en az bir rapor iletisi olup olmadığını görmek üzere kuyruğu denetler. Varsa, MQGMO_ALL_SEGMENTS_AVAILABLE koşulu karşılanır. Ancak, kuyruk yöneticisi mevcut rapor iletilerinin *tipini* denetlemez ve bu nedenle, mantıksal iletinin bölümleriyle ilgili rapor iletilerinde rapor tiplerinin bir karışımı olabilir. Sonuç olarak, MQGMO_ALL_SEGMENTS_AVAILABLE 'in başarısı, MQGMO_COMPLETE_MSG ' un başarılı olduğu anlamına gelmez. Belirli bir mantıksal iletinin bölümleri için var olan rapor tiplerinin bir karışımı varsa, bu rapor iletileri birer birer alınmalıdır.

MQGMO_ALL_SEGMENTS_AVAILABLE seçeneğini diğer MQGMO_* seçeneklerinden herhangi biriyle ve MQMO_* seçeneklerinden herhangi biriyle belirtebilirsiniz.

- z/OS işletim sisteminde bu seçenek özel ve paylaşılan kuyruklar için desteklenir, ancak kuyruğun izin tipi MQIT_GROUP_ID olmalıdır. Paylaşılan kuyruklar için, kuyruk eşlemindeki CFSTRUCT nesnesinin CFLEVEL (3) ya da üstü olması gerekir.
- Aşağıdaki platformlarda:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağı IBM MQ MQI clients için bu seçenek tüm yerel kuyruklar için desteklenir.

Özellik seçenekleri

Aşağıdaki seçenekler iletinin özellikleriyle ilgilidir:

MQGMO_PROPERTIES_AS_Q_DEF

İleti tanımlayıcısında (ya da uzantıda) bulunanlar dışında, iletinin özellikleri **PropertyControl** kuyruk özneliğiyle tanımlandığı şekilde gösterilmelidir. `MsgHandle` belirtiliyorsa, bu seçenek yoksayılır ve **PropertyControl** kuyruk özneliğinin değeri `MQPROP_FORCE_MQRFH2` değilse, iletinin özellikleri `MsgHandle` aracılığıyla kullanılabilir.

Herhangi bir özellik seçeneği belirtilmezse, bu işlem varsayılan işlemdir.

MQGMO_PROPERTIES_IN_HANDLE

İletinin özellikleri `MsgHandle` aracılığıyla kullanılabilir kılınmalıdır. Herhangi bir ileti tanıtıcısı sağlanmazsa, arama `MQRC_HMSG_ERROR` nedeniyle başarısız olur.

Not: İleti daha sonra ileti tanıtıcısı yaratmayan bir uygulama tarafından okunursa, kuyruk yöneticisi ileti özelliklerini `MQRFH2` yapısına yerleştirir. Beklenmeyen bir `MQRFH2` üstbilgisinin varlığının, var olan bir uygulamanın davranışını kesintiye uğratacağını öğrenebilirsiniz.

MQGMO_NO_PROPERTIES

İleti tanımlayıcıda (ya da uzantıda) bulunanlar dışında, iletinin özellikleri alınmaz. Bir `MsgHandle` sağlanırsa yoksayılır.

MQGMO_PROPERTIES_FORCE_MQRFH2

İleti tanımlayıcısında (ya da uzantıda) bulunanlar dışında, iletinin özellikleri `MQRFH2` üstbilgileri kullanılarak gösterilmelidir. Bu, özellikleri almayı bekleyen, ancak ileti tanıtıcılarını kullanacak şekilde değiştirilemeyen uygulamalara ilişkin önceki sürümle uyumluluk sağlar. Bir `MsgHandle` sağlanırsa yoksayılır.

MQGMO_PROPERTIES_COMPATIBILITY

İleti `"mcd."`, `"jms."`, `"usr."` ya da `"mqext."` önekine sahip bir özellik içeriyorsa, tüm ileti özellikleri bir `MQRFH2` üstbilgisinde uygulamaya teslim edilir. Ters durumda, ileti tanımlayıcısında (ya da uzantıda) bulunanlar dışında, iletinin tüm özellikleri atılır ve artık uygulama bunlara erişemez.

Varsayılan seçenek

Açıklanan seçeneklerin hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

MQGMO_NONE

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar. `MQGMO_NONE` program belgelerine yardımcı olur; bu seçeneğin başka bir değerle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Options alanının ilk değeri `MQGMO_NO_WAIT` artı `MQGMO_PROPERTIES_AS_Q_DEF` değeridir.

MQGMO için WaitInterval (MQLONG)

Bu, `MQGET` çağrısının uygun bir iletinin gelmesini (yani, `MQGET` çağrısının **MsgDesc** değiştirgesinde belirtilen seçim ölçütlerini karşılayan bir ileti) bekleyeceği yaklaşık süreyi milisaniye cinsinden ifade eder.

Önemli: Uygun bir ileti hemen varsa, bekleme ya da gecikme yoktur.

Daha fazla ayrıntı için "`MQMD`-İleti tanımlayıcı" sayfa 420 içinde açıklanan `MsgId` alanına bakın). Bu süre geçtikten sonra uygun bir ileti gelmezse, çağrı `MQCC_FAILED` ve neden kodu `MQRC_NO_MSG_AVAILABLE` ile tamamlanır.

`z/OS` sistemlerinde, `MQGET` çağrısının gerçekte beklediği süre, sistem yüklemesi ve iş zamanlamasıyla ilgili önemli noktaların etkilenir ve `WaitInterval` için belirtilen değer ile `WaitInterval` değerinden yaklaşık 100 milisaniye daha büyük olabilir.

WaitInterval , MQGMO_WAIT ya da MQGMO_SET_SIGNAL seçeneğiyle birlikte kullanılır. Bunların hiçbiri belirtilmezse yoksayılır. Bunlardan biri belirtilirse, *WaitInterval* sıfırdan büyük ya da sıfıra eşit ya da aşağıdaki özel değer olmalıdır:

MQWI_SINIRSIZ

Sınırsız bekleme aralığı.

Bu alanın ilk değeri şudur: 0.

Signal1 (MQGMO için MQLONG)

Bu, yalnızca MQGMO_SET_SIGNAL seçeneğiyle birlikte kullanılan bir giriş alanıdır; bir ileti kullanılabilir olduğunda teslim edilecek bir sinyali tanımlar.

Not: Bu alanın veri tipi ve kullanımı ortam tarafından belirlenir; bu nedenle, farklı ortamlar arasında bağlantı oluşturmak istediğiniz uygulamalar sinyal kullanmamalıdır.

- z/OS üzerinde, bu alan bir ECB 'nin (Event Control Block; Olay Denetim Öbeği) adresini içermelidir. MQGET çağrısı yayınlanmadan önce ECB uygulama tarafından temizlenmelidir. ECB 'yi içeren saklama alanı, kuyruk kapatılıncaya kadar serbest bırakılmamalıdır. ECB, açıklanan sinyal tamamlama kodlarından biriyle kuyruk yöneticisi tarafından gönderilir. Bu tamamlama kodları, z/OS eşleme makrosu IHAECB 'de bir kullanıcı tamamlama kodu için tanımlanan alan olan ECB' nin bit 2-31 arasında ayarlanır.
- Diğer tüm ortamlarda bu ayrılmış bir alandır; değeri önemli değildir.

Sinyal tamamlama kodları şunlardır:

MQEC_MSG_TESLIM ALINDI

Kuyruğa uygun bir ileti geldi. Bu ileti çağırıcı için ayrıldı; ikinci bir MQGET isteği yayınlanmalıdır, ancak ikinci istek yapılmadan önce başka bir uygulama iletiyi alabilir.

MQEC_WAIT_INTERVAL_SÜRESİ doldu

Belirtilen *WaitInterval* süresi, uygun bir ileti gelmeden sona erdi.

MQEC_WAIT İPTAL edildi

Bekleme, belirsiz bir nedenle (kuyruk yöneticisinin sonlandırılması ya da geçersiz kılınması gibi) iptal edildi. Daha fazla tanımlama istiyorsanız isteği yeniden yayınlayın.

MQEC_Q_MGR QUIESCING

Kuyruk yöneticisi susturma durumuna girdiği için bekleme iptal edildi (MQGET çağrısında MQGMO_FAIL_IF QUIESCING belirtildi).

MQEC_CONNECTION QUIESCING

Bağlantı susturma durumuna girdiği için bekleme iptal edildi (MQGET çağrısında MQGMO_FAIL_IF QUIESCING belirtildi).

Bu alanın ilk değeri ortam tarafından belirlenir:

- z/OS üzerinde, ilk değer boş değerli işaretçidir.
- Diğer tüm ortamlarda, ilk değer 0 olur.

MQGMO için Signal2 (MQLONG)

Bu, yalnızca MQGMO_SET_SIGNAL seçeneğiyle birlikte kullanılan bir giriş alanıdır. Ayrılmış bir alandır; değeri önemli değildir.

Bu alanın ilk değeri şudur: 0.

MQGMO için ResolvedQName (MQCHAR48)

Bu, kuyruk yöneticisinin, yerel kuyruk yöneticisine tanımlandığı şekilde, iletinin alındığı kuyruğun yerel adına ayarladığı bir çıkış alanıdır. Bu, aşağıdaki durumda kuyruğu açmak için kullanılan addan farklıdır:

- Bir diğer ad kuyruğu açıldı (bu durumda, çözümlenen diğer adın döndürüldüğü yerel kuyruğun adı) ya da
- Bir model kuyruğu açıldı (bu durumda, dinamik yerel kuyruğun adı döndürülür).

Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQGMO için MatchOptions (MQLONG)

Bu seçenekler, uygulamanın MQGET çağrısıyla döndürülen iletiyi seçmek için **MsgDesc** değiştirgesindeki hangi alanların kullanılacağını seçmesini sağlar. Uygulama, bu alanda gerekli seçenekleri ayarlar ve **MsgDesc** parametresindeki karşılık gelen alanları bu alanlar için gerekli olan değerlere ayarlar. Yalnızca, MQMD ' de iletiye ilişkin bu değerleri içeren iletiler, MQGET çağrısında o **MsgDesc** değiştirgesini kullanarak alma adaylarıdır. Karşılık gelen eşleşme seçeneğinin belirlenmediği alanlar, döndürülecek ileti seçilirken yoksayılr. MQGET çağrısında seçim ölçütü belirtmezseniz (*herhangi bir* iletisi kabul edilebilir), *MatchOptions* ögesini MQMO_NONE olarak ayarlayın.

- z/OSişletim sisteminde, kullanılabilir seçim ölçütleri, kuyruk için kullanılan dizin tipiyle sınırlanabilir. Daha fazla ayrıntı için **IndexType** kuyruk özniteliğine bakın.

MQGMO_LOGICAL_ORDER belirtirseniz, sonraki MQGET çağrısıyla yalnızca belirli iletiler döndürülmeye uygun olur:

- Geçerli bir grup ya da mantıksal ileti yoksa, yalnızca *MsgSeqNumber* buna eşit 1 ve *Offset* buna eşit 0 iletilere dönüş için uygun olur. Bu durumda, uygun iletilerden hangilerinin döndürüleceğini seçmek için aşağıdaki eşleşme seçeneklerinden birini ya da birkaçını kullanabilirsiniz:
 - MQMO_MATCH_MSG_ID
 - MQMO_MATCH_CORREL_ID
 - MQMO_MATCH_GROUP_ID
- Yürürlükteki bir grup ya da mantıksal ileti varsa, yalnızca mantıksal iletideki sonraki ileti ya da sonraki bölüm döndürülmeye uygundur ve bu, MQMO_* seçenekleri belirtilerek değiştirilemez.

Önceki her iki durumda da, geçerli olmayan eşleştirme seçeneklerini belirtebilirsiniz; ancak **MsgDesc** parametresindeki ilgili alanın değeri, döndürülecek iletideki ilgili alanın değeriyle eşleşmelidir; çağrı başarısız olur; neden kodu MQRC_MATCH_OPTIONS_ERROR ise bu koşul karşılanmaz.

MQGMO_MSG_UNDER_CURSOR ya da MQGMO_BROWSE_MSG_UNDER_CURSOR belirtirseniz *MatchOptions* yoksayılr.

İleti özelliğine dayalı olarak ileti alma, eşleşme seçenekleri kullanılarak gerçekleştirilmez; daha fazla bilgi için bkz. “MQOD için SelectionString (MQCHARV)” sayfa 489.

Aşağıdaki eşleşme seçeneklerinden birini ya da birkaçını belirleyebilirsiniz:

MQMO_MATCH_MSG_ID

Alınacak iletinin, MQGET çağrısının **MsgDesc** değiştirgesindeki *MsgId* alanının değeriyle eşleşen bir ileti tanıtıcısı olmalıdır. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, ilinti tanıtıcısı) ek olarak kullanılır.

Bu seçeneği atlarsanız, **MsgDesc** parametresindeki *MsgId* alanı yoksayılr ve herhangi bir ileti tanıtıcısı eşleşir.

Not: İleti tanıtıcısı MQMI_NONE, iletiye ilişkin MQMD ' deki herhangi bir ileti tanıtıcısıyla eşleşen özel bir değerdir. Bu nedenle, MQMO_MATCH_MSG_ID 'in MQMI_NONE ile belirtilmesi, MQMO_MATCH_MSG_ID' in belirtilmemesiyle aynıdır.

MQMO_MATCH_CORREL_ID

Alınacak iletinin, MQGET çağrısının **MsgDesc** değiştirgesindeki *CorrelId* alanının değeriyle eşleşen bir ilinti tanıtıcısı olmalıdır. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, ileti tanıtıcısı) ek olarak bulunur.

Bu seçeneği atlarsanız, **MsgDesc** parametresindeki *CorrelId* alanı yoksayılr ve herhangi bir ilinti tanıtıcısı eşleşir.

Not: MQCI_NONE ilinti tanıtıcısı, iletiye ilişkin MQMD ' deki *any* ilinti tanıtıcısıyla eşleşen özel bir değerdir. Bu nedenle, MQMO_MATCH_CORREL_ID 'in MQCI_NONE ile belirtilmesi, MQMO_MATCH_CORREL_ID' in belirtilmemesiyle aynıdır.

MQMO_MATCH_GROUP_ID

Alınacak iletinin, MQGET çağrısının **MsgDesc** değiştirgesindeki *GroupId* alanının değeriyle eşleşen bir grup tanıtıcısı olmalıdır. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, ilinti tanıtıcısı) ek olarak kullanılır.

Bu seçeneği atlarsanız, **MsgDesc** parametresindeki *GroupId* alanı yoksayılr ve herhangi bir grup tanıtıcısı eşleşir.

Not: Grup tanıtıcısı MQGI_NONE, iletiye ilişkin MQMD ' deki *herhangi bir grup tanıtıcısı*yla eşleşen özel bir değerdir. Bu nedenle, MQGI_NONE ile MQMO_MATCH_GROUP_ID belirtilmesi, MQMO_MATCH_GROUP_ID belirtilmemesiyle aynıdır.

MQMO_MATCH_MSG_SEQ_NUMBER

Alınacak iletinin, MQGET çağrısının **MsgDesc** değiştirgesindeki *MsgSeqNumber* alanının değeriyle eşleşen bir ileti sıra numarası olmalıdır. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, grup tanıtıcısı) ek olarak bulunur.

Bu seçeneği atlarsanız, **MsgDesc** parametresindeki *MsgSeqNumber* alanı yoksayılr ve herhangi bir ileti sıra numarası eşleşir.

MQMO_MATCH_OFFSET

Alınacak ileti, MQGET çağrısının **MsgDesc** değiştirgesindeki *Offset* alanının değeriyle eşleşen bir görelî konuma sahip olmalıdır. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, ileti sıra numarası) ek olarak bulunur.

Bu seçeneği atlarsanız, **MsgDesc** parametresindeki *Offset* alanı yoksayılr ve görelî konum eşleşir.

- Bu seçenek z/OSüzerinde desteklenmez.

MQMO_MATCH_MSG_TOKEN

Alınacak iletinin, MQGET çağrısında belirtilen MQGMO yapısındaki *MsgToken* alanının değeriyle eşleşen bir ileti simgesi olmalıdır.

Bu seçeneği tüm yerel kuyruklar için belirleyebilirsiniz. Bunu, *IndexType* MQIT_MSG_TOKEN (WLM tarafından yönetilen kuyruk) içeren bir kuyruk için belirtirseniz, MQMO_MATCH_MSG_TOKEN ile başka bir eşleşme seçeneği belirleyemez.

MQGMO_WAIT ya da MQGMO_SET_SIGNAL ile MQMO_MATCH_MSG_TOKEN belirtmezsiniz. Uygulama, *IndexType* MQIT_MSG_TOKEN içeren bir kuyruğa ileti gelmesini beklemek isterse, MQMO_NONE belirtin.

Bu seçeneği atlarsanız, MQGMO ' daki *MsgToken* alanı yoksayılr ve herhangi bir ileti simgesi eşleşir.

Açıklanan seçeneklerden hiçbirini belirtmezseniz, aşağıdaki seçeneği kullanabilirsiniz:

MQMO_NONE

Döndürülecek iletiyi seçerken eşleşme kullanılmaz; kuyruktaki tüm iletiler alınmaya uygundur (ancak, MQGMO_ALL_MSGS_KULLANILABILIR, MQGMO_ALL_SEGMENTS_KULLANILABILIR ve MQGMO_COMPLETE_MSG seçenekleri tarafından denetlenir).

MQMO_NONE, program belgelerine yardımcı olur. Bu seçeneğin başka bir MQMO_ * seçeneğiyle kullanılması amaçlanmamıştır, ancak değeri sıfır olduğundan bu tür bir kullanım algılanamaz.

Bu bir giriş alanıdır. Bu alanın ilk değeri MQMO_MATCH_MSG_ID ve MQMO_MATCH_CORREL_ID 'dir. *Version* , MQGMO_VERSION_2' den küçükse bu alan yoksayılr.

Not: *MatchOptions* alanının ilk değeri, daha önceki MQSeries kuyruk yöneticileriyle uyumluluk için tanımlanır. Ancak, seçim ölçütleri kullanılmadan bir kuyruktan ileti dizisi okunurken, bu ilk değer uygulamanın her MQGET çağrısından önce *MsgId* ve *CorrelId* alanlarını MQMI_NONE ve MQCI_NONE olarak sıfırlamasını gerektirir. *Version* ögesini MQGMO_VERSION_2ve *MatchOptions* ögesini MQMO_NONE olarak ayarlayarak *MsgId* ve *CorrelId* ögelerini sıfırlama gereksinmesini önleyin.

İlgili kavramlar

JMS ' de ileti seçiciler

MQGMO için GroupStatus (MQCHAR)

Bu işaret, alınan iletinin bir grupta olup olmadığını gösterir.

Aşağıdaki değerlerden birini içerir:

MQGS_NOT_IN_GROUP

İleti bir grupta değil.

MQGS_MSG_IN_GROUP

İleti bir grupta, ancak gruptaki son ileti değil.

MQGS_LAST_MSG_IN_GROUP

İleti, gruptaki son iletidir.

Bu değer, grup yalnızca bir iletiden oluşuyorsa da döndürülür.

Bu bir çıkış alanıdır. Bu alanın ilk değeri MQGS_NOT_IN_GROUP. *Version* , MQGMO_VERSION_2' den küçükse bu alan yoksayılır.

MQGMO için SegmentStatus (MQCHAR)

Bu, alınan iletinin bir mantıksal iletinin parçası olup olmadığını gösteren bir işarettir. Aşağıdaki değerlerden birini içerir:

MQSS_NOT_A_SEGMENT

İleti bir bölüm değil.

MQSS_SEGMENT

İleti bir bölüm, ancak mantıksal iletinin son bölümü değil.

MQSS_LAST_SEGMENT

İleti, mantıksal iletinin son bölümüdür.

Bu değer, mantıksal ileti tek bir bölümden oluşuyorsa da döndürülür.

z/OS üzerinde, kuyruk yöneticisi her zaman bu alanı MQSS_NOT_A_SEGMENT olarak ayarlar.

Bu bir çıkış alanıdır. Bu alanın ilk değeri: MQSS_NOT_A_SEGMENT. *Version* , MQGMO_VERSION_2' den küçükse bu alan yoksayılır.

MQGMO için bölümlenme (MQCHAR)

Bu, alınan ileti için daha fazla bölümlenmeye izin verilip verilmediğini gösteren bir işarettir. Aşağıdaki değerlerden birini içerir:

MQSEG_ENGELLEME

Bölümlenmeye izin verilmiyor.

MQSEG_ALLOWED

Bölümlenmeye izin verilir.

z/OS' da, kuyruk yöneticisi her zaman bu alanı MQSEG_INN olarak ayarlar.

Bu bir çıkış alanıdır. Bu alanın ilk değeri MQSEG_INDÜ. *Version* , MQGMO_VERSION_2' den küçükse bu alan yoksayılır.

MQGMO için Reserved1 (MQCHAR)

Bu ayrılmış bir alandır. Bu alanın ilk değeri boş bir karakterdir. *Version* , MQGMO_VERSION_2' den küçükse bu alan yoksayılır.

MQGMO için MsgToken (MQBYTE16)

MsgToken alanı-MQGMO yapısı. Bu alan, kuyruk yöneticisi tarafından bir iletiyi benzersiz olarak tanımlamak için kullanılır.

Bu, kuyruktaki bir iletiyi benzersiz olarak tanımlamak için kuyruk yöneticisi tarafından oluşturulan bir byte dizisidir. İleti simgesi, ileti kuyruk yöneticisine ilk kez yerleştirildiğinde oluşturulur ve kuyruk yöneticisi yeniden başlatılmadıkça, ileti kalıcı olarak kuyruk yöneticisinden kaldırılincaya kadar iletide kalır.

İleti kuyruktan kaldırıldığında, iletinin eşgörünümünü tanımlayan *MsgToken* artık geçerli değildir ve hiçbir zaman yeniden kullanılmaz. Kuyruk yöneticisi yeniden başlatılırsa, yeniden başlatmadan önce kuyrukte bir ileti saptayan *MsgToken* , yeniden başlatmadan sonra geçerli olmayabilir. Ancak *MsgToken* , farklı bir ileti örneğini tanımlamak için hiçbir zaman yeniden kullanılmaz. *MsgToken* , kuyruk yöneticisi tarafından oluşturulur ve herhangi bir dış uygulama tarafından görünmez.

Sürüm 3 ya da daha yüksek bir MQGMO 'nun sağlandığı MQGET çağrısıyla bir ileti döndürüldüğünde, kuyruk yöneticisi, kuyruktaki iletiyi tanıtan *MsgToken* MQGMO' da döndürülür. Bunun bir kural dışı durumu vardır: İleti, uyumlulaştırma noktasının dışındaki kuyruktan kaldırıldığında, kuyruk yöneticisi *MsgToken* döndürmeyebilir; sonraki bir MQGET çağrısında döndürülen iletiyi saptamak yararlı olmayabilir. Uygulamalar, sonraki MQGET çağrılarında iletiye gönderme yapmak için yalnızca *MsgToken* kullanmalıdır.

Bir *MsgToken* belirtilirse ve *MatchOption* MQMO_MATCH_MSG_TOKEN belirtilirse ve MQGMO_MSG_UNDER_CURSOR ya da MQGMO_BROWSE_MSG_UNDER_CURSOR belirtilmezse, yalnızca o *MsgToken* ile tanıtilan ileti döndürülebilir. Bu seçenek, INDXTYPE ' dan bağımsız olarak tüm yerel kuyruklarda geçerlidir ve z/OS üzerinde INDXTYPE (MSGTOKEN) yalnızca Workload Manager (WLM) kuyruklarında kullanılmalıdır.

Diğer *MatchOptions* belirtilirse ve eşleşmezlerse, MQRC_NO_MSG_AVAILABLE döndürülür. MQGMO_BROWSE_NEXT MQMO_MATCH_MSG_TOKEN ile kodlandıysa, *MsgToken* ile tanıtilan ileti, çağırana tanıtıcı için göz atma imlecinin ötesinde ise döndürülür.

MQGMO_MSG_UNDER_CURSOR ya da MQGMO_BROWSE_MSG_UNDER_CURSOR belirtilirse, MQMO_MATCH_MSG_TOKEN yoksayıdır.

MQMO_MATCH_MSG_TOKEN, aşağıdaki alma iletisi seçenekleriyle geçerli değil:

- MQGMO_WAIT
- MQGMO_SET_SIGNAL

MQMO_MATCH_MSG_TOKEN ile belirtilen bir MQGET çağrısı için, çağrıya sürüm 3 ya da üstü bir MQGMO sağlanmalıdır; tersi durumda, MQRC_ERROR _gmo_version döndürülür.

MsgToken şu anda geçerli değilse, başka bir hata yoksa, MQRC_NO_MSG_AVAILABLE içeren MQCC_FAILED döndürülür.

MQGMO için ReturnedLength (MQLONG)

Bu, kuyruk yöneticisinin **Buffer** değiştirgesindeki MQGET çağrısının döndürdüğü ileti verilerinin bayt cinsinden uzunluğunu ayarladığı bir çıkış alanıdır. Kuyruk yöneticisi bu yeteneği desteklemiyorsa, *ReturnedLength* değeri MQRL_UNDEFINED değerine ayarlanır.

İletiler kodlamalar ya da karakter kümeleri arasında dönüştürüldüğünde, ileti verileri bazen boyutu değiştirebilir. MQGET çağrısından dönüş:

- *ReturnedLength* MQRL_UNDEFINED değilse, döndürülen ileti verilerinin bayt sayısı *ReturnedLength* tarafından verilir.
- *ReturnedLength* ' in MQRL_UNDEFINED değeri varsa, döndürülen ileti verileri byte sayısı genellikle *BufferLength* ve *DataLength* değerinden küçük olanlar tarafından verilir, ancak MQGET çağrısı MQRC_TRUNCATED_MSG_ACCEPTED neden koduyla tamamlanırsa bu değerden küçük olabilir. Bu durumda, **Buffer** değiştirgesindeki önemsiz byte 'lar boş değere ayarlanır.

Aşağıdaki özel değer tanımlandı:

MQRL_UNDEFINED

Döndürülen verilerin uzunluğu tanımlanmadı.

z/OS üzerinde, *ReturnedLength* alanı için döndürülen değer her zaman MQRL_UNDEFINED olur.

Bu alanın ilk değeri MQRL_UNDEFINED 'dir. *Version* , MQGMO_VERSION_3' den küçükse bu alan yoksayıdır.

Reserved2 (MQGMO için MQLONG)

Bu ayrılmış bir alandır. Bu alanın ilk değeri boş bir karakterdir. *Version* **MQGMO_VERSION_4** değerinden küçükse bu alan yoksayılr.

MQGMO için MsgHandle (MQHMSG)

MQGMO_PROPERTIES_AS_Q_DEF seçeneği belirtilirse ve **PropertyControl** kuyruk özniteliği MQPROP_FORCE_MQRFH2 olarak ayarlanmamışsa, bu tanıttıcı, kuyruktan alınmakta olan iletinin özellikleriyle doldurulacak bir iletiye ilişkin tanıttıcıdır. Tanıttıcı bir MQCRTMH çağrısıyla yaratılır. Bir ileti alınmadan önce, tanıttıcı ile önceden ilişkilendirilmiş özellikler temizlenir.

Aşağıdaki değer de belirtilebilir:

MQHM_NONE

İleti tanıttıcısı belirtilmedi.

Geçerli bir ileti tanıttıcısı belirtilirse ve çıkışta ileti özelliklerini içerecek şekilde kullanılırsa, MQGET çağrısında ileti tanımlayıcısı gerekmez; giriş alanları için, ileti tanıttıcısıyla ilişkili ileti tanımlayıcısı kullanılır.

MQGET çağrısında bir ileti tanımlayıcısı belirtilirse, bu her zaman bir ileti tanıttıcısıyla ilişkilendirilmiş ileti tanımlayıcısından önceliklidir.

MQGMO_PROPERTIES_FORCE_MQRFH2 ya da MQGMO_PROPERTIES_Q_DEF belirtilirse ve **PropertyControl** kuyruk özniteliği MQPROP_FORCE_MQRFH2 ise, ileti tanımlayıcı parametresi belirtilmediğinde çağrı başarısız olur.

MQGET çağrısından döndüğünüzde, bu ileti tanıttıcısıyla ilişkili özellikler ve ileti tanımlayıcısı, alınan iletinin durumunu (MQGET çağrısında belirtildiyse, ileti tanımlayıcısının yanı sıra) yansıtacak şekilde güncellenir. Daha sonra, MQINQMP çağrısı kullanılarak iletinin özellikleri sorulabiliyor.

İleti tanımlayıcı uzantıları dışında, varsa, ileti verilerinde MQINQMP çağrısıyla sorulabilecek bir özellik bulunmaz; kuyruktaki ileti, ileti verilerinde özellikler içeriyorsa, veriler uygulamaya döndürülmeden önce bunlar ileti verilerinden kaldırılır.

Hiçbir ileti tanıttıcısı sağlanmazsa ya da Sürüm MQGMO_VERSION_4 ' den küçükse, MQGET çağrısında geçerli bir ileti tanımlayıcısı sağlamanız gerekir. İleti verilerinde, MQGMO yapısında ve **PropertyControl** kuyruk özniteliğinde özellik seçeneklerinin değerine konu olan ileti özellikleri (ileti tanımlayıcısında bulunanlar dışında) döndürülür.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQHM_NONE. **Version** , MQGMO_VERSION_4' den küçükse bu alan yoksayılr.

MQIIH- IMS bilgi üstbilgisi

MQIIH yapısı, IMS köprüsünde IMS ' e gönderilen bir iletiye ilişkin üstbilgi bilgilerini tanımlar. Desteklenen herhangi bir IBM MQ altyapısı için MQIIH yapısını içeren bir ileti yaratabilir ve iletebilir, ancak IMS köprüsünü yalnızca bir IBM MQ for z/OS kuyruk yöneticisi kullanabilir. Bu nedenle, iletinin z/OS dışı bir kuyruk yöneticisinden IMS ' e ulaşabilmesi için kuyruk yöneticisi ağızının, iletinin yönetilebileceği en az bir z/OS kuyruk yöneticisi içermesi gerekir.

Kullanılabilirlik

Tüm IBM MQ sistemleri ve IBM MQ istemcileri.

Biçim Adı

MQFMT_IMS

Karakter kümesi ve kodlama

MQIIH yapısı ve uygulama iletisi verileri için kullanılan karakter kümesi ve kodlama için özel koşullar geçerlidir:

- IMS köprü kuyruğuna sahip kuyruk yöneticisine bağlanan uygulamalar, kuyruk yöneticisinin karakter kümesinde ve kodlamasında bulunan bir MQIIH yapısı sağlamalıdır. Bunun nedeni, bu durumda MQIIH yapısına ilişkin veri dönüştürmenin gerçekleştirilmemiş olmasıdır.
- Diğer kuyruk yöneticilerine bağlanan uygulamalar, desteklenen karakter kümelerinde ve kodlamasında bulunan bir MQIIH yapısı sağlayabilir; IMS köprü kuyruğunun sahibi olan kuyruk yöneticisine bağlı alıcı ileti kanalı aracısı MQIIH ' yi dönüştürür.
- MQIIH yapısını izleyen uygulama iletileri, MQIIH yapısıyla aynı karakter kümesinde ve kodlamada olmalıdır. MQIIH yapısındaki *CodedCharSetId* ve *Encoding* alanlarını, uygulama ileti verilerinin karakter kümesini ve kodlamasını belirtmek için kullanmayın.

Veriler, kuyruk yöneticisi tarafından desteklenen yerleşik biçimlerden biri değilse, uygulama ileti verilerini dönüştürmek için bir veri dönüştürme çıkışı sağlamanız gerekir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 497. MQIIH için MQIIH ' deki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQIIH_STRUC_ID	' IIH¬ '
<u>Sürüm</u> (yapı sürümü numarası)	MQIIH_VERSION_1	1
<u>StrucLength</u> (MQIIH yapısının uzunluğu)	MQIIH_LENGTH_1	84
<u>Kodlama</u> (ayrılmış-bkz. "Karakter kümesi ve kodlama" sayfa 405)	Yok	0
<u>CodedCharSetId</u> (ayrılmış-bkz. "Karakter kümesi ve kodlama" sayfa 405)	Yok	0
<u>Biçim</u> (MQIIH ' yi izleyen verilerinMQ biçimi adı)	MQFMT_NONE	Boşluklar
<u>İşaretler</u> (işaretler)	MQIIH_NONE	0
<u>LTermOverride</u> (mantıksal uçbirim geçersiz kılma)	Yok	Boşluklar
<u>MFSMapName</u> (ileti biçimi hizmetleri eşlemi adı)	Yok	Boşluklar
<u>ReplyToBiçim</u> (MQ yanıt iletilerinin biçim adı)	MQFMT_NONE	Boşluklar
<u>Authenticator</u> (RACF parolası ya da geçiş bileti)	MQIAUT_NONE	Boşluklar
<u>TranInstanceId</u> (hareket eşgörünümü tanıtıcısı)	MQITII_NONE	Boş Değerler
<u>TranState</u> (hareket durumu)	MQITS_NOT_IN_ETKILE ŞİM	' ¬ '
<u>CommitMode</u> (kesinleştirme kipi)	MQICM_COMMIT_THEN _SEND	' 0 '
<u>SecurityScope</u> (güvenlik kapsamı)	MQISS_CHECK	' C '
<u>Ayrılmış</u> (ayrılmış)	Yok	' ¬ '

Çizelge 497. MQIIH için MQIIH ' deki Alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
Notlar:		
1. ~ simgesi, tek bir boş karakteri temsil eder.		
2. C programlama dilinde, makro değişkeniMQIIH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:		
<pre>MQIIH MyIIH = {MQIIH_DEFAULT};</pre>		

Dil bildirimleri

MQIIH için C bildirimini

```
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    MFMapName;       /* 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 */
};
```

MQIIH için COBOL bildirimini

```
** 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-STRUCLNGTH 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.

```

MQIIH için PL/I bildirimi

```

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 MFSMapName 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 */

```

MQIIH için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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
MQIIH_FLAGS DS F Flags
MQIIH_LTERMOVERRIDE 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)

```

MQIIH için Visual Basic bildirimi

```

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'
MFSMapName 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

```


MQIIH için StrucId (MQCHAR4)

Bu, IMS bilgi üstbilgisi yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQIIH_STRUC_ID.

Değer şöyle olmalıdır:

MQIIH_STRUC_ID

IMS bilgi üstbilgisi yapısına ilişkin tanıtıcı.

C programlama dili için, MQIIH_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQIIH_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQIIH için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MQIIH_VERSION_1

IMS bilgi üstbilgisi yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQIIH_CURRENT_VERSION

IMS bilgi üstbilgisi yapısının geçerli sürümü.

Bu alanın ilk değeri MQIIH_VERSION_1' dir.

MQIIH için StrucLength (MQLONG)

Bu, MQIIH yapısının uzunluğudur. Değer şöyle olmalıdır:

MQIIH_LENGTH_1

IMS bilgi üstbilgisi yapısının uzunluğu.

Bu alanın ilk değeri MQIIH_LENGTH_1' dir.

MQIIH için kodlama (MQLONG)

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri 0 'dir.

MQIIH yapısını izleyen desteklenen yapıların kodlaması, MQIIH yapısının kendisiyle aynıdır ve önceki MQ üstbilgisinden alınır.

MQIIH için CodedCharSetId (MQLONG)

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri 0 'dir.

MQIIH yapısını izleyen desteklenen yapılarla ilişkin karakter kümesi tanıtıcısı, MQIIH yapısının kendisiyle aynıdır ve önceki MQ üstbilgisinden alınır.

MQIIH için biçim (MQCHAR8)

MQIIH yapısını izleyen verilerin MQ biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın uzunluğu MQ_FORMAT_LENGTH tarafından verilir. Bu alanın ilk değeri MQFMT_NONE.

MQIIH için İşaretler (MQLONG)

İşaret değeri şöyle olmalıdır:

MQIIH_NONE

Bayrak yok.

MQIIH_PASS_SÜRE SONU

Yanıt iletisi şunları içerir:

- İstek iletisiyle aynı süre bitimi raporu seçenekleri
- Köprünün işleme saati için ayarlama yapılmadan istek iletisinden kalan süre sonu

Bu deęer ayarlanmazsa, süre sonu *sınırsız* olarak ayarlanır.

MQIIH_REPLY_FORMAT_NONE

MQIIH.Format alanı.

MQIIH_IGNORE_PURG

OTMA önekindeki TMAMIPRG göstergesini ayarlar ve OTMA 'nın CM0 hareketleri için TP PCB' deki PURG çağrılarını yoksaymasını ister.

MQIIH_CM0_REQUEST_RESPONSE

Kesinleştirme Kipi 0 (CM0) hareketleri için bu işaret, OTMA önekinde TMAMHRSP göstergesini ayarlar. Bu gösterge ayarlandığında, özgün IMS uygulama programı IOPCB ' ye yanıt vermezse ya da başka bir işleme ileti geçmediğinde OTMA/IMS tarafından bir DFS2082 RESPONSE MODE TRANSACTION TERMINATED NOT REPLY iletisi oluşturulmasını ister.

Bu alanın ilk deęeri MQIIH_NONE.

MQIIH için LTermOverride (MQCHAR8)

GÇ PCB alanına yerleştirilen mantıksal uçbirim geçersiz kılma deęeri. İsteęe baęlıdır; belirtilmezse, TPIPE adı kullanılır. İlk bayt boşsa ya da boş deęerliyse dikkate alınmaz.

Bu alanın uzunluęu MQ_LTERM_OVERRIDE_LENGTH tarafından verilir. Bu alanın ilk deęeri 8 boş karakterdir.

MQIIH için MFSTMapName (MQCHAR8)

GÇ PCB alanına yerleştirilen ileti biçimi hizmetleri eşlemi adı. Bu isteęe baęlıdır. Girişte MID 'yi temsil eder, çıkışta MOD' yi temsil eder. İlk bayt boş deęerli ya da boş ise dikkate alınmaz.

Bu alanın uzunluęu MQ_MFS_MAP_NAME_LENGTH tarafından verilir. Bu alanın ilk deęeri 8 boş karakterdir.

MQIIH için ReplyToBiçimi (MQCHAR8)

Bu, yürürlükteki iletiye yanıt olarak gönderilen yanıt iletisinin MQ biçim adıdır. Bu alanın uzunluęu MQ_FORMAT_LENGTH tarafından verilir. Bu alanın ilk deęeri MQFMT_NONE.

MQGMO_CONVERT kullanarak yanıt iletisindeki verileri dönüştürmek için MQIIH.replyToFormat=MQFMT_STRING ya da MQIIH.replyToFormat=MQFMT_IMS_VAR_STRING belirtin. Bu alanların kullanımına ilişkin açıklamalar için bkz. [“MQMD için biçim \(MQCHAR8\)” sayfa 444.](#)

İstek iletisinde varsayılan deęer (MQIIH.replyToFormat=MQFMT_NONE) kullanılırsa ve yanıt iletisi MQGMO_CONVERT kullanılarak alınırsa, veri dönüştürme gerçekleştirilmez.

MQIIH için Kimlik Doğrulayıcı (MQCHAR8)

Bu, RACF parolası ya da PassTicket' dir. İsteęe baęlıdır; belirtildiyse, güvenlik baęlamı sağlamak üzere IMS ' e gönderilen bir UTOKEN oluşturmak için MQMD güvenlik baęlamındaki kullanıcı kimlięiyle birlikte kullanılır. Belirtilmezse, kullanıcı kimlięi doęrulanmadan kullanılır. Bu, bir kimlik doęrulayıcının var olmasını gerektirebilecek RACF anahtarlarının ayarına baęlıdır.

İlk bayt boş deęerli ya da boş ise bu deęer dikkate alınmaz. Aşaęıdaki özel deęer kullanılabilir:

MQIAUT_NONE

Kimlik doęrulaması yok.

C programlama dili için, MQIAUT_NONE_ARRAY deęişmezi de tanımlandı; bu deęer MQIAUT_NONE ile aynı, ancak dizgi yerine bir karakter dizisi.

Bu alanın uzunluęu MQ_AUTHENTICATOR_LENGTH tarafından verilir. Bu alanın ilk deęeri MQIAUT_NONE.

MQIIH için TranInstanceTanıtıcısı (MQBYTE16)

Bu, hareket eşęörünümü tanıtıcısıdır. Bu alan, IMS' in çıkış iletileri tarafından kullanıldığından, ilk girişte yoksayılr. *TranState* deęerini MQITS_IN_ETKİLEŞİM olarak ayarlarsanız, IMS ' un iletileri doęru

etkileşimde ilintilendirmesini sağlamak için sonraki girişte ve sonraki tüm girişlerde bu belirtilmelidir. Aşağıdaki özel değeri kullanabilirsiniz:

MQITII_NONE

Hareket eşgörünümü tanıtıcısı yok.

C programlama dili için, MQITII_NONE_ARRAY değişmezi de tanımlanır; bu değer MQITII_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu alanın uzunluğu MQ_TRAN_INSTANCE_ID_LENGTH tarafından verilir. Bu alanın ilk değeri MQITII_NONE.

MQIIH için TranState (MQCHAR)

Bu, IMS etkileşim durumunu gösterir. Etkileşim olmadığı için ilk girişte bu yoksayılr. Sonraki girişlerde, bir etkileşimin etkin olup olmadığını gösterir. Çıkışta IMStarafından ayarlanır. Değer aşağıdakilerden biri olmalıdır:

MQITS_IN_ETKILEŞİM


Sohbet sırasında.

MQITS_NOT_IN_ETKILEŞİM

Sohbette değil.

MQITS_MIMARİLMİŞ

Hareket durumu verilerini mimari biçimde döndürür.

Bu değer yalnızca IMS /DISPLAY TRAN komutuyla kullanılır. Hareket durumu verilerini karakter formu yerine IMS mimariye sahip biçiminde döndürür.  Daha fazla bilgi için, bkz. [IMS hareket programlarını IBM MQ aracılığıyla yazma](#).

Bu alanın ilk değeri MQITS_NOT_IN_ETKILEŞİM.

MQIIH için CommitMode (MQCHAR)

Bu, IMS kesinleştirme kipidir. IMS kesinleştirme kiplerine ilişkin ek bilgi için *OTMA Reference* adlı yayına bakın. Değer aşağıdakilerden biri olmalıdır:

MQICM_COMMIT_THEN_SEND

Kesinleştir ve gönder.

Bu mod, çıktının çift kuyruğa alındığını, ancak bölgenin kullanım sürelerinin kısaldığını gösterir. Hızlı yol ve etkileşimli hareketler bu kiple çalışamaz.

MQICM_SEND_THEN_COMMIT

Gönder ve kesinleştir.

MQICM_SEND_THEN_COMMIT kesinleştirme kipinin sonucu olarak başlatılan herhangi bir IMS hareketi, hareketin IMS sistem tanımlamasında nasıl tanımlandığına bakılmaksızın RESPONSE kipinde çalışır (TRANSACT makrosunda MSGTYPE parametresi). Bu, bir işlem anahtarı aracılığıyla başlatılan işlemler için de geçerlidir.

Bu alanın ilk değeri: MQICM_COMMIT_THEN_SEND.

MQIIH için SecurityScope (MQCHAR)

Bu, IMS güvenlik işleminin gerekli olduğunu gösterir. Aşağıdaki değerler tanımlanır:

MQISS_CHECK

Güvenlik kapsamını denetleyin: Denetim bölgesinde bir ACEE oluşturulmuştur, ancak bağımlı bölgede oluşturulmaz.

MQISS_FULL

Tam güvenlik kapsamı: Denetim bölgesinde önbelleğe alınmış bir ACEE oluşturulur ve bağımlı bölgede önbelleğe alınmamış bir ACEE oluşturulur. MQISS_FULL kullanıyorsanız, ACEE 'nin oluşturulduğu kullanıcı kimliğinin bağımlı bölgede kullanılan kaynaklara erişimi olduğunu doğrulayın.

Bu alan için MQISS_CHECK ya da MQISS_FULL belirtilmezse, MQISS_CHECK varsayılr.

Bu alanın ilk deęeri MQISS_CHECK 'dir.

MQIIH için ayrılmıř (MQCHAR)

Bu ayrılmıř bir alandır; boş olmalıdır.

MQIMPO-İleti özellik seçeneklerini sorgularken sor

MQIMPO yapısı, uygulamaların iletilerin özelliklerinin nasıl sorulduęunu denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQINQMP çağrısındaki bir giriş deęiřtirgesidir.

Kullanılabilirlik

Tüm IBM MQ sistemleri ve IBM MQ istemcileri.

Karakter kümesi ve kodlama

MQIMPO içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında (MQENC_NATIVE) yer almalıdır.

Alanlar

Not: Ařaęıdaki tabloda, alanlar alfabetik olarak deęil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

<i>Çizelge 498. MQIPMO 'daki Alanlar</i>		
Alan adı ve açıklaması	Deęiřmezin adı	Deęiřmezin ilk deęeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQIMPO_STRUC_ID	'IMPO'
<u>Sürüm</u> (yapı sürümü numarası)	MQIMPO_VERSION_1	1
<u>Seçenekler</u> (MQINQMP işlemini denetleyen seçenekler)	MQIMPO_INQ_FIRST	
<u>RequestedEncoding</u> (istenen özellięin dönüřtürüleceęi kodlama)	MQENC_NATIVE	
<u>RequestedCCSID</u> (sorulan özellięin karakter takımı)	MQCCSI_APPL	
<u>ReturnedEncoding</u> (döndürülen deęerin kodlaması)	MQENC_NATIVE	
<u>ReturnedCCSID</u>	0	
<u>Reserved1</u> (ayrılmış alan)	bořluk karakteri (4 baytlık alan)	
<u>ReturnedName</u> (sorulmuş özellięin adı)	MQCHARV_DEFAULT	
<u>TypeString</u> (özellięin veri tipinin dizgi gösterimi)	Boř deęerli dizilim ya da bořluklar	

Notlar:

- Boř deęerli dizilim ya da bořluk deęeri, C dizgisindeki boř dizgiyi ve dięer programlama dillerindeki boř karakterleri gösterir.
- C programlama dilinde, makro deęiřkeniMQIMPO_DEFAULT, çizelgede listelenen deęerleri içerir. Yapıdaki alanlar için bařlangıç deęerleri sağlamak üzere bunu ařaęıdaki řekilde kullanın:

```
MQIMPO MyIMPO = {MQIMPO_DEFAULT};
```

Dil bildirimleri

MQIMPO için C bildirimini

```
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 */
};
```

MQIMPO için COBOL bildirimini

```
** 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.
```

MQIMPO için PL/I bildirimini

```
dcl
1 MQIMPO 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 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
```

```

3 TypeString      char(8);      /* Property data type as
                                name */
                                string */

```

MQIMPO için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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
*                ORG 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)

```

MQIMPO için StrucId (MQCHAR4)

Bu, sorgunun ileti özelliği seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQIMPO_STRUC_ID.

Değer şöyle olmalıdır:

MQIMPO_STRUC_ID

Sorma iletisi özellik seçenekleri yapısına ilişkin tanıtıcı.

C programlama dili için, MQIMPO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQIMPO_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQIMPO için Sürüm (MQLONG)

İleti sorma özelliği seçenekleri yapısı-Sürüm alanı

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MQIMPO_VERSION_1

Sorma iletisi özellik seçenekleri yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQIMPO_CURRENT_VERSION

Sorma iletisi özellik seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQIMPO_VERSION_1' dir.

MQIMPO için Seçenekler (MQLONG)

İleti özellik seçenekleri yapısını sorgularken sorma-Seçenekler alanı

Aşağıdaki seçenekler MQINQMP işlemi denetler. Bu seçeneklerden birini ya da birkaçını belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemi kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyse).

Geçerli olmayan seçenek birleşimleri belirtilir; diğer tüm birleşimler geçerlidir.

Değer verileri seçenekleri: Aşağıdaki seçenekler, özellik iletiden alındığında değer verilerinin işlenmesiyle ilgilidir.

MQIMPO_CONVERT_VALUE

Bu seçenek, özelliğin değerinin, MQINQMP çağrısının *Value* alanındaki özellik değerini döndürmesinden önce belirtilen *RequestedCCSID* ve *RequestedEncoding* değerlerine uyacak şekilde dönüştürülmesini ister.

- Dönüştürme başarılı olursa, MQINQMP çağrısından dönüşte *ReturnedCCSID* ve *ReturnedEncoding* alanları *RequestedCCSID* ve *RequestedEncoding* ile aynı değere ayarlanır.
- Dönüştürme başarısız olursa, ancak MQINQMP çağrısı hatasız tamamlanmazsa, özellik değeri dönüştürülmeden döndürülür.

Özellik bir dizgiyse, *ReturnedCCSID* ve *ReturnedEncoding* alanları, dönüştürülmemiş dizginin karakter kümesine ve kodlamasına ayarlanır.

Bu durumda tamamlama kodu MQCC_WARNING, neden kodu MQRC_PROP_VALUE_NOT_CONVERTED. Özellik imleci, döndürülen özelliğe ilerler.

Özellik değeri dönüştürme sırasında genişler ve **Value** değiştirgesinin boyutunu aşarsa, MQCC_FAILED tamamlanma koduyla değer dönüştürülmez; neden kodu MQRC_19TY_VALUE_TOO_BIG olarak ayarlanır.

MQINQMP çağrısının **DataLength** değiştirgesi, uygulamanın dönüştürülen özellik değerini barındırmak için gereken arabellek büyüklüğünü saptamasını sağlamak için özellik değerinin dönüştürüleceği uzunluğu döndürür. Özellik imleci değiştirilmedi.

Bu seçenek ayrıca şunları da ister:

- Özellik adı bir joker karakter içeriyorsa, ve
- *ReturnedName* alanı, döndürülen ad için bir adres ya da görelî konum ile ilk kullanıma hazırlanır.

Daha sonra, döndürülen ad *RequestedCCSID* ve *RequestedEncoding* değerlerine uygun olacak şekilde dönüştürülür.

- Dönüştürme başarılı olursa, *ReturnedName* alanının *VSCCSID* alanı ve döndürülen adın kodlaması, *RequestedCCSID* ve *RequestedEncoding* giriş değerine ayarlanır.
- Dönüştürme başarısız olursa, ancak MQINQMP çağrısı hata ya da uyarı olmadan tamamlanmazsa, döndürülen ad dönüştürülmez. Bu durumda tamamlama kodu MQCC_WARNING, neden kodu MQRC_PROP_NAME_NOT_CONVERTED.

Özellik imleci, döndürülen özelliğe ilerler. Hem değer hem de ad dönüştürülmezse, MQRC_PROP_VALUE_NOT_CONVERTED döndürülür.

Döndürülen ad dönüştürme sırasında genişler ve *RequestedName* alanının *VSBufsize* alanının boyutunu aşarsa, döndürülen dizgi dönüştürülmeden bırakılır ve tamamlama kodu MQCC_FAILED olur ve neden kodu MQRC_NAMETY_TOO_BIG olarak ayarlanır.

MQCHARV yapısının *VSLength* alanı, uygulamanın dönüştürülen özellik değerini barındırmak için gereken arabellek büyüklüğünü saptamasına izin vermek için özellik değerinin dönüştürüleceği uzunluğu döndürür. Özellik imleci değiştirilmedi.

MQIMPO_CONVERT_TYPE

Bu seçenek, özelliğin değerinin yürürlükteki veri tipinden MQINQMP çağrısının **Type** değiştirgesinde belirtilen veri tipine dönüştürülmesini ister.

- Dönüştürme başarılı olursa, MQINQMP çağrısının döndürülmesinde **Type** değiştirgesi değişmez.
- Dönüştürme başarısız olursa, ancak MQINQMP çağrısı hatasız olarak tamamlanırsa, çağrı MQRC_PROP_CONV_NOT_SUPPORTED nedeniyle başarısız olur. Özellik imleci değiştirilmedi.

Veri tipinin dönüştürülmesi, dönüştürme sırasında değer genişlemesine neden olursa ve dönüştürülen değer **Value** parametresinin boyutunu aşarsa, MQCC_FAILED tamamlanma koduyla değer dönüştürülmez ve neden kodu MQRC_19TY_VALUE_TOO_BIG olarak ayarlanır.

MQINQMP çağrısının **DataLength** değıştirgesi, uygulamanın dönüştürülen özellik değerini barındırmak için gereken arabellek büyüklüğünü saptamasını sağlamak için özellik değerinin dönüştürüleceğı uzunluğu döndürür. Özellik imleci değıştirilmedi.

MQINQMP çağrısının **Type** değıştirgesinin değeri geçerli değilse, çağrı başarısız olur; neden MQRC_19TY_TYPE_ERROR?

İstenen veri tipi dönüşümü desteklenmiyorsa, çağrı başarısız olur; nedeni MQRC_PROP_CONV_NOT_SUPPORTED olabilir. Aşağıdaki veri tipi dönüşümleri desteklenir:

Özellik veri tipi	Desteklenen hedef veri tipleri
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 (boş değeri)	Yok

Desteklenen dönüşümleri düzenleyen genel kurallar şunlardır:

- Dönüşürme sırasında veri kaybı olmaması koşuluyla, sayısal özellik değeri bir veri tipinden diğeriye dönüştürülebilir.
Örneğın, MQTYPE_INT32 veri tipine sahip bir özelliğın değeri, MQTYPE_INT64 veri tipine sahip bir değere dönüştürülebilir, ancak MQTYPE_INT16 veri tipiyle bir değere dönüştürülemez.
- Herhangi bir veri tipindeki bir özellik değeri dizgiye dönüştürülebilir.
- Dizginin dönüşürme için doğru biçimlenmesi koşuluyla, dizgi özelliğı değeri başka bir veri tipine dönüştürülebilir. Bir uygulama doğru biçimlenmemiş bir dizgi özelliğı değerini dönüşürmeyi denerse, IBM MQ , MQRC_PROP_NUMBER_FORMAT_ERROR neden kodunu döndürür.
- Bir uygulama desteklenmeyen bir dönüşürme girişiminde bulunursa, IBM MQ MQRC_PROP_CONV_NOT_SUPPORTED neden kodunu döndürür.

Bir özellik değerini bir veri tipinden diğeriye dönüşürmeye ilişkin kurallar şunlardır:

- MQTYPE_BOOLEAN özellik değeri bir dizgiye dönüştürülürken, TRUE değeri "TRUE" dizgisine ve false değeri "FALSE" dizgisine dönüştürülür.
- Bir MQTYPE_BOOLEAN özellik değeri sayısal bir veri tipine dönüştürülürken, TRUE değeri bir e dönüştürülür ve FALSE değeri sıfıra dönüştürülür.
- Bir dizgi özelliğı değeri MQTYPE_BOOLEAN değerine dönüştürülürken, "TRUE" ya da "1" dizgisi TRUE değerine ve "FALSE" ya da "0" dizgisi FALSE değerine dönüştürülür.

"TRUE" ve "FALSE" terimlerinin büyük ve küçük harfe duyarlı olmadığını unutmayın.

Başka dizgiler dönüştürülemiyor; IBM MQ , MQRC_PROP_NUMBER_FORMAT_ERROR neden kodunu döndürür.

- Bir dizgi özelliği değerini MQTYPE_INT8, MQTYPE_INT16, MQTYPE_INT32 ya da MQTYPE_INT64 veri tipli bir değere dönüştürürken, dizgi aşağıdaki biçimde olmalıdır:

```
[blanks][sign]digits
```

Dizginin bileşenlerinin anlamları aşağıdaki gibidir:

blanks

İsteğe bağlı baştaki boş karakterler

sign

İsteğe bağlı bir artı işareti (+) ya da eksi işareti (-) karakteri.

digits

Bitişik sayı karakterleri sırası (0-9). En az bir basamak karakteri bulunmalıdır.

Sayı karakterleri dizgisinden sonra, dizilim rakam karakterleri olmayan diğer karakterleri içerebilir, ancak bu karakterlerin ilkine ulaşılır ulaşılmaz dönüştürme durur. Dizginin bir ondalık tamsayıyı temsil ettiği varsayılır.

IBM MQ , dizgi doğru biçimlendirilmediyse, MQRC_PROP_NUMBER_FORMAT_ERROR neden kodunu döndürür.

- Bir dizgi özelliği değeri MQTYPE_FLOAT32 ya da MQTYPE_FLOAT64 veri tipine sahip bir değere dönüştürülürken, dizgi aşağıdaki biçimde olmalıdır:

```
[blanks][sign]digits[.digits][e_char[e_sign]e_digits]
```

Dizginin bileşenlerinin anlamları aşağıdaki gibidir:

blanks

İsteğe bağlı baştaki boş karakterler

sign

İsteğe bağlı bir artı işareti (+) ya da eksi işareti (-) karakteri.

digits

Bitişik sayı karakterleri sırası (0-9). En az bir basamak karakteri bulunmalıdır.

e_char

"E" ya da "e" olan üstel bir karakter.

e_sign

Üstel için isteğe bağlı artı işareti (+) ya da eksi işareti (-) karakteri.

e_digits

Üs için bitişik sayı karakterleri sırası (0-9). Dizgi bir üstel karakter içeriyorsa, en az bir basamak karakteri bulunmalıdır.

Sayı karakterleri sırasından ya da bir üstel karakteri gösteren isteğe bağlı karakterlerden sonra, dizgi rakam karakteri olmayan diğer karakterleri içerebilir, ancak bu karakterlerin ilkine ulaşıldığında dönüştürme durur. Dizginin, 10 üslü bir ondalık kayan nokta sayısını temsil ettiği varsayılır.

IBM MQ , dizgi doğru biçimlendirilmediyse, MQRC_PROP_NUMBER_FORMAT_ERROR neden kodunu döndürür.

- Sayısal bir özellik değeri dizgiye dönüştürülürken, değer, o değere ilişkin ASCII karakterini içeren dizgiye değil, değer ondalık sayı olarak dizgi gösterimine dönüştürülür. Örneğin, 65 tamsayısı "A" dizgisine değil, "65" dizgisine dönüştürülür.
- Bayt dizilimi özellik değeri dizgiye dönüştürülürken, her bayt, baytı gösteren iki onaltılı karaktere dönüştürülür. Örneğin, {0xF1, 0x12, 0x00, 0xFF} bayt dizisi "F11200FF" dizgisine dönüştürülür.

MQIMPO_SORGU_UZUNLUĞU

Özellik değerinin tipini ve uzunluğunu sorgulayın. Uzunluk, MQINQMP çağrısının **DataLength** değiştirgesinde döndürülür. Özellik değeri döndürülmedi.

Bir **ReturnedName** arabelleği belirtilirse, MQCHARV yapısının *VSLength* alanı özellik adının uzunluğuyla doldurulur. Özellik adı döndürülmez.

Yineleme seçenekleri: Aşağıdaki seçenekler, genel arama karakteri içeren bir ad kullanılarak özellikler üzerinde yineleme ile ilgilidir.

MQIMPO_INQ_FIRST

Belirtilen adla eşleşen ilk özelliği sorgulayın. Bu çağrıdan sonra, döndürülen özellik üzerinde bir imleç oluşturulur.

Bu varsayılan değerdir.

MQIMPO_INQ_PROP_UNDER_CURSOR seçeneği, gerekiyorsa, aynı özelliği yeniden sorgulamak için bir MQINQMP çağrısıyla birlikte kullanılabilir.

Tek bir özellik imleci olduğunu unutmayın; bu nedenle, MQINQMP çağrısında belirtilen özellik adı ilk durumuna getirilirse, geçici çizelge ilk durumuna getirilir.

Bu seçenek aşağıdaki seçeneklerden biriyle geçerli değildir:

MQIMPO_INQ_NEXT
MQIMPO_INQ_PROP_UNDER_CURSOR

MQIMPO_INQ_NEXT

Belirtilen adla eşleşen sonraki özelliğe ilişkin bilgi isteminde, özellik imlecinden aramaya devam edin. İmleç, döndürülen özelliğe ilerler.

Bu, belirtilen ada ilişkin ilk MQINQMP çağrısıya, belirtilen adla eşleşen ilk özellik döndürülür.

MQIMPO_INQ_PROP_UNDER_CURSOR seçeneği, aynı özelliği yeniden sorgulamak için gerekirse bir MQINQMP çağrısıyla birlikte kullanılabilir.

Geçici çizelgenin altındaki özellik silindiyse, MQINQMP silinen özelliğin ardından eşleşen bir sonraki özelliği döndürür.

Bir yineleme devam ederken genel arama karakteriyle eşleşen bir özellik eklenirse, özellik yineleme tamamlanırken döndürülebilir ya da döndürülemeyebilir. Yineleme MQIMPO_INQ_FIRST kullanılarak yeniden başlatıldıktan sonra özellik döndürülür.

Yineleme devam ederken silinen joker karakterle eşleşen bir özellik, silindikten sonra döndürülmez.

Bu seçenek aşağıdaki seçeneklerden biriyle geçerli değildir:

MQIMPO_INQ_FIRST
MQIMPO_INQ_PROP_UNDER_CURSOR

MQIMPO_INQ_PROP_UNDER_CURSOR

Özellik imlecinin gösterdiği özelliğin değerini alın. Özellik imlecinin gösterdiği özellik, MQIMPO_INQ_FIRST ya da MQIMPO_INQ_NEXT seçeneği kullanılarak en son sorulmuş özelliktir.

İleti tanıtıcısı yeniden kullanıldığında, ileti tanıtıcısı MQGET çağrısındaki MQGMO ' nun *MsgHandle* alanında belirtildiğinde ya da ileti tanıtıcısı bir MQPUT çağrısındaki MQPMO yapısının *OriginalMsgHandle* ya da *NewMsgHandle* alanlarında belirtildiğinde özellik geçici çizelgesi ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında ya da özellik imleci tarafından gösterilen özellik silindiğinde bu seçenek kullanılırsa, çağrı tamamlanma kodu MQCC_FAILED ve neden MQRC_PROPERTY_NOT_ALLOWED ile başarısız olur.

Bu seçenek aşağıdaki seçeneklerden biriyle geçerli değildir:

MQIMPO_INQ_FIRST
MQIMPO_INQ_NEXT

Daha önce açıklanan seçeneklerden hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

MQIMPO_NONE

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar.

MQIMPO_NONE, program belgelerine yardımcı olur; bu seçeneğin başka bir öğeyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQIMPO_INQ_FIRST 'dir.

MQIMPO için RequestedEncoding (MQLONG)

İleti özellik seçenekleri yapısını sorgularken- RequestedEncoding alanı

Bu, MQIMPO_CONVERT_VALUE ya da MQIMPO_CONVERT_TYPE belirtildiğinde, sormalı özellik değerinin dönüştürüleceği kodlamadır.

Bu alanın ilk değeri MQENC_NATIVE.

MQIMPO için RequestedCCSID (MQLONG)

İleti özellik seçenekleri yapısını sorgularken- RequestedCCSID alanı

Değer bir karakter dizgisiyse, sorulmuş özellik değerinin dönüştürüleceğini karakter kümesi. Bu, MQIMPO_CONVERT_VALUE ya da MQIMPO_CONVERT_TYPE belirtildiğinde *ReturnedName* 'nin dönüştürüleceği karakter takımıdır.

Bu alanın ilk değeri MQCCSI_APPL 'dir.

MQIMPO için ReturnedEncoding (MQLONG)

İleti özellik seçenekleri yapısını sorgularken- ReturnedEncoding alanı

Çıkışta bu, döndürülen değer kodlamasıdır.

MQIMPO_CONVERT_VALUE seçeneği belirtilirse ve dönüştürme başarılı olursa, dönüşte *ReturnedEncoding* alanı, geçirilen değerle aynıdır.

Bu alanın ilk değeri MQENC_NATIVE.

MQIMPO için ReturnedCCSID (MQLONG)

İleti özellik seçenekleri yapısını sorgularken sor- ReturnedCCSID alanı

Çıkışta, MQINQMP çağrısının **Type** değiştirgesi MQTYPE_STRING ise, döndürülen değer karakter kümesidir.

MQIMPO_CONVERT_VALUE seçeneği belirtilirse ve dönüştürme başarılı olursa, dönüşte *ReturnedCCSID* alanı, geçirilen değerle aynıdır.

Bu alanın ilk değeri sıfırdır.

MQIMPO için Reserved1 (MQCHAR)

Bu ayrılmış bir alandır. Bu alanın ilk değeri boş bir karakterdir (4 baytlık alan).

MQIMPO için ReturnedName (MQCHARV)

İleti özellik seçenekleri yapısını sorgularken- ReturnedName alanı

Sorulmuş özelliğin gerçek adı.

Girişte, MÖCHARV yapısının *VSPtr* ya da *VSoffset* alanı kullanılarak bir dizgi arabelleđi geçirilir. Dizgi arabelleđinin uzunluđu, MÖCHARV yapısının *VSBuFSIZE* alanı kullanılarak belirtilir.

MÖINQMP çağrısından döndükdten sonra, dizgi arabelleđi adı tam olarak içerecek kadar uzun olduđu sürece, dizgi arabelleđi, sorulmuş özelliđin adıyla tamamlanır. MÖCHARV yapısının *VSLength* alanı, özellik adının uzunluđuyla doldurulur. MÖCHARV yapısının *VSCCSID* alanı, döndürülen adın karakter kümesini göstermek için doldurulur.

Bu bir giriş/çıkış alanıdır. Bu alanın ilk değeri MÖCHARV_DEFAULT.

MÖIMPO için TypeString (MÖCHAR8)

İleti özellik seçenekleri yapısını sorgularken- TypeString alanı

Özelliđin veri tipinin dizgi gösterimi.

Özellik bir MÖRFH2 üstbilgisinde belirtildiyse ve MÖRFH2 dt özniteliđi tanınmıyorsa, özelliđin veri tipini saptamak için bu alan kullanılabilir. *TypeString* , kodlanmış karakter kümesi 1208 'de (UTF-8) döndürülür ve özelliđin dt özniteliđinin değerin tanınmayan ilk sekiz baytı olur.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri C programlama dilindeki boş dizgi ve diđer programlama dillerindeki 8 boş karakterdir.

MÖMD-İleti tanımlayıcı

MÖMD yapısı, bir ileti gönderme ve alma uygulamaları arasında dolaşırken uygulama verileriyle birlikte gönderilen denetim bilgilerini içerir. Yapı, MÖGET, MÖPUT ve MÖPUT1 çağrılarında bir giriş/çıkış deđiştirgesidir.

Kullanılabilirlik

Tüm IBM MQ sistemleri ve bu sistemlere bađlı IBM MQ MQI clients .

Sürüm

MÖMD ' nin yürürlükteki sürümü: MÖMD_VERSION_2. Birkaç ortam arasında taşınabilir olması amaçlanan uygulamalar, ilgili tüm ortamlarda MÖMD ' nin gerekli sürümünün desteklendiđinden emin olmalıdır. Yapının yalnızca daha yeni sürümlerinde bulunan alanlar, izleyen açıklamalarda bu şekilde tanımlanır.

Desteklenen programlama dilleri için sağlanan üstbilgi, COPY ve INCLUDE dosyaları, ortam tarafından desteklenen en son MÖMD sürümünü içerir, ancak *Version* alanının ilk değeri MÖMD_VERSION_1olarak ayarlanmıştır. version-1 yapısında var olmayan alanları kullanmak için uygulamanın *Version* alanını gerekli sürümün sürüm numarasına ayarlaması gerekir.

version-1 yapısına ilişkin bir bildirim MÖMD1adıyla kullanılabilir.

Karakter kümesi ve kodlama

MÖMD ' deki veriler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır; bunlar **CodedCharSetId** kuyruk yöneticisi özniteliđi ve MÖENC_NATIVE ile verilir. Ancak, uygulama IBM MQ MQI clientolarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Gönderen ve alan kuyruk yöneticileri farklı karakter kümeleri ya da kodlamalar kullanıyorsa, MÖMD ' deki veriler otomatik olarak dönüştürülür. Uygulamanın MÖMD ' yi dönüştürmesi gerekli deđildir.

MÖMD ' nin farklı sürümlerini kullanma

version-2 MÖMD, version-1 MÖMD ' nin kullanılmasıyla ve ileti verilerinin MÖMDE yapısıyla önceden düzeltilmesiyle eşdeđerdir. Ancak, MÖMDE yapısındaki tüm alanların varsayılan değeri varsa, MÖMDE atlanabilir. version-1 MÖMD ve MÖMDE aşıđıda açıklandığı gibi kullanılır:

- MQPUT ve MQPUT1 çağrılarında uygulama bir version-1 MQMD sağlarsa, uygulama isteğe bağlı olarak ileti verilerine bir MQMDE öneki gönderebilir ve MQMDE 'nin var olduğunu göstermek için MQMD' deki *Format* alanını MQFMT_MD_EXTENSION olarak ayarlayabilir. Uygulama bir MQMDE sağlamazsa, kuyruk yöneticisi MQMDE ' deki alanlar için varsayılan değerleri varsayar.

Not: version-2 MQMD 'de var olan, ancak version-1 MQMD' de olmayan alanlar, MQPUT ve MQPUT1 çağrılarındaki giriş/çıkış alanlarıdır. Ancak, kuyruk yöneticisi MQPUT ve MQPUT1 çağrılarının çıkışında MQMDE ' deki eşdeğer alanlarda herhangi bir değer döndürmez; uygulama bu çıkış değerlerini gerektiriyorsa, bir version-2 MQMD kullanmalıdır.

- MQGET çağrısında, uygulama version-1 MQMD sağlarsa, kuyruk yöneticisi MQMDE ile döndürülen iletiyi önekler, ancak yalnızca MQMDE ' deki alanlardan birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa. MQMD 'deki *Format* alanı, MQMDE ' nin var olduğunu belirtmek için MQFMT_MD_EXTENSION değerine sahip olacaktır.

Kuyruk yöneticisinin MQMDE ' deki alanlar için kullandığı varsayılan değerler, [Çizelge 503 sayfa 470](#) içinde gösterilen bu alanların ilk değerleriyle aynıdır.

Bir ileti iletim kuyruğunda olduğunda, MQMD ' deki bazı alanlar belirli değerlere ayarlanır; ayrıntılar için bkz. ["MQXQH-İletim kuyruğu üstbilgisi"](#) sayfa 612 .

İleti bağlamı

MQMD ' deki bazı alanlar ileti bağlamını içerir. İki tip ileti bağlamı vardır: *kimlik bağlamı* ve *kaynak bağlamı*. Genellikle:

- Kimlik bağlamı, iletiyi *aslen* koyan uygulamayla ilgilidir
- Kaynak bağlamı, iletiyi koyan *en son* uygulamayla ilgilidir.

Bu iki uygulama aynı uygulama olabilir, ancak farklı uygulamalar da olabilir (örneğin, bir ileti bir uygulamadan diğerine iletildiğinde).

Kimlik ve kaynak bağlamın genellikle açıklanan anlamları olsa da, MQMD ' deki her iki tip bağlam alanının içeriği, ileti konusunda belirtilen MQPMO_ * _CONTEXT seçeneklerine bağlıdır. Sonuç olarak, kimlik bağlamı, ilk olarak iletiyi koyan uygulamayla ilgili olmak zorunda değildir ve kaynak bağlamı, mesajı en son koyan uygulamayla ilişkili olmak zorunda değildir; uygulama paketinin tasarımına bağlıdır.

İleti kanalı aracısı (MCA) hiçbir zaman ileti bağlamını değiştirir. Uzak kuyruk yöneticilerinden ileti alan MCA ' lar MQPUT ya da MQPUT1 çağrısında MQPMO_SET_ALL_CONTEXT bağlam seçeneğini kullanır. Bu, alan MCA 'nın gönderen MCA' dan gelen iletiyle birlikte gönderilen ileti bağlamını tam olarak korumasını sağlar. Ancak sonuç, kaynak bağlamın iletiyi gönderen ve alan MCA ' ların herhangi biriyle ilgili olmamasıdır. Kaynak bağlam, iletiyi koyan daha önceki bir uygulamaya gönderme yapıyor. Tüm ara uygulamalar ileti bağlamını geçtiyse, kaynak bağlam kaynak uygulamanın kendisini gösterir.

Tanımlarda, bağlam alanları daha önce anlatıldığı gibi kullanılır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 500. MQMD ' de MQMD için alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQMD_STRUC_ID	'MD'
Sürüm (yapı sürümü numarası)	MQMD_VERSION_1	1
Rapor (rapor iletileri için seçenekler)	MQRO_NONE	0
MsgType (ileti tipi)	MQMT_DATAGRAM	8

Çizelge 500. MQMD ' de MQMD için alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>MQMD-Süre bitimi alanı</u> (ileti geçerlik süresi)	MQEI_UNLIMITED	-1
<u>MQMD-Geribildirim alanı</u> (geribildirim ya da neden kodu)	MQFB_NONE	0
<u>Kodlama</u> (ileti verilerinin sayısal kodlaması)	MQENC_NATIVE	Ortama bağlıdır
<u>CodedCharSetId</u> (ileti verilerinin karakter kümesi tanıtıcısı)	MQCCSI_Q_MGR	0
<u>Biçim</u> (ileti verilerinin biçim adı)	MQFMT_NONE	Boşluklar
<u>Öncelik</u> (ileti önceliği)	MQPRI_PRIORITY_AS_Q_DEF	-1
<u>Kalıcılık</u> (ileti kalıcılığı)	MQPER_PERSISTENCE_AS_Q_DEF	2
<u>MQMD- MsgId alanı</u> (ileti tanıtıcısı)	MQMI_NONE	Boş Değerler
<u>CorrelId</u> (ilinti tanıtıcısı)	MQCI_NONE	Boş Değerler
<u>BackoutCount</u> (geriletme sayacı)	Yok	0
<u>ReplyToQ</u> (yanıt kuyruğunun adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>ReplyToQMgr</u> (yanıt kuyruğu yöneticisinin adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>UserIdentifier</u> (kullanıcı kimliği)	Yok	Boş değerli dizilim ya da boşluklar
<u>AccountingToken</u> (muhasabe simgesi)	MQACT_NONE	Boş Değerler
<u>AppIdentityVerileri</u> (kimlikle ilgili uygulama verileri)	Yok	Boş değerli dizilim ya da boşluklar
<u>PutApplTip</u> (iletiyi koyan uygulamanın tipi)	MQAT_NO_CONTEXT	0
<u>PutApplAdı</u> (iletiyi koyan uygulamanın adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>PutDate</u> (iletinin gönderildiği tarih)	Yok	Boş değerli dizilim ya da boşluklar
<u>PutTime</u> (iletinin konma zamanı)	Yok	Boş değerli dizilim ya da boşluklar
<u>AppOriginVerileri</u> (kaynak ile ilgili uygulama verileri)	Yok	Boş değerli dizilim ya da boşluklar
Not: <i>Version</i> , MQMD_VERSION_2' den küçükse kalan alanlar yoksayılr.		
<u>GroupId</u> (grup tanıtıcısı)	MQGI_NONE	Boş Değerler
<u>MsgSeqNumber</u> (grup içindeki mantıksal iletinin sıra numarası)	Yok	1
<u>Görelî Konum</u> (mantıksal iletinin başlangıcından fiziksel iletideki verilerin görelî konumu)	Yok	0

Çizelge 500. MQMD ' de MQMD için alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
MQMD- MsgFlags alanı (ileti işaretleri)	MQMF_NONE	0
OriginalLength (özgün iletinin uzunluğu)	MQOL_UNDEFINED	-1

Notlar:

- Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.
- C programlama dilinde, makro değişkeniMQMD_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQMD MyMD = {MQMD_DEFAULT};
```

Dil bildirimleri

MQMD için C bildirimi

```
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 */
};
```

MQMD için COBOL bildirimi

```
** 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.

```

MQMD için PL/I bildirimi

```

dcl
1 MQMD based,
3 StructId 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 ReplyToMgr 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 message */
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 */

MQMD için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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_CODEDECHARSETID DS F Character set identifier of message data
*
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 message
*
MQMD_PUTAPPLNAME DS   CL28 Name of application that put the message
*
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 within group
*
MQMD_OFFSET    DS   F    Offset of data in physical message from start of logical 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)

```

MQMD için Visual Basic bildirimi

```

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		

MQMD için StrucId (MQCHAR4)

Bu, ileti tanımlayıcı yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQMD_STRUC_ID.

Değer şöyle olmalıdır:

MQMD_STRUC_ID

İleti tanımlayıcı yapısına ilişkin tanıttıcı.

C programlama dili için, değişmez MQMD_STRUC_ID_ARRAY da tanımlanır. Bu değer MQMD_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQMD için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır ve aşağıdakilerden biri olmalıdır:

MQMD_VERSION_1

Version-1 ileti tanımlayıcı yapısı.

Bu sürüm tüm ortamlarda desteklenir.

MQMD_VERSION_2

Version-2 ileti tanımlayıcı yapısı.

Bu sürüm, bu sistemlere bağlı tüm IBM MQ V6.0 ve sonraki ortamlarda ve IBM MQ MQI clients ' de desteklenir.

Not: version-2 MQMD kullanıldığında, kuyruk yöneticisi, uygulama ileti verilerinin başında var olabilecek MQ üstbilgi yapıları üzerinde ek denetimler gerçekleştirir; daha fazla ayrıntı için MQPUT çağrısına ilişkin kullanım notlarına bakın.

Yalnızca yapının daha yeni sürümünde var olan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQMD_CURRENT_VERSION

İleti tanımlayıcı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQMD_VERSION_1' dir.

MQMD için Rapor (MQLONG)

Rapor iletisi, bir uygulamayı özgün iletiyle ilgili beklenen ya da beklenmeyen olaylar hakkında bilgilendirmek için kullanılan başka bir iletiyle ilgili bir iletidir. *Report* alanı, uygulamanın özgün iletiyi göndererek hangi rapor iletilerinin gerekli olduğunu, uygulama iletisi verilerinin bunlara dahil edilip edilmeyeceğini ve ayrıca (hem raporlar hem de yanıtlar için) rapor ya da yanıt iletisindeki ileti ve ilinti tanıttıcılarının nasıl ayarlanacağını belirtmesini sağlar. Aşağıdaki rapor iletisi türlerinin herhangi biri (ya da hiçbiri) istenebilir:

- Kural dışı durum

- Süre Bitim Tarihi
- Varış sırasında onayla (COA)
- Teslimatı onayla (COD)
- Pozitif işlem bildirimi (PAN)
- Negatif işlem bildirimi (NAN)

Bu seçeneklerden birini ya da birkaçını belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyorsa).

Rapor iletisini alan uygulama, MQMD 'deki *Feedback* alanını inceleyerek raporun oluşturulma nedenini belirleyebilir; daha fazla ayrıntı için *Feedback* alanına bakın.

Bir konuya ileti koyarken rapor seçeneklerinin kullanılması, sıfır, bir veya daha fazla rapor iletisinin oluşturulmasına ve uygulamaya gönderilmesine neden olabilir. Bunun nedeni, yayın iletisinin sıfır, bir ya da daha fazla abone olan uygulamaya gönderilebilir olmasıdır.

Kural dışı durum seçenekleri: Kural dışı durum raporu iletisi istemek için listelenen seçeneklerden birini belirtin.

MQRO_EXCEPTION

İleti kanalı aracısı, bir ileti başka bir kuyruk yöneticisine gönderildiğinde ve ileti belirtilen hedef kuyruğa teslim edilemediğinde bu tip bir rapor oluşturur. Örneğin, hedef kuyruk ya da bir ara iletim kuyruğu dolu olabilir ya da ileti kuyruk için çok büyük olabilir.

Kural dışı durum raporu iletisinin oluşturulması, özgün iletinin kalıcılığına ve özgün iletinin iletildiği ileti kanalının hızına (normal ya da hızlı) bağlıdır:

- Tüm kalıcı iletiler ve normal ileti kanallarında dolaşan kalıcı olmayan iletiler için, kural dışı durum raporu yalnızca, hata koşulu için gönderen uygulama tarafından belirtilen işlem başarıyla tamamlanabiliyorsa oluşturulur. Gönderen uygulama, hata koşulu ortaya çıktığında özgün iletinin atılmasını denetlemek için aşağıdaki işlemlerden birini belirtebilir:
 - MQRO_DEAD_LETTER_Q (bu, özgün iletiyi gönderilmeyen iletiler kuyruğuna yerleştirir).
 - MQRO_DISCARD_MSG (özgün iletiyi atar).

Gönderen uygulama tarafından belirlenen işlem başarıyla tamamlanamazsa, özgün ileti iletim kuyruğunda bırakılır ve kural dışı durum raporu iletisi oluşturulmaz.

- Hızlı ileti kanallarında dolaşan kalıcı olmayan iletiler için, özgün ileti iletim kuyruğundan kaldırılır ve hata koşulu için belirtilen eylem başarıyla tamamlanamasa bile *oluşturulan kural dışı durum raporu*. Örneğin, MQRO_DEAD_LETTER_Q belirtilirse, ancak kuyruk dolu olduğu için özgün ileti teslim edilmeyen ileti kuyruğuna yerleştirilemezse, kural dışı durum raporu iletisi oluşturulur ve özgün ileti atılır.

Normal ve hızlı ileti kanalları hakkında daha fazla bilgi için bkz. [Kalıcı olmayan ileti hızı \(NPMSPEED\)](#).

Özgün iletiyi koyan uygulama, MQPUT ya da MQPUT1 çağrısının döndürdüğü neden kodu aracılığıyla sorunu zamanuyumlu olarak bildirebiliyorsa, kural dışı durum raporu oluşturulmaz.

Uygulamalar, bir iletinin işlenemeyeceğini belirtmek için kural dışı durum raporları da gönderebilir (örneğin, hesabın kredi sınırını aşmasına neden olacak bir borç işlemi olduğundan).

Özgün iletideki ileti verileri rapor iletisine dahil edilmez.

Birden çok MQRO_EXCEPTION, MQRO_EXCEPTION_WITH_DATA ve MQRO_EXCEPTION_WITH_FULL_DATA belirtmeyin.

MQRO_EXCEPTION_WITH_DATA

Bu, özgün iletideki uygulama iletisi verilerinin ilk 100 baytı rapor iletisine dahil edilmesinin dışında, MQRO_EXCEPTION ile aynıdır. Özgün ileti bir ya da daha fazla MQ üstbilgi yapısı içeriyorsa, bunlar 100 baytlık uygulama verilerine ek olarak rapor iletisine eklenir.

Birden çok MQRO_EXCEPTION, MQRO_EXCEPTION_WITH_DATA ve MQRO_EXCEPTION_WITH_FULL_DATA belirtmeyin.

MQRO_EXCEPTION_WITH_FULL_DATA

Tam veri gerektiren kural dışı durum raporları.

Bu, özgün iletideki tüm uygulama iletisi verilerinin rapor iletisine eklenmesi dışında, MQRO_EXCEPTION ile aynıdır.

Birden çok MQRO_EXCEPTION, MQRO_EXCEPTION_WITH_DATA ve MQRO_EXCEPTION_WITH_FULL_DATA belirtmeyin.

Süre sonu seçenekleri: Bir süre sonu raporu iletisi istemek için listelenen seçeneklerden birini belirleyin.

MQRO_SÜRE SONU

Bu rapor tipi, ileti bir uygulamaya teslim edilmeden önce atılırsa kuyruk yöneticisi tarafından oluşturulur çünkü süre bitimi süresi geçmiştir (bkz. *Expiry* alanı). Bu seçenek ayarlanmazsa, bu nedenle bir ileti atılırsa (MQRO_EXCEPTION_ * seçeneklerinden birini belirtseniz bile) rapor iletisi oluşturulmaz.

Özgün iletideki ileti verileri rapor iletisine dahil edilmez.

Birden çok MQRO_EXPIRATION, MQRO_EXPIRATION_WITH_DATA ve MQRO_EXPIRATION_WITH_FULL_DATA belirtmeyin.

MQRO_EXPIRATION_WITH_DATA

Bu, özgün iletideki uygulama iletisi verilerinin ilk 100 baytın rapor iletisine dahil edilmesi dışında, MQRO_SÜRE bitimi ile aynıdır. Özgün ileti bir ya da daha fazla MQ üstbilgi yapısı içeriyorsa, bunlar 100 baytlık uygulama verilerine ek olarak rapor iletisine eklenir.

Birden çok MQRO_EXPIRATION, MQRO_EXPIRATION_WITH_DATA ve MQRO_EXPIRATION_WITH_FULL_DATA belirtmeyin.

MQRO_EXPIRATION_WITH_FULL_DATA

Bu, özgün iletideki tüm uygulama iletisi verilerinin rapor iletisine eklenmesi dışında, MQRO_SÜRE bitimi ile aynıdır.

Birden çok MQRO_EXPIRATION, MQRO_EXPIRATION_WITH_DATA ve MQRO_EXPIRATION_WITH_FULL_DATA belirtmeyin.

Variş sırasında doğrulama seçenekleri: Bir variş sırasında doğrulama raporu iletisi istemek için listelenen seçeneklerden birini belirleyin.

MQRO_COA

Bu tip bir rapor, ileti hedef kuyruğa yerleştirildiğinde hedef kuyruğun sahibi olan kuyruk yöneticisi tarafından oluşturulur. Özgün iletideki ileti verileri rapor iletisine dahil edilmez.

İleti bir iş biriminin parçası olarak yerleştirilirse ve hedef kuyruk yerel bir kuyruksa, kuyruk yöneticisi tarafından oluşturulan COA rapor iletisi yalnızca iş birimi kesinleştirilirse alınabilir.

İleti tanımlayıcıdaki *Format* alanı MQFMT_XMIT_Q_HEADER ya da MQFMT_DEAD_LETTER_HEADER ise COA raporu oluşturulmaz. Bu, ileti bir iletim kuyruğuna konursa ya da teslim edilemezse ve bir teslim edilmeyen ileti kuyruğuna konursa bir COA raporunun oluşturulmasını önler.

IMS köprü kuyruğunda, ileti IMS kuyruğuna ulaştığında COA raporu oluşturulur (IMS ' dan alınan alındı bildirim) ve ileti MQ köprü kuyruğuna yerleştirildiğinde değil. Diğer bir deyişle, IMS etkin değilse, IMS başlatılıncaya ve IMS kuyruğunda bir ileti kuyruğa alınıncaya kadar COA raporu oluşturulmaz.

MQMD.Report= MQRO_COA, yanıt kuyruğu üzerinde + passid yetkisine sahip olmalıdır. Kullanıcının + passid yetkisi yoksa, COA rapor iletisi yanıt kuyruğuna ulaşmaz. Rapor iletisini ölü mektup kuyruğuna koyma girişiminde bulunuldu.

Birden çok MQRO_COA, MQRO_COA_WITH_DATA ve MQRO_COA_WITH_FULL_DATA belirtmeyin.

MQRO_COA_WITH_DATA

Bu, MQRO_COA ile aynıdır; ancak, özgün iletideki uygulama iletisi verilerinin ilk 100 baytı rapor iletisine eklenir. Özgün ileti bir ya da daha fazla MQ üstbilgi yapısı içeriyorsa, bunlar 100 baytlık uygulama verilerine ek olarak rapor iletisine eklenir.

Birden çok MQRO_COA, MQRO_COA_WITH_DATA ve MQRO_COA_WITH_FULL_DATA belirtmeyin.

MQRO_COA_WITH_FULL_DATA

Bu, özgün iletideki tüm uygulama iletisi verilerinin rapor iletisine eklenmesi dışında MQRO_COA ile aynıdır.

Birden çok MQRO_COA, MQRO_COA_WITH_DATA ve MQRO_COA_WITH_FULL_DATA belirtmeyin.

Teslimde doğrulama seçenekleri: Teslimde doğrulama raporu iletisi istemek için listelenen seçeneklerden birini belirleyin.

MQRO_COD

Bu tip bir rapor, bir uygulama iletisi kuyruktan silecek şekilde hedef kuyruktan alındığında kuyruk yöneticisi tarafından oluşturulur. Özgün iletideki ileti verileri rapor iletisine dahil edilmez.

İleti bir iş biriminin parçası olarak alınır, rapor iletisi aynı iş birimi içinde oluşturulur; böylece, iş birimi kesinleştirilinceye kadar rapor kullanılamaz. İş birimi geri çekilirse, rapor gönderilmez.

MQGMO_MARK_SKIP_BACKOUT seçeneğiyle bir ileti alınır, her zaman COD raporu oluşturulmaz. Birincil iş birimi geriletildiyse, ancak ikincil iş birimi kesinleştirildiyse, ileti kuyruktan kaldırılır, ancak COD raporu oluşturulmaz.

İleti tanımlayıcıdaki *Format* alanı MQFMT_DEAD_LETTER_HEADER ise COD raporu oluşturulmaz. Bu, ileti teslim edilemezse ve teslim edilmeyen bir kuyruğa konursa bir COD raporunun oluşturulmasını önler.

Hedef kuyruk bir XCF kuyruğuysa MQRO_COD geçerli değildir.

Birden çok MQRO_COD, MQRO_COD_WITH_DATA ve MQRO_COD_WITH_FULL_DATA belirtmeyin.

MQRO_COD_WITH_DATA

Bu, MQRO_COD ile aynıdır; ancak, özgün iletideki uygulama iletisi verilerinin ilk 100 baytı rapor iletisine eklenir. Özgün ileti bir ya da daha fazla MQ üstbilgi yapısı içeriyorsa, bunlar 100 baytlık uygulama verilerine ek olarak rapor iletisine eklenir.

Özgün iletiye ilişkin MQGET çağrısında MQGMO_ACCEPT_TRUNCATED_MSG belirtilirse ve alınan ileti kesilirse, rapor iletisine yerleştirilen uygulama iletisi verilerinin miktarı ortama bağlıdır:

- z/OS' da en az:
 - Özgün iletinin uzunluğu
 - İletiyi almak için kullanılan arabelleğin uzunluğu
 - 100 bayt.
- Diğer ortamlarda, en az aşağıdakiler geçerlidir:
 - Özgün iletinin uzunluğu
 - 100 bayt.

Hedef kuyruk bir XCF kuyruğuysa MQRO_COD_WITH_DATA geçerli değildir.

Birden çok MQRO_COD, MQRO_COD_WITH_DATA ve MQRO_COD_WITH_FULL_DATA belirtmeyin.

MQRO_COD_WITH_FULL_DATA

Bu, özgün iletideki tüm uygulama iletisi verilerinin rapor iletisine eklenmesi dışında MQRO_COD ile aynıdır.

Hedef kuyruk bir XCF kuyruğuysa MQRO_COD_WITH_FULL_DATA geçerli değildir.

Birden çok MQRO_COD, MQRO_COD_WITH_DATA ve MQRO_COD_WITH_FULL_DATA belirtmeyin.

Action-notification options(Eylem-bildirim seçenekleri): Ayan uygulamanın pozitif eylem ya da negatif eylem rapor iletisi göndermesini istemek için listelenen seçeneklerden birini ya da her ikisini belirtin.

MQRO_PAN

Bu tip bir rapor, iletisi alan ve üzerinde işlem yapan uygulama tarafından oluşturulur. İletide istenen işlemin başarıyla gerçekleştirildiğini gösterir. Raporu oluşturan uygulama, herhangi bir verinin rapora dahil edilip edilmeyeceğini belirler.

Bu isteğin iletiyi alan uygulamaya iletilmesi dışında, kuyruk yöneticisi bu seçeneğe dayalı olarak herhangi bir işlem gerçekleştirmez. Alma uygulaması, uygunsu raporu oluşturmalıdır.

MQRO_NAN

Bu tip bir rapor, iletiyi alan ve üzerinde işlem yapan uygulama tarafından oluşturulur. İletide istenen işlemin başarıyla gerçekleştirilmediğini gösterir. Raporu oluşturan uygulama, herhangi bir verinin rapora dahil edilip edilmeyeceğini belirler. Örneğin, isteğin neden gerçekleştirilemediğini gösteren bazı veriler eklemek isteyebilirsiniz.

Bu isteğin iletiyi alan uygulamaya iletilmesi dışında, kuyruk yöneticisi bu seçeneğe dayalı olarak herhangi bir işlem gerçekleştirmez. Alma uygulaması, uygunsu raporu oluşturmalıdır.

Uygulama, hangi koşulların pozitif bir işleme karşılık geldiğini ve hangilerinin negatif bir işleme karşılık geldiğini belirlemelidir. Ancak, istek yalnızca kısmen gerçekleştirildiyse, istenirse PAN raporu yerine bir NAN raporu oluşturun. Her olası koşul, pozitif bir işleme ya da negatif bir işleme karşılık gelmelidir, ancak her ikisine birden karşılık gelmemelidir.

İleti tanıtıcısı seçenekleri: Rapor iletisinin (ya da yanıt iletisinin) *MsgId* ögesinin nasıl ayarlanacağını denetlemek için listelenen seçeneklerden birini belirleyin.

MQRO_NEW_MSG_ID

Bu varsayılan eylemdir ve bu iletinin sonucu olarak bir rapor ya da yanıt oluşturulursa, rapor ya da yanıt iletisi için yeni bir *MsgId* oluşturulur.

MQRO_PASS_MSG_ID

Bu iletinin sonucu olarak bir rapor ya da yanıt oluşturulursa, bu iletinin *MsgId* 'i raporun ya da yanıt iletisinin *MsgId* ' ine kopyalanır.

Bir yayın iletisinin *MsgId* , yayının bir kopyasını alan her abone için farklı olacaktır ve bu nedenle rapora ya da yanıt iletisine kopyalanan *MsgId* her biri için farklı olacaktır.

Bu seçenek belirlenmezse, MQRO_NEW_MSG_ID varsayılr.

İlinti tanıtıcısı seçenekleri: Rapor iletisinin (ya da yanıt iletisinin) *CorrelId* ögesinin nasıl ayarlanacağını denetlemek için listelenen seçeneklerden birini belirtin.

MQRO_COPY_MSG_ID_TO_CORREL_ID

Bu varsayılan eylemdir ve bu iletinin sonucu olarak bir rapor ya da yanıt oluşturulursa, bu iletinin *MsgId* ögesinin raporun ya da yanıt iletisinin *CorrelId* hedefine kopyalandığını belirtir.

Bir yayın iletisinin *MsgId* ' i, yayının bir kopyasını alan her abone için farklı olacaktır ve bu nedenle *MsgId* raporun ya da yanıt iletisinin *CorrelId* içine kopyalanması her biri için farklı olacaktır.

MQRO_PASS_CORREL_ID

Bu iletinin sonucu olarak bir rapor ya da yanıt oluşturulursa, bu iletinin *CorrelId* 'i raporun ya da yanıt iletisinin *CorrelId* ' ine kopyalanır.

MQSO_SET_CORREL_ID seçeneğini kullanmadığı ve MQSD ' deki SubCorreltanıtıcı alanını MQCI_NONE olarak ayarlamadığı sürece, bir yayınlama iletisinin *CorrelId* değeri aboneye özgüdür. Bu nedenle, raporun ya da yanıt iletisinin *CorrelId* *CorrelId* içine kopyalanması her biri için farklı olabilir.

Bu seçenek belirtilmezse, MQRO_COPY_MSG_ID_TO_CORREL_ID varsayılr.

İstekleri yanıtlayan ya da rapor iletileri oluşturan sunucular, özgün iletide MQRO_PASS_MSG_ID ya da MQRO_PASS_CORREL_ID seçeneklerinin ayarlanıp ayarlanmadığını denetlemelidir. Varsa, sunucuların bu seçenekler için açıklanan işlemi gerçekleştirmeleri gerekir. İkisi de ayarlanmazsa, sunucuların karşılık gelen varsayılan işlemi gerçekleştirmesi gerekir.

Düzenleme seçenekleri: Hedef kuyruğa teslim edilemediğinde özgün iletinin yok edilmesini denetlemek için listelenen seçeneklerden birini belirleyin. Uygulama, kural dışı durum raporları istemesinden bağımsız olarak yok etme seçeneklerini ayarlayabilir.

MQRO_DEAD_LETTER_Q

Bu, varsayılan işlemdir ve ileti hedef kuyruğa teslim edilemezse, iletiyi teslim edilmeyen ileti kuyruğuna yerleştirir. Bu durum aşağıdaki durumlarda oluşur:

- Özgün iletiyi koyan uygulamaya, MQPUT ya da MQPUT1 çağrısının döndürdüğü neden koduyla sorunu zamanuyumlu olarak bildirilemezse. Gönderen tarafından istendiye, bir kural dışı durum rapor iletilisi oluşturulur.
- Özgün iletiyi koyan uygulama bir konuya yerleştirildiğinde

MQRO_DISCARD_MSG

Bu, hedef kuyruğa teslim edilemezse iletiyi atar. Bu durum aşağıdaki durumlarda oluşur:

- Özgün iletiyi koyan uygulamaya, MQPUT ya da MQPUT1 çağrısının döndürdüğü neden koduyla sorunu zamanuyumlu olarak bildirilemezse. Gönderen tarafından istendiye, bir kural dışı durum rapor iletilisi oluşturulur.
- Özgün iletiyi koyan uygulama bir konuya yerleştirildiğinde

Özgün iletiyi gönderene geri döndürmek istiyorsanız, özgün ileti gönderilmeyen iletiler kuyruğuna yerleştirilmeden, gönderenin MQRO_DISCARD_MSG ögesini MQRO_EXCEPTION_WITH_FULL_DATA ile belirtmesi gerekir.

MQRO_PASS_DISCARD_AND_EXPIRY

Bu seçenek bir iletide ayarlanırsa ve bu nedenle bir rapor veya yanıt oluşturulursa, raporun ileti tanımlayıcısı şunları devralır:

- Ayarlandıysa MQRO_DISCARD_MSG.
- İletin kalan süre bitimi (bu bir süre bitimi raporu değilse). Bu bir süre bitimi raporuysa, süre bitimi 60 saniye olarak ayarlanır.

Etkinlik seçeneği

MQRO_ACTIVITY

Bu değerin kullanılması, **herhangi bir** iletisinin rotasının bir kuyruk yöneticisi ağı boyunca izlenmesini sağlar. Rapor seçeneği, herhangi bir geçerli kullanıcı iletisinde belirtilebilir ve anında ağ üzerinden iletinin rotasını hesaplamaya başlamanızı sağlar.

İletiyi oluşturan uygulama, etkinlik raporu oluşturmayı etkinleştiremezse, raporlama, kuyruk yöneticisi yöneticileri tarafından sağlanan bir API geçiş çıkışı kullanılarak etkinleştirilebilir.

Not:

1. Ağda etkinlik raporları oluşturabilecek kuyruk yöneticileri ne kadar az olursa, rota o kadar az ayrıntılı olur.
2. Etkinlik raporlarının, alınan rotayı belirlemek için doğru sırayla yerleştirilmesi zor olabilir.
3. Etkinlik raporları, istenen hedeflerine giden bir rota bulamayabilir.
4. Bu rapor seçeneği ayarlanmış olan iletiler, seçeneği anlamasalar bile, herhangi bir kuyruk yöneticisi tarafından kabul edilmelidir. Bu, IBM WebSphere MQ 6.0 dışı ya da daha sonraki bir kuyruk yöneticisi tarafından işlense bile, rapor seçeneğinin herhangi bir kullanıcı iletisinde ayarlanmasına olanak sağlar.
5. Kuyruk yöneticisi ya da kullanıcı süreci olan bir süreç, bu seçenek ayarlanarak bir ileti üzerinde etkinlik gerçekleştiriyorsa, bir etkinlik raporu oluşturmayı ve yerleştirmeyi seçebilir.

Varsayılan seçenek: Rapor seçeneği gerekmiyorsa aşağıdakileri belirtin:

MQRO_NONE

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın. MQRO_NONE, program belgelerine yardımcı olmak için tanımlandı. Bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Genel bilgiler:

1. Gerekli tüm rapor tipleri, özgün iletiyi gönderen uygulama tarafından özellikle istenmelidir. Örneğin, bir COA raporu istenir, ancak bir kural dışı durum raporu istenmezse, ileti hedef kuyruğa yerleştirildiğinde bir COA raporu oluşturulur, ancak ileti oraya geldiğinde hedef kuyruk doluysa kural dışı durum raporu oluşturulmaz. *Report* seçeneği ayarlanmazsa, kuyruk yöneticisi ya da ileti kanalı aracısı (MCA) tarafından rapor iletilisi oluşturulmaz.

Yerel kuyruk yöneticisi bunları tanımasa da bazı rapor seçenekleri belirtilebilir; bu, seçenek *hedef* kuyruk yöneticisi tarafından işlendiğinde kullanışlıdır. Daha fazla ayrıntı için bkz. "Rapor seçenekleri ve ileti işaretleri" sayfa 891 .

Bir rapor ileti istenirse, raporun gönderileceği kuyruğun adı *ReplyToQ* alanında belirtilmelidir. Bir rapor ileti alındığında, raporun türü, ileti tanımlayıcısındaki *Feedback* alanı incelenerek belirlenebilir.

2. Bir rapor ileti oluşturan kuyruk yöneticisi ya da MCA, rapor iletilisini yanıt kuyruğuna koyamazsa (örneğin, yanıt kuyruğu ya da iletim kuyruğu dolu olduğu için), rapor ileti gönderilmeyen iletiler kuyruğuna yerleştirilir. Bu *da* başarısız olursa ya da gitmeyen iletiler kuyruğu yoksa, yapılan işlem rapor iletilisinin tipine bağlıdır:

- Rapor ileti bir kural dışı durum raporuysa, kural dışı durum raporunu oluşturan ileti iletim kuyruğunda bırakılır; bu, iletinin kaybolmamasını sağlar.
- Diğer tüm rapor tipleri için rapor ileti atılır ve işleme normal şekilde devam eder. Bu, özgün iletinin güvenli bir şekilde teslim edilmesi (COA veya COD rapor iletileri için) ya da artık herhangi bir ilgi alanı olmaması (bir süre sonu rapor ileti için) nedeniyle yapılır.

Bir rapor ileti bir kuyruğa başarıyla yerleştirildiğinde (hedef kuyruk ya da bir ara iletim kuyruğu), ileti artık özel işleme konusu olmaz; ileti, diğer herhangi bir ileti gibi işlenir.

3. Rapor oluşturulduğunda, *ReplyToQ* kuyruğu açılır ve aşağıdaki durumlar dışında, iletiye ilişkin MQMD ' de *UserIdentifier* yetkisi kullanılarak rapor ileti konarak rapora neden olur:

- Alan MCA tarafından oluşturulan kural dışı durum raporları, MCA ' nın rapora neden olan iletiyi yerleştirmeye çalışırken kullandığı herhangi bir yetkiyle birlikte kullanılır.
- Kuyruk yöneticisi tarafından oluşturulan COA raporları, rapora neden olan ileti, raporu oluşturan kuyruk yöneticisine yerleştirildiğinde kullanılan herhangi bir yetkiyle yerleştirilir. Örneğin, ileti MCA 'nın kullanıcı kimliği kullanılarak bir alıcı MCA tarafından konduysa, kuyruk yöneticisi COA raporunu MCA' nın kullanıcı kimliğini kullanarak koyar.

Rapor oluşturan uygulamalar, yanıt oluşturmak için kullandıkları yetkiyi kullanmalıdır; bu genellikle özgün iletidaki kullanıcı kimliğinin yetkisidir.

Raporun uzak bir hedefe gitmesi gerekirse, gönderenler ve alıcılar, diğer iletilerde olduğu gibi raporun kabul edilip edilmeyeceğine karar verebilir.

4. Veri içeren bir rapor ileti istenirse:

- Rapor ileti, her zaman özgün iletiyi gönderen tarafından istenen veri miktarıyla oluşturulur. Rapor ileti yanıt kuyruğu için çok büyükse, yukarıda açıklanan işlem gerçekleşir; rapor ileti hiçbir zaman yanıt kuyruğuna sığacak şekilde kesilmez.
- Özgün iletinin *Format* değeri MQFMT_XMIT_Q_HEADER ise, raporda yer alan veriler MQXQH ' yi içermez. Rapor verileri, özgün iletidaki MQXQH ' nin ötesindeki verilerin ilk byte 'ıyla başlar. Bu, kuyruğun bir iletim kuyruğu olup olmadığını belirler.

5. Yanıt kuyruğunda bir COA, COD ya da süre bitim raporu ileti alınır, özgün iletinin geldiği, teslim edildiği ya da süresinin dolduğu garanti edilir. Ancak, bu rapor iletilerinden biri ya da daha fazlası istenirse ve alınmazsa, aşağıdakilerden biri olduğundan tersi kabul edilemez:

- a. Bir bağlantı kapalı olduğundan rapor ileti tutuldu.
- b. Rapor ileti, bir ara iletim kuyruğunda ya da yanıt kuyruğunda (örneğin, kuyruk dolu ya da girişler için engellenmiş) bir engelleme koşulu bulunduğundan gecikir.
- c. Rapor ileti, gönderilmeyen iletiler kuyruğunda.
- d. Kuyruk yöneticisi rapor iletilisini oluşturmayı denediğinde, iletiyi ne uygun kuyruğa ne de gitmeyen iletiler kuyruğuna koyamadı, bu nedenle rapor ileti oluşturulamadı.
- e. Bildirilmekte olan işlem (varış, teslim ya da süre bitimi) ile ilgili rapor iletilisinin oluşturulması arasında kuyruk yöneticisinde bir hata oluştu. (COD rapor ileti aynı iş birimi içinde oluşturulduğundan, uygulama bir iş birimi içindeki özgün iletiyi alırsa, COD rapor iletileri için bu gerçekleşmez.)

Kural dışı durum raporu iletileri, yukarıdaki 1, 2 ve 3 nedenlerle aynı şekilde tutulabilirler. Ancak, bir MCA bir kural dışı durum raporu iletileri oluşturamazsa (rapor iletileri yanıt kuyruğuna ya da gönderilmeyen iletiler kuyruğuna konamaz), özgün iletileri gönderenin iletilim kuyruğunda kalır ve kanal kapanır. Bu durum, rapor iletilerinin kanalın gönderildiği yerde mi, yoksa gönderildiği yerde mi oluşturulacağına bakılmaksızın ortaya çıkar.

6. Özgün iletileri geçici olarak engellenirse (bir kural dışı durum raporu iletileri oluşturulmasıyla ve özgün iletileri bir teslim edilmeyen iletileri kuyruğuna konmasıyla sonuçlanıyorsa), ancak engelleme temizlenir ve bir uygulama özgün iletileri teslim edilmeyen iletileri kuyruğundan okur ve yeniden hedefine yerleştirirse, aşağıdaki durum oluşabilir:
 - Bir kural dışı durum raporu iletileri oluşturulmasına rağmen, özgün iletileri sonunda hedefine başarıyla ulaşır.
 - Özgün iletileri daha sonra başka bir engelle karşılaşabileceği için, tek bir özgün iletileri göre birden fazla kural dışı durum raporu iletileri oluşturulur.

Bir konuya yerleştirilirken iletileri raporla:

1. Bir konuya iletileri yerleştirilirken raporlar oluşturulabilir. Bu iletileri, konunun tüm abonelerine gönderilir; bu durum sıfır, bir ya da daha fazla olabilir. Sonuç olarak çok sayıda rapor iletileri oluşturulabileceği için rapor seçeneklerini kullanmayı seçerken bu dikkate alınmalıdır.
2. Bir konuya iletileri koyarken, iletilerinin bir kopyasının verileceği birçok hedef kuyruk olabilir. Bu hedef kuyruklardan bazıları kuyruk dolu gibi bir sorun varsa, MQPUT ' nin başarıyla tamamlanması NPMGDLV ya da PMSGDLV ayarına bağlıdır (iletilerinin kalıcı olmasına bağlı olarak). Ayar, hedef kuyruğa iletileri teslimi başarılı olacaksa (örneğin, kalıcı bir aboneye kalıcı bir iletileridir ve PMSGDLV için ALL ya da ALLDUR değeri belirlenmişse), başarı aşağıdaki ölçütlerden biri olarak tanımlanır:
 - Abone kuyruğuna başarıyla yerleştirilmesi
 - Abone kuyruğu iletileri alamazsa MQRO_DEAD_LETTER_Q ve Ölü Harf kuyruğuna başarıyla bir şekilde konması
 - Abone kuyruğu iletileri alamazsa MQRO_DISCARD_MSG kullanılır.

İletileri bölümleri için rapor iletileri:

1. Bölümlenmeye izin verilen iletiler için rapor iletileri istenebilir (MQMF_SEGMENTATION_ALLOWED işaretilerinin tanımına bakın). Kuyruk yöneticisi iletileri bölümlere ayırmayı gerekli bulursa, daha sonra ilgili koşullarla karşılaşan her bölüm için bir rapor iletileri oluşturulabilir. Başvurular, istenen her rapor iletileri tipi için birden çok rapor iletileri almak üzere hazırlanmalıdır. Birden çok raporu özgün iletilerinin grup tanıtıcısıyla ilişkilendirmek için rapor iletilerindeki *GroupId* alanını kullanın ve *Feedback* alanı her bir rapor iletilerinin tipini tanımlar.
2. Bölümlere ilişkin rapor iletilerini almak için MQGMO_LOGICAL_ORDER kullanılıyorsa, ardışık MQGET çağrıları tarafından *farklı tipte* raporların döndürülebileceğini unutmayın. Örneğin, kuyruk yöneticisi tarafından bölümlenmiş bir iletileri için hem COA hem de COD raporları istenirse, rapor iletilerine ilişkin MQGET çağrıları, tahmin edilemeyen bir şekilde serpiştirilmiş COA ve COD rapor iletilerini döndürebilir. MQGMO_COMPLETE_MSG seçeneğini kullanarak bunu önleyin (isteğe bağlı olarak MQGMO_ACCEPT_TRUNCATED_MSG ile). MQGMO_COMPLETE_MSG, kuyruk yöneticisinin aynı rapor tipine sahip rapor iletilerini yeniden birleştirmesine neden olur. Örneğin, ilk MQGET çağrısı özgün iletileriyle ilgili tüm COA iletilerini yeniden birleştirip, ikinci MQGET çağrısı tüm COD iletilerini yeniden birleştirebilir. İlk olarak yeniden birleştirilen iletileri tipi, kuyruktaki ilk olarak hangi tipte rapor iletilerinin oluşacağına bağlıdır.
3. Kendi bölümlerini yerleştirdikleri uygulamalar, her bir bölüm için farklı rapor seçenekleri belirtebilir. Ancak, aşağıdaki noktalara dikkat edin:
 - Kesimler MQGMO_COMPLETE_MSG seçeneği kullanılarak alınır, kuyruk yöneticisi yalnızca *birinci* bölümdeki rapor seçeneklerini kullanır.
 - Bölümler birer birer alınır ve çoğu MQRO_COD_* seçeneklerinden birine sahipse, ancak en az bir bölüm sahip değilse, rapor iletilerini tek bir MQGET çağrısıyla almak için MQGMO_COMPLETE_MSG seçeneğini kullanamazsınız ya da tüm rapor iletilerinin ne zaman geldiğini saptamak için MQGMO_ALL_SEGMENTS_KULLANILABİLİR seçeneğini kullanamazsınız.

4. Bir MQ ağında kuyruk yöneticileri farklı yeteneklere sahip olabilir. Bölümlenmeyi desteklemeyen bir kuyruk yöneticisi ya da MCA tarafından bir bölüm için rapor iletisi oluşturulursa, kuyruk yöneticisi ya da MCA varsayılan olarak rapor iletisine gerekli bölüm bilgilerini içermez ve bu, raporun oluşturulmasına neden olan özgün iletinin saptanmasını zorlaştırabilir. Uygun MQRO_*_WITH_DATA ya da MQRO_*_WITH_FULL_DATA seçeneklerini belirterek, rapor iletisiyle veri isteyerek bu zorlukları önleyin. Ancak, MQRO_*_WITH_DATA belirtilirse, rapor iletisi bölümlenmeyi desteklemeyen bir kuyruk yöneticisi ya da MCA tarafından oluşturulduysa, rapor iletisini alan uygulamaya *100 bayttan az* uygulama iletisi döndürülebileceğini unutmayın.

Bir rapor iletisi için ileti tanımlayıcısının içeriği: Kuyruk yöneticisi ya da ileti kanalı aracısı (MCA) bir rapor iletisi oluşturduğunda, ileti tanımlayıcısındaki alanları aşağıdaki değerlere ayarlar ve iletiyi normal şekilde yerleştirir.

Çizelge 501. Bir rapor iletisi sistem tarafından oluşturulduğunda MQMD alanları için kullanılan değerler

MQMD 'deki Alan	Kullanılan değer
<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>	Raporun türü için uygun olduğu şekilde (MQFB_COA, MQFB_COD, MQFB_EXPIRATION ya da MQRC_* değeri)
<i>Encoding</i>	Özgün ileti tanımlayıcısından kopyalandı
<i>CodedCharSetId</i>	Özgün ileti tanımlayıcısından kopyalandı
<i>Format</i>	Özgün ileti tanımlayıcısından kopyalandı
<i>Priority</i>	Özgün ileti tanımlayıcısından kopyalandı
<i>Persistence</i>	Özgün ileti tanımlayıcısından kopyalandı
<i>MsgId</i>	Özgün ileti tanımlayıcısındaki rapor seçenekleriyle belirtildiği gibi
<i>CorrelId</i>	Özgün ileti tanımlayıcısındaki rapor seçenekleriyle belirtildiği gibi
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	Boşluklar
<i>ReplyToQMGr</i>	Kuyruk yöneticisinin adı
<i>UserIdentifier</i>	MQPMO_PASS_IDENTITY_CONTEXT seçeneği tarafından ayarlandığı gibi
<i>AccountingToken</i>	MQPMO_PASS_IDENTITY_CONTEXT seçeneği tarafından ayarlandığı gibi
<i>AppIdentityData</i>	MQPMO_PASS_IDENTITY_CONTEXT seçeneği tarafından ayarlandığı gibi
<i>PutAppType</i>	MQAT_QMGR ya da ileti kanalı aracısı için uygun olduğu şekilde
<i>PutAppName</i>	Kuyruk yöneticisi adının ya da ileti kanalı aracısı adının ilk 28 baytı. IMS köprüsü tarafından oluşturulan rapor iletileri için bu alan, iletinin ilgili olduğu IMS sisteminin XCF grup adını ve XCF üye adını içerir.
<i>PutDate</i>	Rapor iletisinin gönderildiği tarih
<i>PutTime</i>	Rapor iletisinin gönderildiği zaman
<i>AppOriginData</i>	Boşluklar
<i>GroupId</i>	Özgün ileti tanımlayıcısından kopyalandı
<i>MsgSeqNumber</i>	Özgün ileti tanımlayıcısından kopyalandı

Çizelge 501. Bir rapor iletisi sistem tarafından oluşturulduğunda MQMD alanları için kullanılan değerler (devamı var)

MQMD 'deki Alan	Kullanılan değer
<i>Offset</i>	Özgün ileti tanımlayıcısından kopyalandı
<i>MsgFlags</i>	Özgün ileti tanımlayıcısından kopyalandı
<i>OriginalLength</i>	MQOL_UNDEFINED değilse özgün ileti tanımlayıcısından kopyalanır ve özgün ileti verilerinin uzunluğuna ayarlanır; tersi durumda,

Rapor oluşturan bir uygulamanın, aşağıdakiler dışında benzer değerler ayarlaması önerilir:

- *ReplyToQMGr* alanı boş olarak ayarlanabilir (kuyruk yöneticisi, ileti yerleştirildiğinde bunu yerel kuyruk yöneticisinin adıyla değiştirir).
- Normal olarak MQPMO_PASS_IDENTITY_CONTEXT yanıt için kullanılacak seçeneği kullanarak bağlam alanlarını ayarlayın.

Rapor alanının çözümlenmesi: *Report* alanı alt alanlar içerir; bu nedenle, belirli bir raporu gönderen kişinin “Rapor alanı çözümleniyor” sayfa 892’inde açıklanan tekniklerden birini kullanması gerekir gerekmediğini denetlemesi gereken uygulamalar.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri MQRO_NONE.

MQMD için MsgType (MQLONG)

Bu, iletinin tipini gösterir. İleti tipleri aşağıdaki gibi gruplanır:

MQMT_SYSTEM_FIRST

Sistem tarafından tanımlanan ileti tipleri için en düşük değer.

MQMT_SYSTEM_LAST

Sistem tarafından tanımlanan ileti tipleri için en yüksek değer.

Sistem aralığı içinde tanımlı olan değerler şunlardır:

MQMT_DATAGRAM

İleti, yanıt gerektirmeyen bir ileti.

MQMT_REQUEST

İleti, yanıt gerektiren bir iletidir.

ReplyToQ alanında, yanıtın gönderileceği kuyruğun adını belirleyin. *Report* alanı, yanıtın *MsgId* ve *CorrelId* değerinin nasıl ayarlanacağını gösterir.

MQMT_REPLY

İleti, daha önceki bir istek iletisine (MQMT_REQUEST) verilen yanıttır. İletinin, istek iletisinin *ReplyToQ* alanı tarafından belirtilen kuyruğa gönderilmesi gerekir. Yanıtın *MsgId* ve *CorrelId* ayarlarının nasıl yapılacağını denetlemek için isteğin *Report* alanını kullanın.

Not: Kuyruk yöneticisi, istek-yanıt ilişkisini uygulamaz; bu bir uygulama sorumluluğudur.

MQMT_REPORT

İleti, genellikle başka bir iletiyle ilgili, beklenen ya da beklenmeyen bir oluşumu bildiriyor (örneğin, geçerli olmayan veriler içeren bir istek iletisi alındı). İletiyi, özgün iletinin ileti tanımlayıcısının *ReplyToQ* alanı tarafından belirtilen kuyruğa gönderin. Raporun niteliğini belirtmek için *Feedback* alanlarını ayarlayın. Rapor iletisinin *MsgId* ve *CorrelId* ayarlarının nasıl yapılacağını denetlemek için özgün iletinin *Report* alanını kullanın.

Kuyruk yöneticisi ya da ileti kanalı aracı tarafından oluşturulan rapor iletileri, yukarıda açıklandığı gibi *Feedback* ve *CorrelId* alanları ayarlanmış olarak her zaman *ReplyToQ* kuyruğuna gönderilir.

Uygulama tanımlı değerler de kullanılabilir. Bunlar aşağıdaki aralık içinde olmalıdır:

MQMT_APPL_FIRST

Uygulama tanımlı ileti tipleri için en düşük değer.

MQMT_APPL_LAST

Uygulama tanımlı ileti tipleri için en yüksek değer.

MQPUT ve MQPUT1 çağrılarını için, *MsgType* değeri sistem tanımlı aralık içinde ya da uygulama tanımlı aralık içinde olmalıdır; değilse, çağrı başarısız olur; neden kodu MQRC_MSG_TYPE_ERROR.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri MQMT_DATAGRAM.

MQMD için süre bitimi (MQLONG)

Bu, iletiyi koyan uygulama tarafından belirlenen, saniyenin onda biri olarak ifade edilen bir zaman dönemidir. İleti, bu süre dolmadan önce hedef kuyruktan kaldırılmamışsa atılmaya uygun hale gelir.

Örneğin, süre bitimi için bir dakika ayarlamak için **MQMD**değerini ayarlamamız gerekir.**Expiry** -600.

Değer, iletinin hedef kuyruğa geçirdiği süreyi ve uzak bir kuyruğa yerleştirilmesi durumunda, herhangi bir ara iletim kuyruğunda harcadığı süreyi yansıtabilecek şekilde azaltılır. Bunlar önemliyse, iletim sürelerini yansıtmak için ileti kanalı ajanları tarafından da azaltılabilir. Benzer şekilde, bu iletiyi başka bir kuyruğa ileten bir uygulama, iletiyi önemli bir süre alıkoysaydı, gerekirse değeri azaltabilir. Ancak, süre bitim süresi yaklaşık olarak kabul edilir ve değerinin küçük zaman aralıklarını yansıtmak için azaltılmasına gerek yoktur.

İleti MQGET çağrısıyla bir uygulama tarafından alındığında, *Expiry* alanı hala devam eden süre bitimini gösterir.

Bir iletinin süre bitim tarihi geçtikten sonra, kuyruk yöneticisi tarafından atılmaya uygun hale gelir. İletiyi süresi dolmamış olarak döndüren bir göz atma ya da göz atmama MQGET çağrısı oluşturduğunda ileti atılır. Örneğin, MQGMO ' da *MatchOptions* alanı MQMO_NONE olarak ayarlanmış, FIFO sıralı bir kuyruktan okunan, süresi dolan ilk iletiye kadar olan tüm iletileri atar. Öncelikli sıralı bir kuyruğa, aynı çağrı, süresi dolan daha yüksek öncelikli iletileri ve süresi dolmamış ilk iletimden önce kuyruğa gelen aynı önceliğe sahip iletileri atar.

Süresi dolmuş bir ileti bir uygulamaya hiçbir zaman döndürülmez (göz atma ya da göz atma olmayan bir MQGET çağrısıyla); bu nedenle, başarılı bir MQGET çağrısından sonra ileti tanımlayıcısının *Expiry* alanındaki değer sıfırdan büyük ya da MQEI_UNLIMITED özel değeri.

Bir ileti uzak kuyruğa konursa, iletinin süresi, ileti hedef kuyruğa ulaşmadan önce, ileti bir ara iletim kuyruğundayken sona erebilir (ve atılabilir).

İleti MQRO_EXPIRATION_ * rapor seçeneklerinden birini belirtmişse, süresi dolan bir ileti atıldığında bir rapor oluşturulur. Bu seçeneklerden hiçbiri belirtilmezse, böyle bir rapor oluşturulmaz; iletinin bu zaman döneminden sonra artık ilgili olmadığı varsayılır (belki de daha sonraki bir iletinin yerine geçmesinden dolayı).

Eşitleme noktası içine konan bir ileti için, süre bitimi aralığı, eşitleme noktasının kesinleştirildiği zamanda değil, ileti konduğunda başlar. Süre bitimi aralığı, eşitleme noktası kesinleştirilmeden önce geçebilir. Bu durumda, ileti kesinleştirme işleminden sonra bir zaman atılır ve MQGET işlemine yanıt olarak bir uygulamaya döndürülmez.

İletileri süre bitimine dayalı olarak gönderen diğer herhangi bir program, istenirse uygun bir rapor iletilisi de göndermelidir.

Notlar:

1. *Expiry* saat değeri sıfır ya da 999 999 999 'dan büyük bir sayıyla bir ileti konursa, MQPUT ya da MQPUT1 çağrısı MQRC_EXPIRY_ERROR neden koduyla başarısız olur; bu durumda rapor iletilisi oluşturulmaz.

MQRC_EXPIRY_ERROR neden kodunu etkinleştirmek için AMQ_ENFORCE_MAX_EXPIRY_ERROR ortam değişkenini etkinleştirmeniz gerekir.

Aşağıda Linux için bir örnek kullanılmıştır:

```
$ export AMQ_ENFORCE_MAX_EXPIRY_ERROR=True
```

Aşağıdakileri göz önünde bulundurun:

- Önemli olan değişkeni dışa aktarmak.
 - Gerçek değer yoksayılar, ancak kuruluş gözden geçirilirken True değerinin kullanılması yararlı olabilir.
2. Süresi dolmuş bir ileti daha sonraya kadar atılamayabileceğinden, kuyruқта süre bitimine kadar geçen ve bu nedenle alınmaya uygun olmayan iletiler olabilir. Yine de bu iletiler, derinlik tetikleme de dahil olmak üzere, tüm amaçlar için kuyruktaki ileti sayısına doğru sayılır.
Bir abone/tüketici (istemci) bir iletiyi almayı denirse ve iletinin süresi dolmuşsa, ileti çok eski olduğu için atıldığı için istemci hiçbir şey almaz. Ayrıca, istemci herhangi bir hata ileti almayacaktır.
 3. Bir süre bitim raporu, istenirse, ileti atıldığında değil, atıldığında oluşturulur.
 4. Süresi dolan bir iletinin atılması ve istenirse bir süre sonu raporu oluşturulması, ileti bir iş birimi içinde çalışan bir MQGET çağrısının sonucu olarak atılmak üzere zamanlanmış olsa da, hiçbir zaman uygulamanın iş biriminin bir parçası değildir.
 5. Süresi dolmuş bir ileti bir iş birimi içindeki bir MQGET çağrısı tarafından alınır ve iş birimi daha sonra geriletilirse, ileti yeniden alınmadan önce atılabilir.
 6. Süresi dolmuş bir ileti MQGMO_LOCK ile bir MQGET çağrısı tarafından kilitlenirse, ileti MQGMO_MSG_UNDER_CURSOR; neden kodu MQRC_NO_MSG_UNDER_CURSOR ile MQGET çağrısıyla alınmadan önce atılabilir; bu durumda, sonraki MQGET çağrısına ilişkin MQRC_NO_MSG_UNDER_CURSOR döndürülür.
 7. Süre bitimi sıfırdan büyük olan bir istek ileti alındığında, uygulama yanıt ileti gönderirken aşağıdaki işlemlerden birini gerçekleştirebilir:
 - İstek ileti yanıt ileti kalan süre bitimini kopyalayın.
 - Yanıt ileti süresi bitimini sıfırdan büyük belirtik bir değere ayarlayın.
 - Yanıt ileti süresi bitimini MQEI_UNLIMITED olarak ayarlayın.Yapılacak işlem, uygulamanın tasarımına bağlıdır. Ancak, iletilerin teslim edilmeyen (teslim edilmeyen) bir kuyruğa konmasına ilişkin varsayılan işlem, iletinin kalan süre bitimini korumak ve azaltmaya devam etmek olmalıdır.
 8. Tetikleyici iletileri her zaman MQEI_UNLIMITED ile oluşturulur.
 9. *Format* adı MQFMT_XMIT_Q_HEADER olan bir iletide (olağan olarak bir iletim kuyruğunda) MQXQH içinde ikinci bir ileti tanımlayıcısı vardır. Bu nedenle, kendisiyle ilişkilendirilmiş iki *Expiry* alanı vardır. Bu durumda aşağıdaki ek puanlar dikkate alınmalıdır:
 - Bir uygulama uzak kuyruğa bir ileti koyduğunda, kuyruk yöneticisi ilk olarak iletiyi yerel bir iletim kuyruğuna yerleştirir ve uygulama ileti verilerinin başına MQXQH yapısını ekler. Kuyruk yöneticisi, iki *Expiry* alanının değerlerini uygulama tarafından belirtilenle aynı olacak şekilde ayarlar.
Bir uygulama bir iletiyi doğrudan yerel iletim kuyruğuna koyarsa, ileti verileri bir MQXQH yapısıyla başlamalı ve biçim adı MQFMT_XMIT_Q_HEADER olmalıdır. Bu durumda, uygulamanın bu iki *Expiry* alanının değerlerini aynı olacak şekilde ayarlamasına gerek yoktur. (Kuyruk yöneticisi, MQXQH içindeki *Expiry* alanının geçerli bir değer içerdiğini ve ileti verilerinin bunu içerecek kadar uzun olduğunu denetler.) Doğrudan iletim kuyruğuna yazabilen bir uygulama için, uygulamanın yerleşik ileti tanımlayıcısıyla bir iletim kuyruğu üstbilgisi yaratması gerekir. Ancak, iletim kuyruğuna yazılan ileti tanımlayıcısındaki süre bitimi değeri, yerleşik ileti tanımlayıcısındaki değerle tutarsızsa, süre bitimi hatası reddi oluşur.
 - *Format* adı MQFMT_XMIT_Q_HEADER olan bir ileti kuyruktan alındığında (bu olağan ya da iletim kuyruğu olsun), kuyruk yöneticisi kuyruқта bekleme süresi kadar *her iki* bu *Expiry* alanı da azalır. İleti verileri *Expiry* alanını MQXQH ' ye dahil edecek kadar uzun değilse hata oluşmaz.
 - Kuyruk yöneticisi, iletinin atılmaya uygun olup olmadığını sınamak için ayrı ileti tanımlayıcısında (MQXQH yapısına gömülü ileti tanımlayıcısında değil) *Expiry* alanını kullanır.

- İki *Expiry* alanının başlangıç değerleri farklıysa, ileti alındığında ayrı ileti tanımlayıcısındaki *Expiry* süresi sıfırdan büyük olabilir (bu nedenle ileti atılamaz), ancak MQXQH 'deki *Expiry* alanına göre süre geçmiştir. Bu durumda MQXQH 'deki *Expiry* alanı sıfır olarak ayarlanır.

10. MQIIH 'nin İşaretler alanında MQIIH_PASS_EXPIRATION ayarlanmadıkça, IMS köprüsünden döndürülen bir yanıt iletisindeki süre sonu sınırsız olur. Ek bilgi için [İşaretler](#) konusuna bakın.

Şu özel değer tanınır:

MQEI_UNLIMITED

İletinin sınırsız süre bitimi var.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş alanıdır. Bu alanın ilk değeri MQEI_UNLIMITED.

z/OS üzerinde süresi dolan iletiler

IBM MQ for z/OS üzerinde, süresi dolan iletiler sonraki uygun MQGET çağrısı tarafından atılır.

Ancak, böyle bir çağrı gerçekleşmezse, süresi dolan ileti atılmaz ve bazı kuyruklar için çok sayıda süresi dolan ileti birikebilir. Bunu düzeltmek için, kuyruk yöneticisini belirli aralıklarla kuyrukları taraması ve bir ya da daha çok kuyruktaki süresi dolan iletileri aşağıdaki yollardan biriyle atması için ayarlayın:

Periyodik tarama

EXPRYINT (süre bitimi aralığı) kuyruk yöneticisi özniteliğini kullanarak bir dönem belirtebilirsiniz. Süre bitim aralığına her ulaşıldığında, kuyruk yöneticisi süresi dolan iletileri atmak için taranmaya değer aday kuyrukları arar.

Kuyruk yöneticisi, her kuyrukta süresi dolan iletilere ilişkin bilgileri saklar ve süresi dolan iletilere ilişkin taramanın yararlı olup olmadığını bilir. Bu nedenle, herhangi bir zamanda yalnızca bir kuyruk seçimi taranır.

Paylaşılan kuyruklar, bir kuyruk paylaşım grubunda yalnızca bir kuyruk yöneticisi tarafından taranır. Genellikle, yeniden başlatılır ilk kuyruk yöneticisi ya da EXPRYINT ayarlı ilk kuyruk yöneticisidir. Bu kuyruk yöneticisi sonlandırılırsa, kuyruk paylaşım grubundaki başka bir kuyruk yöneticisi kuyruk taramasını devralır. Bir kuyruk paylaşım grubu içindeki tüm kuyruk yöneticileri için süre bitim aralığı değerini aynı değere ayarlayın.


EXPRYINT ayarından bağımsız olarak, bir kuyruk yöneticisi yeniden başlatıldığında her kuyruk için süre bitimi işleminin gerçekleştiğini unutmayın.



Belirtik istek

Taranmasını istediğiniz kuyruğu ya da kuyrukları belirterek REFRESH QMGR TYPE (EXPIRY) komutunu verin.

Daha düşük süre bitim sürelerini uygulama

Yöneticiler, kuyruktaki ya da konudaki **CUSTOM** özniteliğinde belirtilen **CAPEXPY** özniteliğini kullanarak bir kuyruğa ya da konuya konan herhangi bir iletinin süre bitim süresini sınırlayabilir.

Önemli:  Havuzun tamamı z/OS üzerindeyse, IBM MQ 9.3.1 içinde sunulan **CAPEXPY** özniteliğini bir kümeyle birlikte kullanamazsınız.

  IBM MQ 9.3.1' den yöneticiler, bir kuyruğun ya da konunun **CAPEXPY** özniteliğini **CUSTOM** özniteliğinde belirtmesine gerek kalmadan kullanarak süre bitim süresini sınırlayabilir. Bir kuyruk ya da konu zaten **CUSTOM** özniteliği içinde ayarlanmış **CAPEXPY** özniteliğine sahipse, yeni **CAPEXPY** özniteliği değiştirilmeden önce **CUSTOM** özniteliğinden **CAPEXPY** özniteliğini kaldırmanız gerekir. Bunu tek bir komutta yapabilirsiniz, örneğin:

```
ALTER QLOCAL(Q1) CAPEXPY(1000) CUSTOM('')
```

Not: Bir nesneyi ürünün önceki bir sürümünden geçirdiğinizde, **CAPEXPY** değeriniz NOLIMIT varsayılan değerine ayarlanır. **CAPEXPY** özniteliği **CUSTOM** özniteliğinin içinde ayarlandıysa, bu seçenek öncelikli olur.

Yeni **CAPEXPY** özniteliğini kullanmak istiyorsanız, önce **CUSTOM** özniteliğinden **CAPEXPY** özniteliğini kaldırmanız gerekir. Her iki seçeneği de ayarlamak işe yaramıyor.

Multi **V 9.3.1** Doğrudan doğruya kuyruklarda ya da konularda ayarlanan (**CUSTOM** özniteliği içinde değil) düzeltilen **CAPEXPY** özniteliği kümelenmiş bir özniteliktir. Bir küme kuyruğunun tüm eşgörünümlerinin **CAPEXPY** öznitelikleri için aynı değeri kullanması gerektiğini unutmayın. İletim kuyruğunda **CAPEXPY** değeri belirlendiyse ve değer küme kuyruğunun **CAPEXPY** özniteliğinden düşükse, bir iletim kuyruğunun iletinin süre bitim süresini kısaltması yine de olasıdır.

MQMD ' nin **Expiry** alanında bir uygulama tarafından belirtilen ve kuyrukta ya da konuda belirtilen **CAPEXPY** değerinden büyük olan süre bitimi, o **CAPEXPY** değeriyle değiştirilir. Bir uygulama tarafından belirtilen ve **CAPEXPY** değerinden daha düşük bir süre bitimi kullanılacak.

CAPEXPY değerinin saniyenin onda biri olarak ifade edildiğini, bu nedenle bir dakika değerinin 600 olduğunu unutmayın.

Çözüm yolunda birden çok nesne kullanılırsa (örneğin, bir ileti bir diğer ada ya da uzak kuyruğa konduğunda), ileti süre bitimi üst sınırı olarak tüm **CAPEXPY** değerlerinin en küçüğü kullanılır.

CAPEXPY değerlerinde yapılan değişiklikler hemen yürürlüğe girer. Süre bitim değeri, bir kuyruğa ya da konuya konan her değer için değerlendirilir ve bu nedenle, her bir koyma işlemi arasında farklılık gösterebilecek nesne çözümüne duyarlıdır.

Ancak, **CAPEXPY** içindeki bir değişiklikten önce kuyrukta var olan iletilerin değişiklikten etkilenmediğini unutmayın (yani, süre bitimi değişmeden kalır). Yalnızca **CAPEXPY** içindeki değişiklikten sonra kuyruğa konan yeni iletiler yeni süre bitimine sahip olur.

MQOO_BIND_NOT_FIXED ile açılan bir kuyruğa koyma işlemi gerçekleştirildiği bir kümede, iletilere, iletiyi seçilen hedef kuyruk yöneticisine gönderen kanal tarafından kullanılan iletim kuyruğu için ayarlanan **CAPEXPY** değerine bağlı olarak, her bir koymada farklı süre bitimi değerleri atanabilir.

Teslim gecikmesi, hedef kuyruk ya da konu için çözülen süre sona erme süresini aşıyorsa, MQRC_EXPIRY_ERROR ile bir JMS uygulaması tarafından kuyruğa ya da konuya konmanın başarısız olduğunu unutmayın. Bir JMS hedefi için bir kuyrukta çözümlenen **CAPEXPY** özniteliği bu hataya neden olabilir.

Not: CAPEXPY , herhangi bir SYSTEM.CLUSTER.* gibi dahili olarak oluşturulan iletileri bulunduracak IBM MQ kuyruklarında kullanılmamalıdır. kuyruğunu ve SYSTEM.PROTECTION.POLICY.QUEUE.

İlgili başvurular

[DEFINE kuyrukları](#)

[KONU TANIM](#)

MQMD için geribildirim (MQLONG)

Feedback alanı, raporun niteliğini belirtmek için MQMT_REPORT tipinde bir iletiyle birlikte kullanılır ve yalnızca o ileti tipiyle anlamlıdır.

Alan MQFB_* değerlerinden birini ya da MQRC_* değerlerinden birini içerebilir. Geribildirim kodları aşağıdaki gibi gruplanır:

MQFB_NONE

Geribildirim sağlanmadı.

MQFB_SYSTEM_FIRST

Sistem tarafından oluşturulan geribildirim için en düşük değer.

MQFB_SYSTEM_LAST

Sistem tarafından oluşturulan geribildirim için en yüksek değer.

MQFB_SYSTEM_FIRST ile MQFB_SYSTEM_LAST arasında sistem tarafından oluşturulan geribildirim kodları aralığı, bu konuda listelenen genel geribildirim kodlarını (MQFB_*) ve ileti hedef kuyruğa konamadığında oluşabilecek neden kodlarını (MQRC_*) içerir.

MQFB_APPL_FIRST

Uygulama tarafından oluşturulan geribildirim için en düşük değer.

MQFB_APPL_LAST

Uygulama tarafından oluşturulan geribildirim için en yüksek değer.

Rapor iletileri oluşturan uygulamalar, kuyruk yöneticisi ya da ileti kanalı aracısı tarafından oluşturulan rapor iletilerinin benzetimini yapmak istemedikçe, sistem aralığındaki (MQFB_QUIT dışında) geribildirim kodlarını kullanmamalıdır.

MQPUT ya da MQPUT1 çağrılarında, belirtilen değer MQFB_NONE ya da sistem aralığı ya da uygulama aralığı içinde olmalıdır. Bu, *MsgType* değeri ne olursa olsun denetlenir.

Genel geribildirim kodları:

MQFB_COA

Hedef kuyruğa varış onayı (bkz. MQRO_COA).

MQFB_COD

Alan uygulamaya teslim onayı (bkz. MQRO_COD).

MQFB_SÜRE SONU

İleti, süresi dolmadan önce hedef kuyruktan kaldırılmadığı için atıldı.

MQFB_PAN

Pozitif işlem bildirim (bkz. MQRO_PAN).

MQFB_NAN

Negatif işlem bildirim (bkz. MQRO_NAN).

MQFB_QUIT

Uygulamayı sona erdir.

Bu, çalışmakta olan bir uygulama programının eşgörünümlerinin sayısını denetlemek için bir iş yükü zamanlama programı tarafından kullanılabilir. Bu geribildirim koduyla bir MQMT_REPORT iletilerinin uygulama programının bir eşgörünümüne gönderilmesi, o eşgörünümüne işlemeyi durdurması gerektiğini belirtir. Ancak, bu kurala uymak uygulama için bir sorundur; kuyruk yöneticisi tarafından uygulanmaz.

Kanal geribildirim kodları:

MQFB_CHANNEL_COMPLETED

Bir kanal normal şekilde sona erdi.

MQFB_CHANNEL_FAIL

Bir kanal olağandışı sona erdi ve DURDURULDU durumuna geçti.

MQFB_CHANNEL_FAIL_RETRY

Bir kanal olağandışı sona erdi ve RETRY durumuna girdi.

IMS-bridge geribildirim kodları

Bu kodlar, beklenmeyen bir IMS-OTMA algılama kodu alındığında kullanılır. Algılama kodu 0x1A olduğunda, algılama koduyla ilişkili neden kodu *Geribildirim*'de gösterilir.

1. MQFB_IMS_FIRST (300)-MQFB_IMS_LAST (399) aralığındaki *Geribildirim* kodları için 0x1A dışında bir algılama kodu alındı. *algılama kodu* ifadesi tarafından verilir (*Geribildirim* - MQFB_IMS_FIRST+1)
2. MQFB_IMS_NACK_1A_REASON_FIRST (600)- MQFB_IMS_NACK_1A_REASON_LAST (855) aralığındaki *Geribildirim* kodları için 0x1A algılama kodu alındı. Algılama koduyla ilişkili *neden kodu* ifade tarafından verilir (*Geribildirim* - MQFB_IMS_NACK_1A_REASON_FIRST)

IMS-OTMA algılama kodlarının ve ilgili neden kodlarının anlamı *Open Transaction Manager Access Guide and Reference* adlı belgede açıklanmıştır.

IMS köprüsü tarafından aşağıdaki geribildirim kodları oluşturulabilir:

MQFB_DATA_LENGTH_ZERO

İletinin uygulama verilerinde bir kesim uzunluğu sıfırdı.

MQFB_DATA_LENGTH_NEGATIF

İletinin uygulama verilerinde bir kesim uzunluğu negatif.

MQFB_DATA_LENGTH_TOO_BIG

İletinin uygulama verilerinde kesim uzunluğu çok fazla.

MQFB_BUFFER_OVERFLOW

Uzunluk alanlarından birinin değeri, verilerin ileti arabelleğine taşmasına neden olur.

MQFB_LENGTH_OFF_BY_ONE

Uzunluk alanlarından birinin değeri 1 byte çok kısa.

MQFB_IIH_HATA

MQMD 'deki *Format* alanı MQFMT_IMS' yi belirtiyor, ancak ileti geçerli bir MQIIH yapısıyla başlamıyor.

MQFB_NOT_AUTHORIZED_FOR_IMS

İleti tanımlayıcı MQMD ' de bulunan kullanıcı kimliği ya da MQIIH yapısındaki *Authenticator* alanında bulunan parola, IMS köprüsü tarafından gerçekleştirilen geçerlilik denetiminde başarısız oldu. Sonuç olarak, ileti IMS' e iletilmedi.

MQFB_IMS_HATASI

IMS beklenmeyen bir hata döndürdü. Hatayla ilgili daha fazla bilgi için IMS köprüsünün bulunduğu sistemdeki IBM MQ hata günlüğüne bakın.

MQFB_IMS_FIRST

IMS-OTMA algılama kodu 0x1A değilse, IMStarafından oluşturulan geribildirim kodları MQFB_IMS_LAST (399) ile MQFB_IMS_FIRST (300) aralığındadır. IMS-OTMA algılama kodunun kendisi *Feedback* eksi MQFB_IMS_ERROR.

MQFB_IMS_SON

Algılama kodu 0x1A olmadığında IMStarafından oluşturulan geribildirim için en yüksek değer.

MQFB_IMS_NACK_1A_REASON_FIRST

Algılama kodu 0x1A olduğunda, IMStarafından oluşturulan geribildirim kodları MQFB_IMS_NACK_1A_REASON_FIRST (600)- MQFB_IMS_NACK_1A_REASON_LAST (855) aralığındadır.

MQFB_IMS_NACK_1A_REASON_LAST

Algılama kodu 0x1A olduğunda IMStarafından oluşturulan geribildirim için en yüksek değer

CICS-bridge geribildirim kodları: CICS bridge tarafından aşağıdaki geribildirim kodları oluşturulabilir:

MQFB_CICS_APPL_ABENDED

İletide belirtilen uygulama programı olağandışı sona erdi. Bu geribildirim kodu yalnızca MQDLH yapısının *Reason* alanında oluşur.

MQFB_CICS_APPL_NOT_STARTED

İletide belirtilen uygulama programına ilişkin EXEC CICS LINK başarısız oldu. Bu geribildirim kodu yalnızca MQDLH yapısının *Reason* alanında oluşur.

MQFB_CICS_BRIDGE_FAILURE

CICS bridge olağan hata işlemeyi tamamlamadan olağandışı bitti.

MQFB_CICS_CCSID_HATA

Karakter takımı tanıtıcısı geçersiz.

MQFB_CICS_CIH_HATASI

CICS bilgi üstbilgisi yapısı eksik ya da geçersiz.

MQFB_CICS_COMMAREA_HATA

CICS COMMAREA uzunluğu geçersiz.

MQFB_CICS_CORREL_ID_HATA

İlinti tanıtıcısı geçersiz.

MQFB_CICS_DLQ_HATA

CICS bridge görevi, bu isteğin yanıtını teslim edilmeyen iletiler kuyruğuna kopyalayamadı. İstek geri çekildi.

MQFB_CICS_ENCODING_ERROR

Kodlama geçerli değil.

MQFB_CICS_INTERNAL_HATA

CICS bridge beklenmeyen bir hatayla karşılaştı.

Bu geribildirim kodu yalnızca MQDLH yapısının *Reason* alanında oluşur.

MQFB_CICS_NOT_YETKILI

Kullanıcı kimliği yetkili değil ya da parola geçersiz.

Bu geribildirim kodu yalnızca MQDLH yapısının *Reason* alanında oluşur.

MQFB_CICS_UOW_BACKED_OUT

Aşağıdaki nedenlerden biri nedeniyle iş birimi geriletildi:

- Aynı iş birimi içindeki başka bir istek işlenirken bir hata saptandı.
- İş birimi devam ederken bir CICS olağandışı sonu oluştu.

MQFB_CICS_UOW_HATASI

İş birimi denetim alanı *UOWControl* geçersiz.

İzleme yönlendirmeli ileti geribildirim kodları:**MQFB_ACTIVITY**

Etkinlik raporlarını izleyen kullanıcı verileri seçeneğine izin vermek için MQFMT_EMBEDDED_PCF biçimiyle kullanılır.

MQFB_MAX_ACTIVITIES

İletin dahil olduğu etkinlik sayısı etkinlik üst sınırını aştığından izlemeli rota iletisi atıldığında döndürülür.

MQFB_NOT_ILETILDI

İzleme rotası iletisi, izleme rotası iletilerini desteklemeyen bir uzak kuyruk yöneticisine gönderilmek üzere olduğu için atıldığında döndürülür.

MQFB_NOT_TESLIM EDİLDİ

İzleme rotası iletisi yerel bir kuyruğa konmak üzere olduğu için atıldığında döndürülür.

MQFB_UNSUPPORTED_FORWARDING

İletme parametresindeki bir değer tanınmadığı ve reddedilen bit maskesinde olduğu için izleme-rotası iletisi atıldığında döndürülür.

MQFB_UNSUPPORTED_DELIVERY

Teslim parametresindeki bir değer tanınmadığı ve reddedilen bit maskesinde olduğu için izleme-rotası iletisi atıldığında döndürülür.

IBM MQ neden kodları: Kural dışı durum raporu iletileri için, *Feedback* bir IBM MQ neden kodu içerir. Olası neden kodları şunlardır:

MQRC_PUT_INENGELLEME

(2051, X'803 ') Kuyruk için çağrılarını engelleme.

MQRC_Q_FULL

(2053, X'805 ') Kuyruk zaten ileti sayısı üst sınırını içeriyor.

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_Q_SPACE_NOT_KULLANILAMIYOR

(2056, X'808 ') Diskte kuyruk için kullanılabilir alan yok.

MQRC_PERSISTENT_NOT_ALLOWED

(2048, X'800 ') Kuyruk kalıcı iletileri desteklemiyor.

MQRC_MSG_TOO_BIG_FOR_Q_MGR

(2031, X'7EF') Kuyruk yöneticisi için ileti uzunluğu üst sınırından fazla.

MQRC_MSG_TOO_BIG_FOR_Q

(2030, X'7EE') İleti uzunluğu kuyruk için üst sınırdan fazla.

Neden kodlarının tam listesi için bkz:

- IBM MQ for z/OS için bkz. [API tamamlama ve neden kodları](#).

- Diğer tüm platformlar için bkz. [API tamamlama ve neden kodları](#).

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri MQFB_NONE.

MQMD için kodlama (MQLONG)

Bu, iletideki sayısal verilerin sayısal kodlamasını belirtir; MQMD yapısındaki sayısal veriler için geçerli değildir. Sayısal kodlama, ikili tamsayılar, paketlenmiş ondalık tamsayılar ve kayan noktalı sayılar için kullanılan gösterimi tanımlar.

z/OS' ta, Encoding alanının ikili tamsayı kısmı, karşılık gelen karakter kümesi tanıttıcısı, karakter kümesinin gösteriminin ikili tamsayılar için kullanılan kodlamaya bağlı olduğunu gösterdiğinde, ileti gövdesindeki karakter verilerinin tamsayı kodlamasını belirtmek için de kullanılır. Bu yalnızca belirli çok baytlık karakter kümelerini etkiler (örneğin, UTF-16 karakter kümeleri).

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Kuyruk yöneticisi, alanın geçerli olup olmadığını denetlemez. Aşağıdaki özel değer tanımlandı:

MQENC_NATIVE

Kodlama, uygulamanın çalıştığı programlama dili ve makine için varsayılan değerdir.

Not: Bu değişmez değeri, programlama diline ve çevreye bağlıdır. Bu nedenle, uygulamaların, uygulamanın çalışacağı ortama uygun üstbilgi, makro, COPY ya da INCLUDE dosyaları kullanılarak derlenmesi gerekir.

İletileri koyan uygulamalar genellikle MQENC_NATIVE değerini belirtir. İletileri alan uygulamalar bu alanı MQENC_NATIVE değeriyle karşılaştırmalıdır; değerler farklıysa, uygulamanın iletideki sayısal verileri dönüştürmesi gerekebilir. Kuyruk yöneticisinden iletiyi MQGET çağrısının işlenmesinin bir parçası olarak dönüştürmesini istemek için MQGMO_CONVERT seçeneğini kullanın. Encoding alanının nasıl oluşturulduğuna ilişkin ayrıntılar için bkz. "Makine kodlamaları" sayfa 888 .

MQGET çağrısında MQGMO_CONVERT seçeneğini belirtirseniz, bu alan bir giriş/çıkış alanıdır. Uygulama tarafından belirtilen değer, gerekirse ileti verilerinin dönüştürülecek kodlamadır. Dönüştürme başarılı ya da gereksizse, değer değişmez. Dönüştürme başarısız olursa, MQGET çağrısından sonraki değer, uygulamaya döndürülen dönüştürülmemiş iletinin kodlamasını gösterir.

Diğer durumlarda, bu alan MQGET çağrısına ilişkin bir çıkış alanıdır ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri MQENC_NATIVE.

MQMD için CodedCharSetId (MQLONG)

Bu alan, ileti gövdesi içindeki karakter verilerinin karakter kümesi tanıttıcısını belirtir.

Not: MQMD ve çağrılarda değiştirge olan diğer MQ veri yapılarındaki karakter verileri, kuyruk yöneticisinin karakter kümesinde olmalıdır. Bu, kuyruk yöneticisinin **CodedCharSetId** özneliği tarafından tanımlanır; bu özneliğin ayrıntıları için bkz. "Kuyruk yöneticisine ilişkin öznelikler" sayfa 789 .

Seçeneklerde MQGMO_CONVERT ile MQGET çağrılırken bu alan MQCCSI_Q_MGR olarak ayarlanırsa, davranış istemci ve sunucu uygulamaları arasında farklıdır. Sunucu uygulamalarında, karakter dönüştürme için kullanılan kod sayfası kuyruk yöneticisinin CodedCharSetId kod sayfasıdır; istemci uygulamaları için, karakter dönüştürme için kullanılan kod sayfası yürürlükteki ülke değeri kod sayfasıdır.

İstemci uygulamaları için, kuyruk yöneticisindeki yerine istemcinin ülke değerine dayalı olarak MQCCSI_Q_MGR doldurulur. Bu kuralın kural dışı durumu, bir IMS köprü kuyruğuna ileti koyduğunuzda ortaya çıktı; MQMD ' nin CodedCharSetId alanında döndürülen, kuyruk yöneticisinin CCSID 'sidir.

Aşağıdaki özel değeri kullanmamalısınız:

MQCCSI_APPL

Bu, MQMD ' nin CodedCharSetId alanında yanlış bir değerle sonuçlanır ve MQRC_SOURCE_CCSD_ERROR (ya da z/OS için MQRC_FORMAT_ERROR) dönüş koduna neden olur. İleti, MQGMO_CONVERT seçeneğiyle MQGET çağrısı kullanılarak alındığında.

Aşağıdaki özel değerleri kullanabilirsiniz:

MQCCSI_Q_MGR

İletideki karakter verileri kuyruk yöneticisinin karakter takımındaki karakterdir.

MQPUT ve MQPUT1 çağrılarında, kuyruk yöneticisi, iletiyle birlikte kuyruk yöneticisinin gerçek karakter kümesi tanıtıcısına gönderilen MQMD ' de bu değeri değiştirir. Sonuç olarak, MQGET çağrısı MQCCSI_Q_MGR değerini hiçbir zaman döndürmez.

MQCCSI_DEFAULT

String alanındaki verilerin CodedCharSetId değeri, MQCFH yapısından önce gelen üstbilgi yapısındaki CodedCharSetId alanı tarafından ya da MQCFH iletinin başlangıcında ise MQMD ' deki CodedCharSetId alanı tarafından tanımlanır.

MQCCSI_INHERIT

İletideki karakter verileri bu yapıyla aynı karakter kümesinde; bu, kuyruk yöneticisinin karakter kümesidir. (Yalnızca MQMD için, MQCCSI_INHERIT MQCCSI_Q_MGR ile aynı anlama sahiptir).

Kuyruk yöneticisi, iletiyle birlikte gönderilen MQMD 'deki bu değeri MQMD' nin gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı MQCCSI_INHERIT değerini döndürmez.

MQMD ' deki PutApp1Type alanının değeri MQAT_BROKER ise, MQCSI_INHERIT kullanmayın.

MQCCSI_EMBEDDED

İletideki karakter verileri, ileti verilerinin kendisinde bulunan tanıtıcıya sahip bir karakter kümesinde bulunur. Verilerin farklı bölümlerine uygulanarak, ileti verilerinin içine yerleştirilmiş herhangi bir sayıda karakter kümesi tanıtıcısı olabilir. Bu değer, karakter kümeleri karışımında veri içeren PCF iletileri (MQFMT_ADMIN, MQFMT_EVENT ya da MQFMT_PCF biçimiyle) için kullanılmalıdır. PCF iletilerinde bulunan her MQCFST, MQCFSL ve MQCFSF yapısının, MQCCSI_DEFAULT değil, belirtilmiş bir karakter kümesi tanıtıcısı olmalıdır.

MQFMT_EMBEDDED_PCF biçimindeki bir ileti karakter kümelerinin karışımında veri içermek içinse, MQCCSI_EMBEDDED kullanmayın. Bunun yerine, MQEPH yapısındaki İşaretler alanında MQEPH_CSID_EMBEDDED ögesini ayarlayın. Bu, önceki yapıda MQCCSI_EMBEDDED ayarının eşdeğeridir. Bu durumda, PCF iletilerinde bulunan her MQCFST, MQCFSL ve MQCFSF yapısının, MQCCSI_DEFAULT değil, belirtik bir karakter kümesi tanıtıcısı belirtilmiş olmalıdır. MQEPH yapısıyla ilgili daha fazla bilgi için bkz. [“MQEPH-Gömülü PCF üstbilgisi” sayfa 364.](#)

Bu değeri yalnızca MQPUT ve MQPUT1 çağrılarında belirtin. MQGET çağrısında belirtilirse, iletinin dönüştürülmesini engeller.

MQPUT ve MQPUT1 çağrılarında, kuyruk yöneticisi yukarıda açıklandığı gibi iletiyle birlikte gönderilen MQMD 'de MQCCSI_Q_MGR ve MQCSI_INHERIT değerlerini değiştirir, ancak MQPUT ya da MQPUT1 çağrısında belirtilen MQMD' yi değiştirmez. Belirtilen değer üzerinde başka bir denetim gerçekleştirilmez.

İletileri alan uygulamalar bu alanı uygulamanın beklediği değerle karşılaştırmalıdır; değerler farklıysa, uygulamanın iletideki karakter verilerini dönüştürmesi gerekebilir.

z/OSüzerinde, MQMD 'nin [Encoding](#) alanı, ileti gövdesindeki karakter verilerinin tamsayı kodlamasını belirtmek için kullanılır; MQMD' nin CodedCharSetId alanı, karakter kümesinin gösteriminin ikili tamsayılar için kullanılan kodlamaya bağımlı olduğunu belirtir. [Açık Çoklu platformlar](#): Karakter verilerinin byte sırası, kuyruk yöneticisinin çalıştığı platform için yerel tamsayı kodlamasıyla aynı olarak kabul edilir. Bu yalnızca belirli çok baytlık karakter kümelerini etkiler (örneğin, UTF-16 karakter kümeleri).

MQGET çağrısında MQGMO_CONVERT seçeneğini belirtirseniz, bu alan bir giriş/çıkış alanıdır. Uygulama tarafından belirlenen değer, gerekirse ileti verilerinin dönüştürülecek kodlanmış karakter takımı tanıtıcısıdır. Dönüştürme başarılı ya da gereksizse, değer değişmez (MQCCSI_Q_MGR ya da MQCSI_INHERIT değerinin gerçek değere dönüştürülmesi dışında). Dönüştürme başarısız olursa, MQGET çağrısından sonraki değer, uygulamaya döndürülen dönüştürülmemiş iletinin kodlanmış karakter kümesi tanıtıcısını gösterir.

Tersi durumda, bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri MQCCSI_Q_MGR 'dir.

MQMD için biçim (MQCHAR8)

Bu, iletiyi gönderen kişinin iletideki verilerin niteliğini alıcıya göstermek için kullandığı bir addır. Kuyruk yöneticisinin karakter takımındaki karakterler, ad için belirtilebilir, ancak adı aşağıdaki karakterlerle sınırlamanız gerekir:

- Büyük A-Z arası
- 0-9 arası sayısal rakamlar

Başka karakterler kullanılırsa, adı gönderen ve alan kuyruk yöneticilerinin karakter kümeleri arasında çevirmek mümkün olmayabilir.

Adı alan uzunluğuna kadar boşluklarla doldurun ya da alan sonundan önce adı sonlandırmak için boş bir karakter kullanın; boş değer ve sonraki karakterler boşluk olarak kabul edilir. Başında ya da içinde boşluk olan bir ad belirlemeyin. MQGET çağrısı için, kuyruk yöneticisi alanın uzunluğuna kadar boşluklarla doldurarak adı döndürür.

Kuyruk yöneticisi, adın yukarıda açıklanan önerilere uygun olup olmadığını denetlemez.

Büyük, küçük ve büyük harf karışık MQ başlayan adların, kuyruk yöneticisi tarafından tanımlanan anlamları vardır; kendi biçimleriniz için bu harflerle başlayan adları kullanmayın. Kuyruk yöneticisi yerleşik biçimleri şunlardır:

MQFMT_NONE

Verilerin türü tanımsız: İleti MQGMO_CONVERT seçeneği kullanılarak bir kuyruktan alındığında veriler dönüştürülemez.

MQGET çağrısında MQGMO_CONVERT belirtirseniz ve iletideki verilerin karakter kümesi ya da kodlaması **MsgDesc** değiştirgesinde belirtilenden farklıysa, ileti aşağıdaki tamamlanma ve neden kodlarıyla (başka hata olmadığı varsayılarak) döndürülür:

- MQFMT_NONE verileri iletinin başındaysa ise, tamamlama kodu MQCC_WARNING ve neden kodu MQRC_FORMAT_ERROR.
- MQFMT_NONE verileri iletinin sonundaysa (başında bir ya da daha çok MQ üstbilgi yapısı varsa), tamamlama kodu MQCC_OK ve neden kodu MQRC_NONE. MQ üstbilgi yapıları, bu durumda istenen karakter kümesine ve kodlamaya dönüştürülür.

C programlama dili için, MQFMT_NONE_ARRAY değişmezi de tanımlanır; bu değer MQFMT_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.


MQFMT_ADMIN

İleti, programlanabilir komut biçimindeki (PCF) bir komut sunucusu isteği ya da yanıt iletisidir. MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir. Programlanabilir komut biçimi iletilerinin kullanılmasına ilişkin ek bilgi için [Programlanabilir Komut Biçimlerinin Kullanılması](#) başlıklı konuya bakın.

C programlama dili için, MQFMT_ADMIN_ARRAY değişmezi de tanımlanır; bu değer MQFMT_ADMIN ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_CICS

İleti verileri, CICS bilgi üstbilgisi MQCIH ile başlar ve bunu uygulama verileri izler. Uygulama verilerinin biçim adı, MQCIH yapısındaki Format alanı tarafından verilir.

 z/OS üzerinde, MQFMT_CICS biçimindeki iletileri dönüştürmek için MQGET çağrısında MQGMO_CONVERT seçeneğini belirtin.

C programlama dili için, MQFMT_CICS_ARRAY değişmezi de tanımlanır; bu değer MQFMT_CICS ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_COMMAND_1

İleti, nesne sayısını, tamamlanma kodunu ve neden kodunu içeren bir MQSC komut sunucusu yanıt iletisidir. MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

C programlama dili için, MQFMT_COMMAND_1_ARRAY değişmezi de tanımlanır; bu değer MQFMT_COMMAND_1 ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_COMMAND_2

İleti, istenen nesnelere ilişkin bilgileri içeren bir MQSC komut sunucusu yanıt iletisidir. MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

C programlama dili için MQFMT_COMMAND_2_ARRAY değişmezi de tanımlanır; bu değer MQFMT_COMMAND_2 ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_DEAD_LETTER_HEADER

İleti verileri, giden harf üstbilgisi MQDLH ile başlar. Özgün iletideki veriler MQDLH yapısını hemen izler. Özgün ileti verilerinin biçim adı, MQDLH yapısındaki *Format* alanı tarafından verilir; bu yapının ayrıntıları için bkz. “MQDLH-Harf üstbilgisi” sayfa 352 . MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

MQFMT_DEAD_LETTER_HEADER *Format* içeren iletiler için COA ve COD raporları oluşturulmaz.

C programlama dili için, MQFMT_DEAD_LETTER_HEADER_ARRAY değişmezi de tanımlanır; bu değer MQFMT_DEAD_LETTER_HEADER ile aynı, ancak bir dizgi yerine bir karakter dizisidir.

MQFMT_DIST_HEADER

İleti verileri dağıtım listesi üstbilgisi MQDH ile başlar; bu, MQOR ve MQPMR kayıtlarının dizilerini içerir. Dağıtım listesi üstbilgisini ek veriler izleyebilir. MQDH yapısındaki *Format* alanı tarafından verilen ek verilerin biçimi (varsa); bu yapının ayrıntıları için bkz. “MQDH-Dağıtım üstbilgisi” sayfa 346 . MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, MQFMT_DIST_HEADER biçimindeki iletiler dönüştürülebilir.

Bu biçim aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

C programlama dili için, MQFMT_DIST_HEADER_ARRAY değişmezi de tanımlandı; bu değer MQFMT_DIST_HEADER ile aynı, ancak bir dizgi yerine bir karakter dizisidir.

MQFMT_EMBEDDED_PCF

PCF komutu değerinin MQCMD_TRACE_ROUTE olarak ayarlanması koşuluyla, izleme rotası iletisinin biçimi. Bu biçimin kullanılması, uygulamaların önceki PCF parametreleriyle başa çıkabilmesi koşuluyla, kullanıcı verilerinin izleme rotası iletisiyle birlikte gönderilmesini sağlar.

PCF üstbilgisinin ilk üstbilgi olması gerekir; tersi durumunda, ileti bir izleme rotası iletisi olarak kabul edilmez. Bu, iletinin bir grupta yer almayacağı ve izleme rotası iletilerinin bölümlenemediği anlamına gelir. Bir grupta bir izleme rotası iletisi gönderilirse, ileti MQRC_MSG_NOT_ALLOWED_IN_GROUP neden koduyla reddedilir.

MQFMT_ADMIN ' in bir izleme rotası iletisinin biçimi için de kullanılabilceğini, ancak bu durumda izleme rotası iletisiyle birlikte kullanıcı verilerinin gönderilebileceğini unutmayın.

MQFMT_EVENT

İleti, oluşan bir olayı bildiren bir MQ olay iletisidir. Olay iletileri programlanabilir komutlarla aynı yapıya sahiptir; bu yapıyla ilgili ek bilgi için [PCF komut iletileri](#) ve olaylarla ilgili bilgi için [Olay izleme başlıklı](#) konuya bakın.

MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, Version-1 olay iletileri tüm ortamlarda dönüştürülebilir. Version-2 olay iletileri yalnızca z/OS üzerinde dönüştürülebilir.

C programlama dili için, MQFMT_EVENT_ARRAY değişmezi de tanımlanır; bu değer MQFMT_EVENT ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_IMS

İleti verileri, uygulama verilerinin izlediği IMS bilgi üstbilgisi MQIIH ile başlar. Uygulama verilerinin biçim adı, MQIIH yapısındaki Format alanı tarafından verilir.

MQGMO_CONVERT ile MQGET kullanılırken MQIIH yapısının nasıl işlendiğine ilişkin ayrıntılar için bkz. [“MQIIH için biçim \(MQCHAR8\)” sayfa 409](#) ve [“MQIIH için ReplyToBiçimi \(MQCHAR8\)” sayfa 410](#).

C programlama dili için, MQFMT_IMS_ARRAY değişmezi de tanımlanır; bu değer MQFMT_IMS ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_IMS_VAR_STRING

İleti, 11zzccc biçiminde bir dizgi olan IMS değişken dizgisidir; burada:

11

IMS değişken dizgi ögesinin toplam uzunluğunu belirten 2 baytlık bir alandır. Bu uzunluk, 11 (2 bayt) uzunluğuna, zz (2 bayt) uzunluğuna ve karakter dizgisinin kendisinin uzunluğuna eşittir. 11 , Encoding alanı tarafından belirtilen kodlamada 2 baytlık bir ikili tamsayıdır.

zz

IMS için önemli olan işaretleri içeren 2 baytlık bir alandır. zz , iki MQBYTE alanından oluşan ve gönderenden alıcıya değişiklik yapılmadan iletilen bir byte dizgisidir (yani, zz herhangi bir dönüştürmeye tabi değildir).

ccc

11-4 karakterlerini içeren değişken uzunluklu bir karakter dizilimidir. ccc , CodedCharSetId alanı tarafından belirtilen karakter kümesinde bulunur.

z/OS işletim sistemlerinde, ileti verileri, her dizgi 11zzccc biçiminde olacak şekilde, bir araya getirilmiş IMS değişken dizgileri dizgisinden oluşabilir. Ardışık IMS değişken dizgileri arasında atlanan bayt olmamalıdır. Bu, birinci dizginin tek bir uzunluğu varsa, ikinci dizginin yanlış hizalanacağı, yani ikinin katı olan bir sınırla başlamayacağı anlamına gelir. Temel veri tiplerinin hizalanmasını gerektiren makinelerde bu tür dizgiler oluştururken dikkatli olun.

MQFMT_IMS_VAR_STRING biçimindeki iletileri dönüştürmek için MQGET çağrısında MQGMO_CONVERT seçeneğini kullanın.

C programlama dili için, MQFMT_IMS_VAR_STRING_ARRAY sabiti de tanımlanır; bu değer MQFMT_IMS_VAR_STRING ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_MD_EXTENSION

İleti verileri, ileti tanımlayıcı uzantısı MQMDE ile başlar ve isteğe bağlı olarak diğer veriler (genellikle uygulama iletileri verileri) tarafından izlenir. MQMDE 'yi izleyen verilerin biçim adı, karakter kümesi ve kodlaması, MQMDE' deki Format, CodedCharSetId ve Encoding alanları tarafından verilir. Bu yapının ayrıntıları için bkz. [“MQMDE-İleti tanımlayıcı uzantısı” sayfa 468](#) . MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

C programlama dili için, MQFMT_MD_EXTENSION_ARRAY değişmezi de tanımlanır; bu değer MQFMT_MD_EXTENSION ile aynı, ancak dizgi yerine bir karakter dizisidir.

MQFMT_PCF

İleti, programlanabilir komut biçimi (PCF) iletilerinin yapısına uyan kullanıcı tanımlı bir iletidir. MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir. Programlanabilir komut biçimi iletilerinin kullanılmasına ilişkin ek bilgi için [Programlanabilir Komut Biçimlerinin Kullanılması](#) başlıklı konuya bakın.

C programlama dili için, MQFMT_PCF_ARRAY değişmezi de tanımlanır; bu değer MQFMT_PCF ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_REF_MSG_HEADER

İleti verileri, başvuru iletileri üstbilgisi MQRMH ile başlar ve isteğe bağlı olarak diğer veriler tarafından takip edilir. Verilerin biçim adı, karakter kümesi ve kodlaması, MQRMH ' deki Format, CodedCharSetId ve Encoding alanları tarafından verilir. Bu yapının ayrıntıları için bkz. [“MQRMH-Başvuru iletileri üstbilgisi” sayfa 543](#) . MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

Bu biçim aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

C programlama dili için, MQFMT_REF_MSG_HEADER_ARRAY değişmezi de tanımlanır; bu değer MQFMT_REF_MSG_HEADER ile aynı, ancak bir dizgi yerine bir karakter dizisidir.

MQFMT_RF_HEADER

İleti verileri, kurallar ve biçimleme üstbilgisi MQRFH ile başlar ve isteğe bağlı olarak diğer veriler tarafından takip edilir. Verilerin biçim adı, karakter kümesi ve kodlaması (varsa), MQRFH 'deki Format, CodedCharSetIdve Encoding alanları tarafından verilir. MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

C programlama dili için, MQFMT_RF_HEADER_ARRAY değişmezi de tanımlanır; bu değer MQFMT_RF_HEADER ile aynıdır, ancak bir dizgi yerine bir karakter dizisidir.

MQFMT_RF_HEADER_2

İleti verileri, version-2 kuralları ve MQRFH2 üstbilgisiyle başlar ve isteğe bağlı olarak diğer veriler tarafından takip edilir. İsteğe bağlı verilerin biçim adı, karakter kümesi ve kodlaması (varsa), MQRFH2' deki Format, CodedCharSetIdve Encoding alanları tarafından verilir. MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

C programlama dili için, MQFMT_RF_HEADER_2_ARRAY değişmezi de tanımlanır; bu değer MQFMT_RF_HEADER_2 ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQFMT_STRING

Uygulama ileti verileri bir SBCS dizilimi (tek baytlık karakter takımı) ya da bir DBCS dizilimi (çift baytlık karakter takımı) olabilir. MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

C programlama dili için, MQFMT_STRING_ARRAY sabiti de tanımlanır; bu değer MQFMT_STRING ile aynıdır, ancak dizgi yerine bir karakter dizisidir.


MQFMT_TRIGGER

İleti, MQTM yapısı tarafından açıklanan bir tetikleyici iletidir; bu yapının ayrıntıları için bkz. “MQTM-Tetikleyici ileti” sayfa 593 . MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

C programlama dili için, MQFMT_TRIGGER_ARRAY değişmezi de tanımlanır; bu değer MQFMT_TRIGGER ile aynı değere sahiptir, ancak dizgi yerine bir karakter dizisidir.

MQFMT_WORK_INFO_HEADER

İleti verileri, iş bilgileri üstbilgisi MQWIH ile başlar ve bunu uygulama verileri izler. Uygulama verilerinin biçim adı, MQWIH yapısındaki Format alanı tarafından verilir.

 z/OS üzerinde, MQFMT_WORK_INFO_HEADER biçimindeki iletilerde kullanıcı verilerini dönüştürmek için MQGET çağrısında MQGMO_CONVERT seçeneğini belirtin. Ancak, MQWIH yapısının kendisi her zaman kuyruk yöneticisinin karakter kümesi ve kodlamasında döndürülür (yani, MQWIH yapısı MQGMO_CONVERT seçeneğinin belirtilip belirtilmediğine bakılmaksızın dönüştürülür).

C programlama dili için, MQFMT_WORK_INFO_HEADER_ARRAY değişmezi de tanımlandı; bu değer MQFMT_WORK_INFO_HEADER ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQFMT_XMIT_Q_HEADER

İleti verileri, iletim kuyruğu üstbilgisi MQXQH ile başlar. Özgün iletideki veriler MQXQH yapısını hemen izler. Özgün ileti verilerinin biçim adı, iletim kuyruğu üstbilgisi MQXQH 'nin bir parçası olan MQMD

yapısındaki Format alanı tarafından verilir. Bu yapının ayrıntıları için bkz. “MQXQH-İletim kuyruğu üstbilgisi” sayfa 612 .

COA and COD reports are not generated for messages that have a Format of MQFMT_XMIT_Q_HEADER.

C programlama dili için, MQFMT_XMIT_Q_HEADER_ARRAY değişmezi de tanımlanır; bu değer MQFMT_XMIT_Q_HEADER ile aynı, ancak bir dizgi yerine bir karakter dizisidir.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın uzunluğu MQ_FORMAT_LENGTH tarafından verilir. Bu alanın ilk değeri MQFMT_NONE.

MQMD için Öncelik (MQLONG)

MQPUT ve MQPUT1 çağrıları için değer sıfırdan büyük ya da sıfıra eşit olmalıdır; sıfır en düşük önceliktir. Aşağıdaki özel değer de kullanılabilir:

MQPRI_PRIORITY_AS_Q_DEF

- Kuyruk bir küme kuyruğuyorsa, iletinin önceliği, iletinin yerleştirildiği kuyruğun belirli eşgörünümüne sahip olan *hedef* kuyruk yöneticisinde tanımlanan **DefPriority** özniteliğinden alınır.

Küme kuyruğunun birden çok eşgörünümü varsa ve bu öznitelikte farklılık gösterdiklerinde, bunlardan birinin tane seçkullanım seçtiği değer tahmin edilemez. Bu nedenle, bu özniteliği tüm eşgörünümlerde aynı değere ayarlamalısınız. Böyle bir durumda değilse, kuyruk yöneticisi günlüklerine AMQ9407 hata iletisi gönderilir. Ayrıca bkz. [Diğer adlar, uzak kuyruklar ve küme kuyrukları için hedef nesne öznitelikleri nasıl çözülür?](#)

DefPriority değeri, ileti hedef kuyruğa yerleştirildiğinde *Priority* alanına kopyalanır. *DefPriority* daha sonra değiştirilirse, kuyruğa önceden yerleştirilen iletiler bundan etkilenmez.

- Kuyruk bir küme kuyruğu değilse, hedef kuyruk yöneticisi uzak olsa bile, iletinin önceliği *yerel* kuyruk yöneticisinde tanımlanan **DefPriority** özniteliğinden alınır.

Kuyruk adı çözüm yolunda birden çok tanımlama varsa, varsayılan öncelik, yoldaki *birinci* tanımlamada bu özniteliğin değerinden alınır. Bu şöyle olabilir:

- Diğer ad kuyruğu
- Yerel bir kuyruk
- Uzak kuyruğun yerel tanımlaması
- Kuyruk yöneticisi diğer adı
- Bir iletim kuyruğu (örneğin, *DefXmitQName* kuyruğu)

İleti konduğunda, *DefPriority* değeri *Priority* alanına kopyalanır. Daha sonra *DefPriority* değiştirilirse, önceden konmuş olan iletiler etkilenmez.

MQGET çağrısıyla döndürülen değer her zaman sıfırdan büyük ya da sıfıra eşittir; MQPRI_PRIORITY_AS_Q_DEF değeri hiçbir zaman döndürülmez.

Yerel kuyruk yöneticisi tarafından desteklenen üst sınırdan daha yüksek bir önceliğe sahip bir ileti konursa (bu üst sınır **MaxPriority** kuyruk yöneticisi özniteliği tarafından verilir), ileti kuyruk yöneticisi tarafından kabul edilir, ancak kuyruk yöneticisinin en yüksek önceliğine göre kuyruğa yerleştirilir; MQPUT ya da MQPUT1 çağrısı MQCC_WARNING ile tamamlanır ve neden kodu MQRC_PRIORITY_AŞIRI_MAXIMUM. Ancak, *Priority* alanı iletiyi koyan uygulama tarafından belirtilen değeri korur.

z/OSİletim sistemlerinde, MsgSeq1 numaralı bir ileti, ileti teslim sırası MQMDS_PRIORITY ve dizin tipi MQIT_GROUP_ID olan bir kuyruğa yerleştirilirse, kuyruk iletiyi farklı bir önceliğe sahip olarak kabul edebilir. İleti, öncelik düzeyi 0 ya da 1 olan kuyruğa yerleştirildiyse, öncelik değeri 2 gibi işlenir. Bunun nedeni, bu kuyruk tipine yerleştirilen iletilerin sırasının, verimli grup tamlığı sınamalarını etkinleştirmek için eniyilenmiş olmasıdır. MQMDS_PRIORITY ileti teslim sırası ve MQIT_GROUP_ID dizin tipi hakkında daha fazla bilgi için bkz. [MsgDeliverySequence özniteliği](#).

Bir iletiyi yanıtlarken, uygulamaların yanıt iletisine ilişkin istek iletisinin önceliğini kullanması gerekir. Diğer durumlarda, MQPRI_PRIORITY_AS_Q_DEF belirtilmesi, öncelik ayarlarının uygulama değiştirilmeden gerçekleştirilmesine olanak sağlar.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş alanıdır. Bu alanın ilk değeri: MQPRI_PRIORITY_AS_Q_DEF.

MQMD için kalıcılık (MQLONG)

Bu, iletinin sistem hatalarından kurtulup kurtulmadığını ve kuyruk yöneticisinin yeniden başlatılıp başlatılmadığını gösterir. MQPUT ve MQPUT1 çağrıları için değer aşağıdakilerden biri olmalıdır:

MQPER_KALICI

İleti, sistem hatalarından kurtulur ve kuyruk yöneticisini yeniden başlatır. İleti konduktan ve konduğu iş birimi kesinleştirildikten sonra (ileti bir iş biriminin parçası olarak konursa), ileti yardımcı saklama alanında korunur. İleti kuyruktan kaldırılıncaya ve alındığı iş birimi kesinleştirilinceye (ileti bir iş biriminin parçası olarak alındıysa) kadar orada kalır.

Uzak kuyruğa kalıcı bir ileti gönderildiğinde, saklama ve iletme mekanizması, iletinin bir sonraki kuyruk yöneticisine geldiği bilinceye kadar, hedef rotadaki her kuyruk yöneticisinde iletiyi tutar.

Kalıcı iletilerin yerleştirilemediği yer:

- Geçici dinamik kuyruklar
- CFLEVEL (2) ya da altındaki bir CFSTRUCT nesnesiyle eşleşen ya da CFSTRUCT nesnesinin RECOVER (NO) olarak tanımlandığı paylaşılan kuyruklar.

Kalıcı iletiler kalıcı dinamik kuyruklara ve önceden tanımlanmış kuyruklara yerleştirilebilir.

MQPER_NOT_KALICI

İleti genellikle sistem hatalarından kurtulmaz ya da kuyruk yöneticisi yeniden başlatılır. Bu, kuyruk yöneticisi yeniden başlatıldığında yardımcı saklama alanında iletinin bozulmamış bir kopyası bulunsa da geçerlidir.

NPMCLASS (HIGH) kuyrukları kalıcı olmayan iletilerde olağan bir kuyruk yöneticisi kapatma ve yeniden başlatma işlemi devam eder.

Paylaşılan kuyruklar söz konusu olduğunda, kalıcı olmayan iletiler kuyruk paylaşım grubunda yeniden başlatılır, ancak iletileri paylaşılan kuyruklara saklamak için kullanılan bağlaşım olanağının başarısızlıklarından kurtulmaz.

MQPER_PERSISTENCE_AS_Q_DEF

- Kuyruk bir küme kuyruğuyorsa, iletinin kalıcılığı, iletinin yerleştirildiği kuyruğun belirli eşgörünümüne sahip olan *hedef* kuyruk yöneticisinde tanımlanan **DefPersistence** özneliğinden alınır.

Küme kuyruğunun birden çok eşgörünümü varsa ve bu öznelikte farklılık gösterdiklerinde, bunlardan birinin tane seçkullanım seçtiği değer tahmin edilemez. Bu nedenle, bu özneliği tüm eşgörünümelerde aynı değere ayarlamalısınız. Böyle bir durumda değilse, kuyruk yöneticisi günlüklerine AMQ9407 hata iletisi gönderilir. Ayrıca bkz. [Diğer adlar, uzak kuyruklar ve küme kuyrukları için hedef nesne öznelikleri nasıl çözülür?](#)

DefPersistence değeri, ileti hedef kuyruğa yerleştirildiğinde *Persistence* alanına kopyalanır. *DefPersistence* daha sonra değiştirilirse, kuyruğa önceden yerleştirilen iletiler bundan etkilenmez.

- Kuyruk bir küme kuyruğu değilse, hedef kuyruk yöneticisi uzak olsa bile, iletinin kalıcılığı *yerel* kuyruk yöneticisinde tanımlanan **DefPersistence** özneliğinden alınır.

Kuyruk adı çözüm yolunda birden çok tanımlama varsa, varsayılan kalıcılık, yoldaki *birinci* tanımlamada bu özneliğin değerinden alınır. Bu şöyle olabilir:

- Diğer ad kuyruğu
- Yerel bir kuyruk
- Uzak kuyruğun yerel tanımlaması

- Kuyruk yöneticisi diğer adı
- Bir iletim kuyruğu (örneğin, *DefXmitQName* kuyruğu)

İleti konduğunda, *DefPersistence* değeri *Persistence* alanına kopyalanır. Daha sonra *DefPersistence* değiştirilirse, önceden konmuş olan iletiler etkilenmez.

Kalıcı ve kalıcı olmayan iletiler aynı kuyrukta bulunabilir.

Bir iletiyi yanıtlarken, uygulamaların yanıt iletilisine ilişkin istek iletilisinin kalıcılığını kullanması gerekir.

Bir MQGET çağrısı için, döndürülen değer: MQPER_PERSISTENT ya da MQPER_NOT_PERSISTENT.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri: MQPER_PERSISTENCE_AS_Q_DEF.

MQMD için MsgId (MQBYTE24)

Bu, bir iletiyi diğerinden ayırmak için kullanılan bir bayt dizilimidir. Genellikle, kuyruk yöneticisi tarafından izin verilmemesine rağmen, iki iletinin aynı ileti tanıtıcısına sahip olmaması gerekir. İleti tanıtıcısı, iletinin kalıcı bir özelliğidir ve kuyruk yöneticisinin yeniden başlatması boyunca devam eder. İleti tanıtıcısı bir karakter dizisi değil bayt dizisi olduğundan, ileti bir kuyruk yöneticisinden diğerine aktığında ileti tanıtıcısı karakter kümeleri arasında dönüştürülmez.

MQPUT ve MQPUT1 çağrıları için, uygulama tarafından MQMI_NONE ya da MQPMO_NEW_MSG_ID belirtilirse, kuyruk yöneticisi benzersiz bir ileti tanıtıcısı oluşturur.³ileti konduğunda ve iletiyle birlikte gönderilen ileti tanımlayıcısına yerleştirdiğinde. Kuyruk yöneticisi, gönderen uygulamaya ait ileti tanımlayıcısında bu ileti tanıtıcısını da döndürür. Uygulama, belirli iletilere ilişkin bilgileri kaydetmek ve uygulamanın diğer bölümlerinden gelen sorgulara yanıt vermek için bu değeri kullanabilir.

İleti bir konuya konuluyorsa, kuyruk yöneticisi yayınlanan her ileti için gerektiği gibi benzersiz ileti tanıtıcıları oluşturur. Uygulama tarafından MQPMO_NEW_MSG_ID belirtilirse, kuyruk yöneticisi çıkışta döndürülecek benzersiz bir ileti tanıtıcısı oluşturur. MQMI_NONE uygulama tarafından belirtilirse, çağrıdan dönüştürülen MQMD 'deki *MsgId* alanının değeri değişmez.

Alıkonan yayınlarla ilgili daha fazla ayrıntı için [“MQPMO için Seçenekler \(MQLONG\)” sayfa 501](#) içindeki MQPMO_RETAIN açıklamasına bakın.

İleti bir dağıtım listesine konuluyorsa, kuyruk yöneticisi gerektiğinde benzersiz ileti tanıtıcıları oluşturur; ancak, MQMI_NONE ya da MQPMO_NEW_MSG_ID belirtilmiş olsa da, MQMD 'deki *MsgId* alanının değeri çağrıdan dönüştürülen değeri değişmez. Uygulamanın kuyruk yöneticisi tarafından oluşturulan ileti tanıtıcılarını bilmesi gerekiyorsa, uygulamanın *MsgId* alanını içeren MQPMR kayıtlarını sağlaması gerekir.

Gönderen uygulama, ileti tanıtıcısı için MQMI_NONE dışında bir değer de belirtebilir; bu, kuyruk yöneticisinin benzersiz bir ileti tanıtıcısı oluşturmasını durdurur. Bir iletiyi ileten bir uygulama, özgün iletinin ileti tanıtıcısını yaymak için bunu kullanabilir.

Kuyruk yöneticisi aşağıdaki durumlar dışında bu alanı kullanmaz:

- Yukarıda açıklandığı gibi, istenirse benzersiz bir değer oluşturun
- İletiyeye ilişkin alma isteğini veren uygulamaya değeri sağlayın
- Değeri, bu ileti hakkında oluşturduğu herhangi bir rapor iletilisinin *CorrelId* alanına kopyalayın (*Report* seçeneklerine bağlı olarak)

³ Kuyruk yöneticisi tarafından oluşturulan bir *MsgId* , 4 baytlık ürün tanıtıcısından (ASCII ya da EBCDIC 'deAMQ- ya da CSQ- ; burada - , boş bir karakteri temsil eder) ve ardından benzersiz bir dizginin ürüne özgü somutlamasından oluşur. IBM MQ içinde bu, kuyruk yöneticisi adının ilk 12 karakterini ve sistem saatinden türetilmiş bir değeri içerir. Bu nedenle, ileti tanıtıcılarının benzersiz olmasını sağlamak için, iletişim kurabilen tüm kuyruk yöneticilerinin ilk 12 karakterde farklı adları olmalıdır. Benzersiz bir dizgi oluşturma yeteneği, sistem saatinin geriye doğru değiştirilmemesine de bağlıdır. Kuyruk yöneticisi tarafından oluşturulan bir ileti tanıtıcısının uygulama tarafından çoğaltılması olasılığını ortadan kaldırmak için uygulama, ASCII ya da EBCDIC 'de (X'41'-X'49 've X'C1'-X'C9') A-I aralığındaki ilk karakterlerle tanıtıcı oluşturmaktan kaçınmalıdır. Ancak, uygulamanın bu aralıklardaki ilk karakterlerle tanıtıcı oluşturmaması engellenmez.

Kuyruk yöneticisi ya da bir ileti kanalı aracısı bir rapor iletisi oluşturduğunda, *MsgId* alanını özgün iletinin *Report* alanı (MQRO_NEW_MSG_ID ya da MQRO_PASS_MSG_ID) tarafından belirlenen şekilde ayarlar. Rapor iletileri oluşturan uygulamalar da bunu yapmalıdır.

MQGET çağrısı için, *MsgId* kuyruktan belirli bir iletiyi almak için kullanılacak beş alandan biridir. Olağan durumda MQGET çağrısı kuyruktaki sonraki iletiyi döndürür, ancak herhangi bir birleşimde beş seçim ölçütünden biri ya da daha fazlası belirtilerek belirli bir ileti elde edilebilir; bu alanlar şunlardır:

- *MsgId*
- *CorrelId*
- *GroupId*
- *MsgSeqNumber*
- *Offset*

Uygulama bu alanlardan birini ya da daha fazlasını gerekli değerlere ayarlar ve ardından MQGMO 'nun *MatchOptions* alanındaki ilgili MQMO_* eşleşme seçeneklerini, bu alanları seçim ölçütü olarak kullanacak şekilde ayarlar. Yalnızca bu alanlarda belirtilen değerleri içeren iletiler alma adaylarıdır. *MatchOptions* alanının varsayılan değeri (uygulama tarafından değiştirilmediyse), ileti tanıtıcısı ve ilinti tanıtıcısıyla eşleşmektedir.

z/OS işletim sisteminde, kullanabileceğiniz seçim ölçütleri, kuyruk için kullanılan dizin tipiyle sınırlıdır. Daha fazla ayrıntı için **IndexType** kuyruk özneteliğine bakın.

Olağan durumda, döndürülen ileti, seçim ölçütlerini karşılayan, kuyruktaki *ilk* iletidir. Ancak MQGMO_BROWSE_NEXT belirtilirse, döndürülen ileti seçim ölçütlerini karşılayan *sonraki* iletidir; bu iletiye ilişkin tarama, yürürlükteki imleç konumundan *sonra* iletiliyle başlar.

Not: Kuyruk, seçim ölçütlerini karşılayan bir ileti için sırayla taranır, bu nedenle alma süreleri, seçim ölçütü belirtilmemesinden daha yavaştır, özellikle de uygun bir ileti bulunmadan önce birçok iletinin taranması gerektiyse. Bunun istisnaları şunlardır:

- **Multi** *CorrelId* dizininin gerçek bir sıralı tarama gerçekleştirme gereksinimini ortadan kaldırdığı 64 bitlik Multiplatforms üzerinde *CorrelId* tarafından bir MQGET çağrısı.
- **z/OS** z/OS üzerinde *IndexType* tarafından MQGET çağrısı.

Her iki durumda da, geri alma performansı iyileştirilmiştir.

Seçim ölçütlerinin çeşitli durumlarda nasıl kullanıldığına ilişkin ek bilgi için bkz. [Çizelge 495 sayfa 393](#) .

İleti tanıtıcısı olarak MQMI_NONE belirtilmesi, MQMO_MATCH_MSG_ID belirtilmemesiyle aynı etkiye sahiptir; yani, *herhangi bir* ileti tanıtıcısı eşleşir.

MQGET çağrısındaki **GetMsgOpts** değiştirilmesinde MQGMO_MSG_UNDER_CURSOR seçeneği belirtilirse bu alan yoksayılır.

Bir MQGET çağrısından döndüğünüzde, *MsgId* alanı döndürülen iletinin ileti tanıtıcısına (varsa) ayarlanır.

Aşağıdaki özel değer kullanılabilir:

MQMI_NONE

İleti tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQMI_NONE_ARRAY değişmezi de tanımlanır; bu değer MQMI_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu, MQGET, MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş/çıkış alanıdır. Bu alanın uzunluğu MQ_MSG_ID_LENGTH tarafından verilir. Bu alanın ilk değeri MQMI_NONE.

MQMD için CorrelId (MQBYTE24)

CorrelId alanı, ileti üstbilgisinde belirli bir iletiyi ya da ileti grubunu tanımlamak için kullanılacak bir özelliktir.

Bu, uygulamanın bir iletiyi başka bir iletiyle ilişkilendirmek ya da iletiyi uygulamanın gerçekleştirdiği diğer işle ilişkilendirmek için kullanılabileceği bir bayt dizgisidir. İlinti tanıtıcısı, iletinin kalıcı bir özelliğidir ve kuyruk yöneticisinin yeniden başlatması boyunca devam eder. İlinti tanıtıcısı bir karakter dizgisi değil bayt dizgisi olduğundan, ileti bir kuyruk yöneticisinden diğerine aktığında, ilinti tanıtıcısı karakter kümeleri arasında dönüştürülmez.

MQPUT ve MQPUT1 çağrıları için uygulama herhangi bir değeri belirtebilir. Kuyruk yöneticisi bu değeri iletiyle iletir ve ileti için alma isteğini gönderen uygulamaya teslim eder.

Uygulama MQPMO_NEW_CORREL_ID belirtirse, kuyruk yöneticisi iletiyle birlikte gönderilen benzersiz bir ilinti tanıtıcısı oluşturur ve MQPUT ya da MQPUT1 çağrısının çıkışında gönderen uygulamaya döndürülür.

Kuyruk yöneticisi tarafından oluşturulan bir ilinti tanıtıcısı, 3 baytlık bir ürün tanıtıcısından (ASCII ya da EBCDIC ' de AMQ ya da CSQ) ve ardından bir ayrılmış bayt ve benzersiz bir dizginin ürüne özgü bir somutlamasından oluşur. IBM MQ içinde bu ürüne özgü uygulama dizgisi, kuyruk yöneticisi adının ilk 12 karakterini ve sistem saatinden türetilmiş bir değeri içerir. Bu nedenle, ileti tanıtıcılarının benzersiz olmasını sağlamak için, iletişim kurabilen tüm kuyruk yöneticilerinin ilk 12 karakterden farklı adları olmalıdır. Benzersiz bir dizgi oluşturma yeteneği, sistem saatinin geriye doğru değiştirilmemesine de bağlıdır. Kuyruk yöneticisi tarafından oluşturulan bir ileti tanıtıcısının uygulama tarafından çoğaltılması olasılığını ortadan kaldırmak için uygulama, ASCII ya da EBCDIC 'de (X'41'-X'49 've X'C1'-X'C9') A-I aralığındaki ilk karakterlerle tanıtıcı oluşturmaktan kaçınmalıdır. Ancak, uygulamanın bu aralıklardaki ilk karakterlerle tanıtıcı oluşturmaması engellenmez.

Oluşturulan bu ilinti tanıtıcısı korunursa iletiyle birlikte alıkonur ve ileti, MQSUB çağrısına geçirilen MQSD ' de SubCorrelId alanında MQCI_NONE belirten abonelere yayın olarak gönderildiğinde ilinti tanıtıcısı olarak kullanılır. Alıkonan yayınlarla ilgili daha fazla ayrıntı için bkz. MQPMO seçenekleri .

Kuyruk yöneticisi ya da bir ileti kanalı aracısı bir rapor iletisi oluşturduğunda, özgün iletinin *Report* alanı (MQRO_COPY_MSG_ID_TO_CORREL_ID ya da MQRO_PASS_CORREL_ID) tarafından belirtilen şekilde *CorrelId* alanını ayarlar. Rapor iletileri oluşturan uygulamalar da bunu yapmalıdır.

MQGET çağrısı için *CorrelId* , kuyruktan alınacak belirli bir iletiyi seçmek için kullanılabilecek beş alandan biridir. Bu alana ilişkin değerlerin nasıl belirleneceğine ilişkin ayrıntılar için *MsgId* alanının açıklamasına bakın.

MQCI_NONE ' ın ilinti tanıtıcısı olarak belirtilmesi, MQMO_MATCH_CORREL_ID belirtilmemesiyle aynı etkiye sahiptir; yani, *herhangi bir* ilinti tanıtıcısı eşleşir.

MQGET çağrısındaki **GetMsgOpts** değiştirilmesinde MQGMO_MSG_UNDER_CURSOR seçeneği belirtilirse, bu alan yoksayıdır.

Bir MQGET çağrısından döndüğünüzde, *CorrelId* alanı döndürülen iletinin ilinti tanıtıcısına (varsa) ayarlanır.

Aşağıdaki özel değerler kullanılabilir:

MQCI_NONE

İlinti tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQCI_NONE_ARRAY değişmezi de tanımlanır; bu değer MQCI_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQCI_YENI_OTURUM

İleti, yeni bir oturumun başlangıcını belirtir.

Bu değer, CICS bridge tarafından yeni bir oturum başlangıcı, yani yeni bir ileti dizisinin başlangıcı olarak tanınır.

C programlama dili için, MQCI_NEW_SESSION_ARRAY değişmezi de tanımlanır; bu değer MQCI_NEW_SESSION ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQGET çağrısı için bu bir giriş/çıkış alanıdır. MQPUT ve MQPUT1 çağrıları için, MQPMO_NEW_CORREL_ID belirtilmediyse bu bir giriş alanıdır ve MQPMO_NEW_CORREL_ID belirtildiyse bir çıkış alanıdır. Bu alanın uzunluğu MQ_CORREL_ID_LENGTH tarafından verilir. Bu alanın ilk değeri MQCI_NONE.

Not:

Bir sıradüzendeki bir yayının ilinti tanıtıcısını iletemezsiniz. Alan, kuyruk yöneticisi tarafından kullanılır.

MQMD için BackoutCount (MQLONG)

Bu, iletinin bir iş biriminin parçası olarak MQGET çağrısı tarafından kaç kez döndürüldüğünü ve daha sonra kaç kez geri çekildiğini gösterir. Uygulamanın, ileti içeriğine dayalı işleme hatalarını saptamasına yardımcı olur. Sayı, MQGMO_BROWSE_* seçeneklerinden herhangi birini belirten MQGET çağrılarını dışlar.

Bu sayının doğruluğu **HardenGetBackout** kuyruk özniteliğinden etkilenir; bkz. "[Kuyruklara ilişkin öznitelikler](#)" sayfa 826.

z/OS' da 255 değeri, iletinin 255 ya da daha fazla kez geriletildiği anlamına gelir; döndürülen değer hiçbir zaman 255 'ten büyük değildir.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. MQPUT ve MQPUT1 çağrıları için yoksayıdır. Bu alanın ilk değeri şudur: 0.

MQMD için ReplyToQ (MQCHAR48)

Bu, ileti için alma isteğini yayınlayan uygulamanın MQMT_REPLY ve MQMT_REPORT iletilerini gönderdiği ileti kuyruğunun adıdır. Ad, *ReplyToQMgr* ile tanımlanan kuyruk yöneticisinde tanımlanan bir kuyruk yerel adıdır. Bu kuyruk bir model kuyruğu olmamalıdır; ancak, gönderen kuyruk yöneticisi ileti konduğunda bunu doğrulamaz.

MQPUT ve MQPUT1 çağrıları için, *MsgType* alanında MQMT_REQUEST değeri varsa ya da *Report* alanı tarafından herhangi bir rapor iletisi isteniyorsa bu alan boş olmamalıdır. Ancak, belirtilen (ya da yerine konan) değer, ileti tipi ne olursa olsun, ileti için alma isteğini veren uygulamaya iletilir.

ReplyToQMgr alanı boşsa, yerel kuyruk yöneticisi *ReplyToQ* adını kendi kuyruk tanımlamalarında arar. Bu ada sahip bir uzak kuyruk yerel tanımlaması varsa, iletilen iletideki *ReplyToQ* değeri, uzak kuyruk tanımındaki **RemoteQName** özniteliğinin değeriyle değiştirilir ve alan uygulama iletisi için bir MQGET çağrısı yayınladığında ileti tanımlayıcısında bu değer döndürülür. Uzak kuyruğun yerel tanımlaması yoksa, *ReplyToQ* değişmez.

Ad belirlenirse, sondaki boşlukları içerebilir; ilk boş karakter ve ardından gelen karakterler boşluk olarak işlenir. Ters durumda, adın kuyruklara ilişkin adlandırma kurallarına uygun olup olmadığı denetlenmez; iletilen iletide *ReplyToQ* değiştirilirse, bu, iletilen ad için de geçerlidir. Yapılan tek denetim, koşullar gerektiriyorsa, bir adın belirtilmiş olmasıdır.

Yanıt kuyruğu gerekmiyorsa, *ReplyToQ* alanını boş değere ya da (C programlama dilinde) boş değere ya da boş karakterin izlediği bir ya da daha çok boşluğa ayarlayın; alanı kullanıma hazırlanmamış olarak bırakmayın.

MQGET çağrısı için, kuyruk yöneticisi her zaman alanın uzunluğuna kadar boşluklarla doldurarak adı döndürür.

Bir rapor iletisi gerektiren bir ileti teslim edilemezse ve rapor iletisi de belirtilen kuyruğa teslim edilemezse, hem özgün ileti hem de rapor iletisi teslim edilmeyen mektup (teslim edilmeyen ileti) kuyruğuna gider (bkz. "[Kuyruk yöneticisine ilişkin öznitelikler](#)" sayfa 789 içinde açıklanan **DeadLetterQName** özniteliğine bakın).

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş alanıdır. Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQMD için ReplyToQMgr (MQCHAR48)

Bu, yanıt iletisinin ya da rapor iletisinin gönderileceği kuyruk yöneticisinin adıdır. *ReplyToQ*, bu kuyruk yöneticisinde tanımlanan bir kuyruğun yerel adıdır.

ReplyToQMgr alanı boşsa, yerel kuyruk yöneticisi kuyruk tanımlamalarında *ReplyToQ* adını arar. Bu ada sahip bir uzak kuyruk yerel tanımlaması varsa, iletilen iletideki *ReplyToQMgr* değeri, uzak kuyruk tanımındaki **RemoteQMgrName** özniteliğinin değeriyle değiştirilir ve alan uygulama iletisi için bir MQGET

çağrısı yayınladığında ileti tanımlayıcısında bu değer döndürülür. Uzak kuyruğun yerel tanımlaması yoksa, iletilen *ReplyToQMgr* yerel kuyruk yöneticisinin adıdır.

Ad belirlenirse, sondaki boşlukları içerebilir; ilk boş karakter ve ardından gelen karakterler boşluk olarak işlenir. Ters durumda, adın kuyruk yöneticilerine ilişkin adlandırma kurallarına uygun olup olmadığı ya da bu adın gönderen kuyruk yöneticisi tarafından bilinip bilinmediği denetlenmez; iletilen *ReplyToQMgr* başkasıyla değiştirilirse, iletilen ad için de geçerlidir.

Yanıt kuyruğu gerekmiyorsa, *ReplyToQMgr* alanını boş değere ya da (C programlama dilinde) boş değere ya da boş karakterin izlediği bir ya da daha çok boşluğa ayarlayın; alanı kullanıma hazırlanmamış olarak bırakmayın.

MQGET çağrısı için, kuyruk yöneticisi her zaman alanın uzunluğuna kadar boşluklarla doldurarak adı döndürür.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQMD için UserIdentifier (MQCHAR12)

Bu, iletinin **kimlik bağlamının** bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. "MQMD-İleti tanımlayıcı" sayfa 420 ve İleti bağlamı.

UserIdentifier, iletiyi oluşturan uygulamanın kullanıcı kimliğini belirtir. Kuyruk yöneticisi bu bilgileri karakter verileri olarak kabul eder, ancak biçimini tanımlamaz.

Bir ileti alındıktan sonra, sonraki bir MQOPEN ya da MQPUT1 çağrısının **ObjDesc** değiştirgesinin *AlternateUserId* alanında *UserIdentifier* komutunu kullanarak, açma işlemini gerçekleştiren uygulama yerine *UserIdentifier* kullanıcısı için yetki denetimi gerçekleştirebilirsiniz.

Kuyruk yöneticisi bir MQPUT ya da MQPUT1 çağrısı için bu bilgileri ürettiğinde:

- z/OS sistemlerinde, MQOO_ALTERNATE_USER_AUTHORITY ya da MQPMO_ALTERNATE_USER_AUTHORITY seçeneği belirtildiyse, kuyruk yöneticisi MQOPEN ya da MQPUT1 çağrısının **ObjDesc** değiştirgesinden *AlternateUserId* komutunu kullanır. İlgili seçenek belirtilmediyse, kuyruk yöneticisi ortamdan belirlenen bir kullanıcı kimliğini kullanır.
- Diğer ortamlarda, kuyruk yöneticisi her zaman ortamdan belirlenen bir kullanıcı kimliğini kullanır.

Kullanıcı kimliği ortamdan belirlendiğinde:

- z/OS' da, kuyruk yöneticisi aşağıdakileri kullanır:
 - MVS (toplu) için, JES JOB kartındaki ya da başlatılan görevdeki kullanıcı kimliği
 - TSO için, kullanıcı kimliği iş sunulurken işe yayıldı
 - CICS için, görevle ilişkili kullanıcı kimliği
 - IMS için, kullanıcı kimliği uygulamanın tipine bağlıdır:

- Bunun için:

- İleti dışı BMP bölgeleri
- İleti dışı IFP bölgeleri
- Başarılı bir GU çağrısı yayınlamamış olan BMP ve ileti IFP bölgeleri

Kuyruk yöneticisi, bölge JES JOB kartındaki kullanıcı kimliğini ya da TSO kullanıcı kimliğini kullanır. Bunlar boş değerli ya da boşsa, program belirtimi öbeğinin (PSB) adını kullanır.

- Bunun için:

- 'in başarılı bir GU çağrısı verdiği BMP ve ileti IFP bölgeleri
- MPP bölgeleri

Kuyruk yöneticisi aşağıdakilerden birini kullanır:

- İletiyile ilişkili oturum açmış kullanıcı kimliği

- Mantıksal uçbirim (LTERM) adı
- Bölge JES JOB kartındaki kullanıcı kimliği
- TSO kullanıcı kimliği
- PCB adı
- IBM i' de, kuyruk yöneticisi uygulama işiyle ilişkilendirilmiş kullanıcı tanımının adını kullanır.
- AIX and Linux' da, kuyruk yöneticisi aşağıdakileri kullanır:
 - Uygulamanın oturum açma adı
 - Oturum açma yoksa, işlemin etkin kullanıcı kimliği
 - Uygulama bir CICS işlemiyse, işlemle ilişkili kullanıcı kimliği
- Windows sistemlerinde, kuyruk yöneticisi oturum açmış kullanıcı adının ilk 12 karakterini kullanır.

Bu alan normalde kuyruk yöneticisi tarafından üretilen bir çıkış alanıdır; ancak bir MQPUT ya da MQPUT1 çağrısı için, bu alanı giriş/çıkış alanı yapabilir ve kuyruk yöneticisinin bu bilgileri oluşturmaya izin vermek yerine *UserIdentification* (kullanıcı kimliği) alanını belirtebilirsiniz. *PutMsgdeğiştir*inde MQPMO_SET_IDENTITY_CONTEXT ya da MQPMO_SET_ALL_CONTEXT belirtin ve kuyruk yöneticisinin bir MQPUT ya da MQPUT1 çağrısı için *UserIdentifier* alanını oluşturmamasını istemiyorsanız, *UserIdentifier* alanında bir kullanıcı kimliği belirtin.

MQPUT ve MQPUT1 çağrıları için, **PutMsgOpts** değiştirgesinde MQPMO_SET_IDENTITY_CONTEXT ya da MQPMO_SET_ALL_CONTEXT belirtildiyse, bu bir giriş/çıkış alanıdır. Alan içinde boş bir karakteri izleyen bilgiler atılır. Kuyruk yöneticisi boş karakteri ve aşağıdaki karakterleri boşluklara dönüştürür. MQPMO_SET_IDENTITY_CONTEXT ya da MQPMO_SET_ALL_CONTEXT belirtilmezse, girişte bu alan yoksayılır ve yalnızca çıkış alanıdır.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen *UserIdentifier* ögesini içerir. Bu, alıkonan yayınlarla ilgili daha fazla ayrıntı için iletiyle birlikte tutulan *UserIdentifier* değeri olur (MQPMO_RETAIN tanımına bakın), ancak ileti, abonelere gönderilen tüm yayınlarda *UserIdentifier* değerini geçersiz kılacak bir değer sağladığından, abonelere yayın olarak gönderildiğinde *UserIdentifier* olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen boştur.



Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu MQ_USER_ID_LENGTH tarafından verilir. Bu alanın ilk değeri C ' deki boş dizgi ve diğer programlama dillerindeki 12 boş karakterdir.

MQMD için AccountingToken (MQBYTE32)

Bu, iletinin *kimlik bağlamının* bir parçası olan muhasebe simgesidir. İleti bağlamıyla ilgili daha fazla bilgi için bkz. [“MQMD-İleti tanımlayıcı” sayfa 420](#) ve [İleti bağlamı](#).

AccountingToken , bir uygulamanın ileti sonucu olarak yapılan iş için uygun şekilde ücretlendirme yapmasını sağlar. Kuyruk yöneticisi bu bilgileri bir bit dizgisi olarak işler ve içeriğini denetlemez.

Kuyruk yöneticisi bu bilgileri aşağıdaki gibi oluşturur:

- Alanın ilk baytı, izleyen baytlarda var olan muhasebe bilgilerinin uzunluğuna ayarlanır; bu uzunluk, sıfır ile 30 aralığındadır ve ilk baytta ikili tamsayı olarak saklanır.
- İkinci ve sonraki baytlar (uzunluk alanı tarafından belirlendiği şekilde), ortama uygun muhasebe bilgilerine ayarlanır.
 -  z/OS ' de muhasebe bilgileri aşağıdaki şekilde ayarlanır:
 - z/OS toplu iş için, JES JOB kartındaki ya da EXEC kartındaki bir JES ACCT deyimindeki muhasebe bilgileri (virgül ayırıcıları X'FF ' olarak değiştirilir). Gerekirse, bu bilgiler 31 bayta kesilir.
 - TSO için, kullanıcının hesap numarası.
 - CICS için, LU 6.2 iş birimi tanıtıcısı (UEPUOWDS) (26 bayt).
 - IMS için, 8 karakterlik PSB adı 16 karakterlik IMS kurtarma simgesiyle bitleştirilir.
 -  IBM i işletim tarihinde, muhasebe bilgileri için muhasebe koduna ayarlanır.

- **Linux** **AIX** AIX and Linux' da, muhasebe bilgileri ASCII karakterlerinde sayısal kullanıcı kimliğine ayarlanır.
- **Windows** Windows üzerinde, muhasebe bilgileri sıkıştırılmış biçimde bir Windows güvenlik tanıtıcısına (SID) ayarlanır. SID, *UserIdentifier* alanında saklanan kullanıcı kimliğini benzersiz olarak tanımlar. SID *AccountingToken* alanında saklandığında, 6 baytlık Tanıtıcı Yetkisi (SID ' nin üçüncü ve sonraki baytlarında bulunur) atlanır. Örneğin, Windows SID 28 bayt uzunluğdaysa, SID bilgilerinin 22 baytı *AccountingToken* alanında saklanır.
- Muhasebe alanının son baytı (bayt 32), muhasebe simgesi tipine ayarlanır (bu durumda M`QACTT_NT_SECURITY_ID`, x 'Ob'):

M`QACTT_CICS_LUOW_ID`

CICS LUOW tanıtıcısı.

Windows **M`QACTT_NT_SECURITY_ID`**

Windows güvenlik tanıtıcısı.

IBM i **M`QACTT_OS400_ACCOUNT_TOKEN`**

IBM i hesap simgesi.

UNIX **M`QACTT_UNIX_NUMERIC_ID`**

UNIX sayısal tanıtıcı.

M`QACTT_USER`

Kullanıcı tanımlı hesap simgesi.

M`QACTT_UNKNOWN`

Bilinmeyen muhasebe simgesi tipi.

Muhasebe simgesi tipi, yalnızca aşağıdaki ortamlarda belirtik bir değere ayarlanır:

- **AIX** AIX
- **IBM i** IBM i
- **Linux** Linux
- **Windows** Windows

ve bu sistemlere bağlı IBM MQ MQI clients için. Diğer ortamlarda, muhasebe simgesi tipi M`QACTT_UNKNOWN` değerine ayarlanır. Bu ortamlarda, alınan muhasebe simgesi tipini çıkarmak için *PutAppLType* alanını kullanın.

- Diğer tüm baytlar ikili sıfır olarak ayarlanır.

M`QPUT` ve M`QPUT1` çağrıları için, **PutMsgOpts** değiştirgesinde M`QPMO_SET_IDENTITY_CONTEXT` ya da M`QPMO_SET_ALL_CONTEXT` belirtildiyse, bu bir giriş/çıkış alanıdır. M`QPMO_SET_IDENTITY_CONTEXT` ya da M`QPMO_SET_ALL_CONTEXT` belirtilmezse, girişte bu alan yoksayılr ve yalnızca çıkış alanıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

Bir M`QPUT` ya da M`QPUT1` çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen *AccountingToken* ögesini içerir. Bu değer, alıkonursa iletiyle birlikte tutulan *AccountingToken* değeridir (alıkonan yayınlarla ilgili daha fazla ayrıntı için "[MQPMO için Seçenekler \(M`QLONG`\)](#)" sayfa 501 içindeki M`QPMO_RETAIN` tanımına bakın), ancak ileti, abonelere gönderilen tüm yayınlarda *AccountingToken* değerini geçersiz kılacak bir değer sağladıkları için yayın olarak gönderildiğinde *AccountingToken* olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen ikili sıfırdır.

Bu, M`QGET` çağrısına ilişkin bir çıkış alanıdır.

Bu alan, kuyruk yöneticisinin karakter kümesine dayalı olarak herhangi bir çeviriye tabi değildir; alan, karakter dizgisi olarak değil, bit dizgisi olarak işlenir.

Kuyruk yöneticisi bu alandaki bilgilerle hiçbir şey yapmaz. Bilgileri muhasebe amacıyla kullanmak istiyorsa, uygulama bilgiyi yorumlamalıdır.

AccountingToken alanı için aşağıdaki özel değeri kullanabilirsiniz:

MQACT_NONE

Hesap simgesi belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQACT_NONE_ARRAY değişmezi de tanımlanır; bu değer MQACT_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu alanın uzunluğu MQ_ACCOUNTING_TOKEN_LENGTH tarafından verilir. Bu alanın ilk değeri MQACT_NONE.

MQMD için ApplIdentityVerileri (MQCHAR32)

Bu, iletinin **kimlik bağlamının** bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. "MQMD-İleti tanımlayıcı" sayfa 420 ve İleti bağlamı.

ApplIdentityData, uygulama grubu tarafından tanımlanan bilgilerdir ve ileti ya da oluşturucusu hakkında ek bilgi sağlamak için kullanılabilir. Kuyruk yöneticisi bu bilgileri karakter verileri olarak kabul eder, ancak biçimini tanımlamaz. Kuyruk yöneticisi bu bilgileri oluşturduğunda tamamen boş olur.

MQPUT ve MQPUT1 çağrıları için, **PutMsgOpts** değiştirgesinde MQPMO_SET_IDENTITY_CONTEXT ya da MQPMO_SET_ALL_CONTEXT belirtildiyse, bu bir giriş/çıkış alanıdır. Boş bir karakter varsa, boş değer ve aşağıdaki karakterler kuyruk yöneticisi tarafından boşluklara dönüştürülür. MQPMO_SET_IDENTITY_CONTEXT ya da MQPMO_SET_ALL_CONTEXT belirtilmezse, girişte bu alan yoksayılr ve yalnızca çıkış alanıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. İleti bağlamı.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen *ApplIdentityData* ögesini içerir. Bu, alıkonan yayınlarla ilgili daha fazla ayrıntı için iletiyle birlikte tutulan *ApplIdentityData* değeri olur (MQPMO_RETAIN tanımına bakın), ancak ileti, abonelere gönderilen tüm yayınlarda *ApplIdentityData* değerini geçersiz kılacak bir değer sağladığından, abonelere yayın olarak gönderildiğinde *ApplIdentityData* olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen boştur.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu MQ_APPL_IDENTITY_DATA_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerinde 32 boş karakterdir.

MQMD için PutApplTipi (MQLONG)

Bu, iletiyi koyan uygulamanın tipidir ve iletinin **kaynak bağlamının** bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. "MQMD-İleti tanımlayıcı" sayfa 420 ve İleti bağlamı.

PutApplType aşağıdaki standart tiplerden birine sahip olabilir. Kendi tiplerinizi de tanımlayabilirsiniz, ancak yalnızca MQAT_USER_LAST ile MQAT_USER_FIRST aralığındaki değerlerle tanımlayabilirsiniz.

MQAT_AIX

AIX uygulaması (MQAT_UNIX ile aynı değer).

MQAT_AMQP

AMQP iletişim kuralı uygulaması

MQAT_BROKER

Borsacı.

MQAT_CICS

CICS hareketi.

MQAT_CICS_BRIDGE

CICS bridge.

MQAT_CICS_VSE

CICS/VSE hareketi.

MQAT_DOS

PC DOS üzerinde IBM MQ MQI client uygulaması.

MQAT_DQM

Dağıtılmış kuyruk yöneticisi aracı.

MQAT_GUARDIAN

Tandem Guardian uygulaması (MQAT_NSK ile aynı değer).

MQAT_IMS

IMS uygulaması.

MQAT_IMS_BRIDGE

IMS köprüsü.

MQAT_JAVA

Java.

MQAT_MVS

MVS ya da TSO uygulaması (MQAT_ZOS ile aynı değer).

MQAT_NOTES_AGENT

Lotus Notes Agent uygulaması.

MQAT_OS390

OS/390 uygulaması (MQAT_ZOS ile aynı değer).

MQAT_OS400

IBM i uygulaması.

MQAT_QMGR

Kuyruk yöneticisi.

MQAT_UNIX

UNIX uygulaması.

MQAT_VOS

Stratus VOS uygulaması.

MQAT_WINDOWS

16 bit Windows uygulaması.

MQAT_WINDOWS_NT

32 bit Windows uygulaması.

MQAT_WLM

z/OS iş yükü yöneticisi uygulaması.

MQAT_XCF

-XCF.

MQAT_ZOS

z/OS uygulaması.

MQAT_DEFAULT

Varsayılan uygulama tipi.

Bu, uygulamanın çalıştığı altyapıya ilişkin varsayılan uygulama tipidir.

Not: Bu değişmez değeri ortama özgüdür. Bu nedenle, her zaman üstbilgiyi kullanarak uygulamayı derleyin, uygulamanın çalışacağı platforma uygun dosyaları ekleyin ya da kopyalayın ekleyin.

MQAT_BILINMIYOR

Diğer bağlam bilgileri var olsa da uygulama tipinin bilinmediğini belirtmek için bu değeri kullanın.

MQAT_USER_FIRST

Kullanıcı tanımlı uygulama tipi için en düşük değer.

MQAT_USER_LAST

Kullanıcı tanımlı uygulama tipi için en yüksek değer.

Aşağıdaki özel değer de oluşabilir:

MQAT_NO_CONTEXT

Bu değer, bağlam olmadan bir ileti konduğunda (yani, MQPMO_NO_CONTEXT bağlam seçeneği belirtildiğinde) kuyruk yöneticisi tarafından ayarlanır.

Bir ileti alındığında, iletinin bağlamı olup olmadığına karar vermek için *PutApplType* sınanabilir (diğer bağlam alanlarından herhangi biri boş değilse, MQPMO_SET_ALL_CONTEXT kullanan bir uygulama tarafından *PutApplType* ' un hiçbir zaman MQAT_NO_CONTEXT olarak ayarlanmaması önerilir).

Kuyruk yöneticisi bu bilgileri bir uygulama koymasının sonucu olarak oluşturduğunda, alan ortam tarafından belirlenen bir değere ayarlanır. IBM üzerinde MQAT_OS400; kuyruk yöneticisi IBM üzerinde MQAT_CICS kullanmaz.

MQPUT ve MQPUT1 çağrıları için, **PutMsgOpts** değiştirilmesinde MQPMO_SET_ALL_CONTEXT belirtilirse bu bir giriş/çıkış alanıdır. MQPMO_SET_ALL_CONTEXT belirtilmezse, girişte bu alan yoksayılar ve yalnızca çıkış alanıdır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın ilk değeri MQAT_NO_CONTEXT 'dir.

MQMD için PutApplAdı (MQCHAR28)

Bu, iletiyi koyan uygulamanın adıdır ve iletinin *kaynak bağlamının* bir parçasıdır. İçerik platformlar arasında farklılık gösterir ve yayınlar arasında farklılık gösterebilir.

İleti bağlamıyla ilgili daha fazla bilgi için bkz. “MQMD-İleti tanımlayıcı” sayfa 420 ve İleti bağlamı.

IBM MQ 9.1.2 ' den ek programlama dillerinde uygulama adını belirtebilirsiniz. Ek bilgi için Desteklenen programlama dillerinde uygulama adının belirtilmesi başlıklı konuya bakın.

PutApplName biçimi *PutApplType* değerine bağlıdır ve bir yayın düzeyinden diğerine değişebilir. Değişiklikler nadirdir, ancak ortam değişirse gerçekleşir.

Kuyruk yöneticisi bu alanı (MQPMO_SET_ALL_CONTEXT dışındaki tüm seçenekler için) ayarladığında, alanı ortamın belirlediği bir değere ayarlar:

- **z/OS** z/OS' da, kuyruk yöneticisi aşağıdakileri kullanır:
 - z/OS toplu iş için, JES JOB kartındaki 8 karakterlik iş adı
 - TSO için 7 karakterlik TSO kullanıcı kimliği
 - CICS için, 8 karakterli applid, ardından 4 karakterli tranid
 - IMS için, 8 karakterlik IMS sistem tanıtıcısı ve ardından 8 karakterlik PSB adı
 - XCF için, 8 karakterlik XCF grubu adı ve ardından 16 karakterlik XCF üye adı
 - Kuyruk yöneticisi tarafından oluşturulan bir ileti için, kuyruk yöneticisi adının ilk 28 karakteri
 - CICS olmadan dağıtılmış kuyruğa alma için, kanal başlatıcısının 8 karakterlik iş adı ve ardından, teslim edilmeyen iletiler kuyruğuna yerleştirilen modülün 8 karakterlik adı ve ardından 8 karakterlik bir görev tanıtıcısı gelir.

Ad ya da adların her biri, alanın geri kalanında olduğu gibi, sağda boşluklarla doldurulur. Birden çok ad varsa, aralarında ayırıcı yoktur.

- **Windows** Windows sistemlerinde kuyruk yöneticisi aşağıdaki adları kullanır:
 - CICS uygulaması için CICS işlem adı
 - CICS dışı bir uygulama için, yürütülür dosyanın tam olarak nitelenmiş adının en sağdaki 28 karakter

- **IBM i** IBM i' da, kuyruk yöneticisi tam olarak nitelenmiş iş adını kullanır.

- **Linux** **AIX** AIX and Linux üzerinde, kuyruk yöneticisi aşağıdaki adları kullanır:

- CICS uygulaması için CICS işlem adı
- CICS dışı bir uygulama için, MQ işletim sisteminden işlemin adını ister. Bu, tam yol olmadan program dosyası adı olarak döndürülür. Daha sonra MQ , bu süreç adını MQMD.PutApplName alanı aşağıdaki gibidir:

AIX **AIX**

Ad 28 byte 'tan küçük ya da ona eşitse, ad boşluklarla sağa doldurulur.

Ad 28 bayttan büyükse, adın en soldaki 28 baytı eklenir.

Linux Linux

Ad 15 byte 'tan küçük ya da ona eşitse, ad boşluklarla sağa doldurulur.

Ad 15 bayttan büyükse, adın en solundaki 15 bayt araya eklenir ve sağa boşluklarla doldurulur.

Örneğin, /opt/mqm/samp/bin/amqspu QNAME QMNAME çalıştırırsanız, PutApplAdı ' amqspu ' olur. Bu MQCHAR28 alanında 21 boşluk karakteri vardır. /opt/mqm/samp/bin içeren tam yolun PutApplAdına eklenmediğini unutmayın.

MQPUT ve MQPUT1 çağrılarını için, **PutMsgOpts** değıştirgesinde MQPMO_SET_ALL_CONTEXT belirtilirse bu bir giriş/çıkış alanıdır. Alan içinde boş bir karakteri izleyen bilgiler atılır. Boş karakter ve aşağıdaki karakterler, kuyruk yöneticisi tarafından boşluklara dönüştürülür. MQPMO_SET_ALL_CONTEXT belirtilmezse, girişte bu alan yoksayılr ve yalnızca çıkış alanıdır.

MQMD için PutDate (MQCHAR8)

Bu, iletinin konma tarihidir ve iletinin **kaynak bağlamının** bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. “MQMD-İleti tanımlayıcı” sayfa 420 ve [İleti bağlamı](#).

Bu alanın kuyruk yöneticisi tarafından oluşturulduğu tarih için kullanılan biçim:

- YYYYAAAGG

karakterlerin temsil ettiği yer:

YYYY

yıl (dört sayısal basamak)

MM

yılın ayı (01-12)

GG

ayın günü (01-31)

Greenwich Ortalama Saati (GMT), sistem saatinin GMT ' ye doğru şekilde ayarlanmasına bağlı olarak *PutDate* ve *PutTime* alanları için kullanılır.

İleti bir iş biriminin parçası olarak konduysa, tarih, iş biriminin kesinleştirildiği tarih değil, iletinin konduğu tarihtir.

MQPUT ve MQPUT1 çağrılarını için, **PutMsgOpts** değıştirgesinde MQPMO_SET_ALL_CONTEXT belirtilirse bu bir giriş/çıkış alanıdır. Alanın içeriği kuyruk yöneticisi tarafından denetlenmez; ancak, alan içindeki boş karakterin ardından gelen bilgiler atılır. Kuyruk yöneticisi boş karakteri ve aşağıdaki karakterleri boşluklara dönüştürür. MQPMO_SET_ALL_CONTEXT belirtilmezse, girişte bu alan yoksayılr ve yalnızca çıkış alanıdır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu MQ_PUT_DATE_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 8 boş karakterdir.

MQMD için PutTime (MQCHAR8)

İletinin konma zamanıdır ve iletinin **kaynak bağlamının** bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. “MQMD-İleti tanımlayıcı” sayfa 420 ve [İleti bağlamı](#).

Bu alanın kuyruk yöneticisi tarafından oluşturulduğu saat için kullanılan biçim:

- HHMMSSSTH

burada karakterler (sırayla) temsil edilir:

SS

saat (00-23)

MM

dakika (00-59)

SS

saniye (00-59; nota bakın)

T

saniyenin onda biri (0-9)

H

saniyenin yüzde biri (0-9)

Not: Sistem saati çok doğru bir zaman standardıyla eşitlenirse, nadir durumlarda *PutTime* içindeki saniyeler için 60 ya da 61 geri gönderilebilir. Bu, artık saniyeler genel zaman standardına eklendiğinde olur.

Greenwich Ortalama Saati (GMT), sistem saatinin GMT ' ye doğru şekilde ayarlanmasına bağlı olarak *PutDate* ve *PutTime* alanları için kullanılır.

İleti bir iş biriminin parçası olarak konduysa, zaman, iş biriminin kesinleştirildiği zaman değil, iletinin bulunduğu zamandır.

MQPUT ve MQPUT1 çağrılarını için, **PutMsgOpts** değiştirilmesinde MQPMO_SET_ALL_CONTEXT belirtilirse bu bir giriş/çıkış alanıdır. Kuyruk yöneticisi, alan içindeki boş karakterin ardından gelen bilgilerin atılması dışında, alanın içeriğini denetlemez. Kuyruk yöneticisi boş karakteri ve aşağıdaki karakterleri boşluklara dönüştürür. MQPMO_SET_ALL_CONTEXT belirtilmezse, girişte bu alan yoksayılr ve yalnızca çıkış alanıdır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu MQ_PUT_TIME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 8 boş karakterdir.

MQMD için ApplOriginVerileri (MQCHAR4)

Bu, iletinin *kaynak bağlamının* bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. "[MQMD-İleti tanımlayıcı](#)" sayfa 420 ve [İleti bağlamı](#).

ApplOriginData , uygulama grubu tarafından tanımlanan ve iletinin kökenine ilişkin ek bilgi sağlamak için kullanılabilir. Örneğin, kimlik verilerinin güvenilir olup olmadığını belirtmek için uygun kullanıcı yetkisiyle çalışan uygulamalar tarafından ayarlanabilir.

Kuyruk yöneticisi bu bilgileri karakter verileri olarak kabul eder, ancak biçimini tanımlamaz. Kuyruk yöneticisi bu bilgileri oluşturduğunda tamamen boş olur.

MQPUT ve MQPUT1 çağrılarını için, **PutMsgOpts** değiştirilmesinde MQPMO_SET_ALL_CONTEXT belirtilirse bu bir giriş/çıkış alanıdır. Alan içinde boş bir karakteri izleyen bilgiler atılır. Kuyruk yöneticisi boş karakteri ve aşağıdaki karakterleri boşluklara dönüştürür. MQPMO_SET_ALL_CONTEXT belirtilmezse, girişte bu alan yoksayılr ve yalnızca çıkış alanıdır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu MQ_APPL_ORIGIN_DATA_LENGTH tarafından verilir. Bu alanın ilk değeri, C ' deki boş dizgi ve diğer programlama dillerindeki 4 boş karakterdir.

ApplOriginData ayarlansa da ileti yayınlandığında, aldığı abonelikte boş olur.

MQMD için GroupId (MQBYTE24)

Bu, fiziksel iletinin ait olduğu belirli ileti grubunu ya da mantıksal iletiyi tanımlamak için kullanılan bir bayt dizilimidir. *GroupId* , ileti için bölümlenmeye izin verildiğinde de kullanılır. Tüm bu durumlarda, *GroupId* boş olmayan bir değere sahiptir ve *MsgFlags* alanında aşağıdaki işaretlerden biri ya da daha fazlası ayarlanır:

- MQMF_MSG_IN_GROUP
- MQMF_LAST_MSG_IN_GROUP
- MQMF_SEGMENT
- MQMF_LAST_SEGMENT
- MQMF_SEGMENTATION_ALLOWED

Bu işaretlerin hiçbiri ayarlanmazsa, *GroupId* özel boş değerli MQGI_NONE değerine sahiptir.

Aşağıdaki durumlarda, uygulamanın MQPUT ya da MQGET çağrısında bu alanı ayarlamasına gerek yoktur:

- MQPUT çağrısında MQPMO_LOGICAL_ORDER belirtildi.

- MQGET çağrısında MQMO_MATCH_GROUP_ID belirtilmedi.

Bunlar, rapor iletisi olmayan iletiler için bu çağrılarını kullanmanın önerilen yollarıdır. Ancak, uygulama daha fazla denetim gerektiriyorsa ya da çağrı MQPUT1ise, uygulamanın *GroupId* ' in uygun bir değere ayarlandığından emin olması gerekir.

İleti grupları ve bölümler yalnızca grup tanıtıcısı benzersizse doğru olarak işlenebilir. Bu nedenle, *uygulamaları kendi grup tanıtıcılarını oluşturmamalıdır* ; Bunun yerine, uygulamalar aşağıdakilerden birini yapmalıdır:

- MQPMO_LOGICAL_ORDER belirtilirse, kuyruk yöneticisi mantıksal iletinin grup ya da kesiminde ilk ileti için otomatik olarak benzersiz bir grup tanıtıcısı oluşturur ve mantıksal iletinin grup ya da bölümlerinde kalan iletiler için bu grup tanıtıcısını kullanır; bu nedenle, uygulamanın özel bir işlem gerçekleştirilmesi gerekmez. Bu önerilen yordamdır.
- MQPMO_LOGICAL_ORDER belirtilmezse, uygulamanın kuyruk yöneticisinden grup tanıtıcısını oluşturmasını istemesi gerekir; bunun için, ilk MQPUT ya da MQPUT1 çağrısında mantıksal iletinin gruptaki ya da bölümündeki bir ileti için *GroupId* ögesini MQGI_NONE olarak ayarlayın. Bu çağrıdan çıkışta kuyruk yöneticisi tarafından döndürülen grup tanıtıcısı, mantıksal iletinin gruptaki ya da bölümlerindeki geri kalan iletiler için kullanılmalıdır. Bir ileti grubu bölümlenmiş iletiler içeriyorsa, gruptaki tüm kesimler ve iletiler için aynı grup tanıtıcısı kullanılmalıdır.

MQPMO_LOGICAL_ORDER belirtilmediğinde, gruplardaki ve mantıksal ileti bölümlerindeki iletiler herhangi bir sıraya konabilir (örneğin, ters sırada), ancak grup tanıtıcısı bu iletilerin herhangi biri için yayınlanan *ilk* MQPUT ya da MQPUT1 çağrısıyla ayrılmalıdır.

MQPUT ve MQPUT1 çağrılarında girişte, kuyruk yöneticisi Kuyruktaki fiziksel sıralamak kısmında açıklanan değeri kullanır. MQPUT ve MQPUT1 çağrılarının çıkışında, kuyruk yöneticisi bu alanı, açılan nesne bir dağıtım listesi değil de tek bir kuyruksa, iletiyle birlikte gönderilen değere ayarlar, ancak açılan nesne bir dağıtım listesiye değiştirilmeden bırakır. İkinci durumda, uygulamanın oluşturulan grup tanıtıcılarını bilmesi gerekiyorsa, uygulamanın *GroupId* alanını içeren MQPMR kayıtlarını sağlaması gerekir.

MQGET çağrısına girişte, kuyruk yöneticisi Çizelge 495 sayfa 393 içinde açıklanan değeri kullanır. MQGET çağrısının çıkışında, kuyruk yöneticisi bu alanı alınan iletinin değerine ayarlar.

Aşağıdaki özel değer tanımlandı:

MQGI_NONE

Grup tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır. Bu, mantıksal ileti kesimleri değil, gruplarda olmayan ve bölümlenmeye izin verilmeyen iletiler için kullanılan değerdir.

C programlama dili için, MQGI_NONE_ARRAY değişmezi de tanımlanır; bu değer MQGI_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu alanın uzunluğu MQ_GROUP_ID_LENGTH tarafından verilir. Bu alanın ilk değeri MQGI_NONE. *Version* , MQMD_VERSION_2' den küçükse bu alan yoksayılır.

MsgSeqMQMD için sayı (MQLONG)

Bu, bir grup içindeki mantıksal iletinin sıra numarasıdır.

Sıra numaraları 1 'den başlar ve gruptaki her yeni mantıksal ileti için en çok 999 999 999 'a kadar 1 artırılır. Bir grupta olmayan fiziksel bir iletinin sıra numarası 1 'dir.

Aşağıdaki durumlarda, uygulamanın MQPUT ya da MQGET çağrısında bu alanı ayarlamasına gerek yoktur:

- MQPUT çağrısında MQPMO_LOGICAL_ORDER belirtildi.
- MQGET çağrısında MQMO_MATCH_MSG_SEQ_NUMBER belirtilmedi.

Bunlar, rapor iletisi olmayan iletiler için bu çağrılarını kullanmanın önerilen yollarıdır. Ancak, uygulama daha fazla denetim gerektiriyorsa ya da çağrı MQPUT1ise, uygulamanın *MsgSeqNumber* ' in uygun bir değere ayarlandığından emin olması gerekir.

MQPUT ve MQPUT1 çağrılarında girişte, kuyruk yöneticisi Kuyruktaki fiziksel sıralamaksında açıklanan değeri kullanır. MQPUT ve MQPUT1 çağrılarının çıkışında, kuyruk yöneticisi bu alanı iletiyle birlikte gönderilen değere ayarlar.

MQGET çağrısına girişte, kuyruk yöneticisi Çizelge 495 sayfa 393 içinde gösterilen değeri kullanır. MQGET çağrısının çıkışında, kuyruk yöneticisi bu alanı alınan iletinin değerine ayarlar.

Bu alanın ilk değeri birdir. *Version* , MQMD_VERSION_2' den küçükse bu alan yoksayılr.

MQMD için görelı konum (MQLONG)

Bu, fiziksel iletideki verilerin, veri formunun bir parçası olduđu mantıksal iletinin başlangıcından itibaren bayt cinsinden görelı konumudur. Bu verilere *segment* denir. Görelı konum, 0-999 999 999 aralığındadır. Mantıksal iletinin bir parçası olmayan fiziksel bir iletinin görelı konumu sıfır.

Aşağıdaki durumlarda, uygulamanın MQPUT ya da MQGET çağrısında bu alanı ayarlamasına gerek yoktur:

- MQPUT çağrısında MQPMO_LOGICAL_ORDER belirtildi.
- MQGET çağrısında MQMO_MATCH_OFFSET belirtilmedi.

Bunlar, rapor iletisi olmayan iletiler için bu çağrıları kullanmanın önerilen yollarıdır. Ancak, uygulama bu koşullara uymuyorsa ya da çağrı MQPUT1 ise, uygulamanın *Offset* ' in uygun bir değere ayarlandığından emin olması gerekir.

MQPUT ve MQPUT1 çağrılarında girişte, kuyruk yöneticisi Kuyruktaki fiziksel sıralamaksında açıklanan değeri kullanır. MQPUT ve MQPUT1 çağrılarının çıkışında, kuyruk yöneticisi bu alanı iletiyle birlikte gönderilen değere ayarlar.

Mantıksal bir iletinin bir kesimine ilişkin rapor iletisi raporlaması için, kuyruk yöneticisi tarafından tutulan kesim bilgilerindeki görelı konumu güncellemek için *OriginalLength* alanı (MQOL_UNDEFINED değilse) kullanılır.

MQGET çağrısına girişte, kuyruk yöneticisi Çizelge 495 sayfa 393 içinde gösterilen değeri kullanır. MQGET çağrısının çıkışında, kuyruk yöneticisi bu alanı alınan iletinin değerine ayarlar.

Bu alanın ilk değeri sıfırdır. *Version* , MQMD_VERSION_2' den küçükse bu alan yoksayılr.

MQMD için MsgFlags (MQLONG)

MsgFlags , iletinin özniteliklerini belirten ya da işlenmesini denetleyen işaretlerdir.

MsgFlags aşağıdaki kategorilere ayrılır:

- Bölümleme işaretleri
- Durum işaretleri

Bölümleme işaretleri: Bir ileti kuyruk için çok büyükse, iletiyi kuyruğa koyma girişimi genellikle başarısız olur. Bölümleme, kuyruk yöneticisinin ya da uygulamanın iletiyi segmentler adı verilen daha küçük parçalara ayırdığı ve her kesimi ayrı bir fiziksel ileti olarak kuyruğa yerleştirdiği bir tekniktir. İletiyi alan uygulama, bölümleri birer birer alabilir ya da kuyruk yöneticisinden bölümleri MQGET çağrısıyla döndürülen tek bir iletide yeniden birleştirmesini isteyebilir. İkincisi, MQGET çağrısında MQGMO_COMPLETE_MSG seçeneği belirtilerek ve tüm iletiyi barındıracak kadar büyük bir arabellek sağlanarak gerçekleştirilir. (MQGMO_COMPLETE_MSG seçeneğinin ayrıntıları için bkz. “MQGMO-Get-message seçenekleri” sayfa 368 .) Bir ileti, gönderen kuyruk yöneticisinde, bir ara kuyruk yöneticisinde ya da hedef kuyruk yöneticisinde kesimlere ayrılabilir.

Bir iletinin bölümlenmesini denetlemek için aşağıdakilerden birini belirtebilirsiniz:

MQMF_SEGMENTATION_INHIBE

Bu seçenek, iletinin kuyruk yöneticisi tarafından bölümlere ayrılmasını önler. Zaten bir bölüm olan bir ileti için belirtilirse, bu seçenek kesimin daha küçük bölümlere bölünmesini önler.

Bu işaretin değeri ikili sıfır. Bu varsayılandır.

MQMF_SEGMENTATION_ALLOWED

Bu seçenek, iletinin kuyruk yöneticisi tarafından bölümlere ayrılmasını sağlar. Zaten bir bölüm olan bir ileti için belirtilirse, bu seçenek kesimin daha küçük bölümlere bölünmesine izin verir. MQMF_SEGMENTATION_ALLOWED, MQMF_SEGMENT ya da MQMF_LAST_SEGMENT ayarlanmadan ayarlanabilir.

- z/OS işletim sistemi üzerinde, kuyruk yöneticisi iletilerin bölümlere ayrılmasını desteklemez. Bir ileti kuyruk için çok büyükse, MQPUT ya da MQPUT1 çağrısı MQRC_MSG_TOO_BIG_FOR_Q neden koduyla başarısız olur. Ancak, MQMF_SEGMENTATION_ALLOWED seçeneği yine de belirtilebilir ve iletinin uzak bir kuyruk yöneticisinde bölümlenmesine izin verir.

Kuyruk yöneticisi bir iletiyi kestiğinde, kuyruk yöneticisi her bölümle birlikte gönderilen MQMD 'nin kopyasında MQMF_SEGMENT işaretini açar, ancak MQPUT ya da MQPUT1 çağrısında uygulama tarafından sağlanan MQMD' de bu işaretlerin ayarlarını değiştirmez. Mantıksal iletideki son bölüm için kuyruk yöneticisi, bölümle birlikte gönderilen MQMD ' de MQMF_LAST_SEGMENT işaretini de açar.

Not: İletileri MQPMO_LOGICAL_ORDER olmadan MQMF_SEGMENTATION_ALLOWED ile koyarken dikkatli olun. İleti:

- Bir bölüm değil, ve
- Bir grupta değil, ve
- İftira edilmiyor.

Kuyruk yöneticisinin her ileti için benzersiz bir grup tanıtcısı oluşturabilmesi için, uygulamanın *GroupId* alanını *her bir* MQPUT ya da MQPUT1 çağrıdan önce MQGI_NONE olarak sıfırlaması gerekir. Bu yapılmazsa, ilgisiz iletiler aynı grup tanıtcısına sahip olabilir ve bu daha sonra yanlış işleme neden olabilir. *GroupId* alanının ne zaman sıfırlanacağı hakkında daha fazla bilgi için *GroupId* alanının ve MQPMO_LOGICAL_ORDER seçeneğinin açıklamalarına bakın.

Kuyruk yöneticisi, kesimlerin (ve gerekli üstbilgi verilerinin) kuyruğa sığması için iletileri gerektiği şekilde bölümlere ayırır. Ancak, kuyruk yöneticisi tarafından oluşturulan bir kesimin büyüklüğü için alt sınır vardır ve yalnızca bir iletiden yaratılan son kesim bu sınırdan daha küçük olabilir (uygulama tarafından oluşturulan kesimin büyüklüğü için alt sınır bir byte). Kuyruk yöneticisi tarafından oluşturulan bölümler eşit uzunlukta olmayabilir. Kuyruk yöneticisi iletiyi aşağıdaki gibi işler:

- Kullanıcı tanımlı biçimler 16 baytın katları olan sınırlarda bölünür; kuyruk yöneticisi 16 bayttan küçük kesimler oluşturmaz (son kesim dışında).
- MQFMT_STRING dışındaki yerleşik biçimler, var olan verilerin niteliğine uygun noktalarda bölünür. Ancak, kuyruk yöneticisi hiçbir zaman bir IBM MQ üstbilgi yapısının ortasında bir iletiyi bölmez. Bu, tek bir MQ üstbilgi yapısını içeren bir kesimin kuyruk yöneticisi tarafından daha fazla bölünemeyeceği ve sonuç olarak, bu ileti için olası kesim büyüklüğü alt sınırının 16 byte 'tan fazla olduğu anlamına gelir.

Kuyruk yöneticisi tarafından oluşturulan ikinci ya da sonraki bölüm aşağıdakilerden biriyle başlar:

- MQ üstbilgi yapısı
- Uygulama ileti verilerinin başlangıcı
- Uygulama ileti verilerinde yolun bir parçası
- MQFMT_STRING, var olan verilerin niteliğine bakılmaksızın bölünür (SBCS, DBCS ya da karma SBCS/DBCS). Dizilim DBCS ya da karışık SBCS/DBCS olduğunda, bu, bir karakter kümesinden diğerine dönüştürülemeyen bölümlerle sonuçlanabilir. Kuyruk yöneticisi MQFMT_STRING iletilerini hiçbir zaman 16 byte 'tan küçük bölümlere ayırmaz (son kısım dışında).
- Kuyruk yöneticisi, her kesimin MQMD ' deki *Format*, *CodedCharSetId* ve *Encoding* alanlarını, kesimin *başlangıcında* bulunan verileri doğru olarak tanımlamak üzere ayarlar; biçim adı yerleşik bir biçimin adı ya da kullanıcı tanımlı bir biçimin adıdır.
- *Offset* sıfırdan büyük bölümlere sahip MQMD ' deki *Report* alanı değiştirilir. Her rapor tipi için, rapor seçeneği MQRO_*_WITH_DATA ise, ancak bölüm kullanıcı verilerinin ilk 100 baytından herhangi birini (yani IBM MQ üstbilgi yapılarını izleyen veriler) içermeyse, rapor seçeneği MQRO_* olarak değiştirilir.

Kuyruk yöneticisi yukarıdaki kuralları izler, ancak aksi takdirde iletileri beklenmedik şekilde böler; iletinin nerede bölündüğü konusunda varsayımında bulunmayın.

Kalıcı iletiler için, kuyruk yöneticisi yalnızca bir iş birimi içinde bölümlene gerçekleştirebilir:

- MQPUT ya da MQPUT1 çağrısı kullanıcı tanımlı bir iş birimi içinde çalışıyorsa, o iş birimi kullanılır. Bölümlene işlemi sırasında çağrı başarısız olursa, kuyruk yöneticisi, başarısız olan çağrı nedeniyle kuyruğa yerleştirilen bölümleri kaldırır. Ancak, başarısızlık iş biriminin başarıyla kesinleştirilmesini engellemez.
- Çağrı kullanıcı tanımlı bir iş birimi dışında çalışıyorsa ve kullanıcı tanımlı bir iş birimi yoksa, kuyruk yöneticisi çağrı süresi boyunca bir iş birimi yaratır. Çağrı başarılı olursa, kuyruk yöneticisi iş birimini otomatik olarak kesinleştirir. Çağrı başarısız olursa, kuyruk yöneticisi iş birimini geri alır.
- Çağrı kullanıcı tanımlı bir iş biriminin dışında çalışıyorsa, ancak kullanıcı tanımlı bir iş birimi varsa, kuyruk yöneticisi bölümlene gerçekleştiremez. İleti bölümlene gerektirmiyorsa, çağrı yine de başarılı olabilir. Ancak ileti bölümlene gerektiriyorsa, çağrı MQRC_UOW_NOT_KULLANILABILIR neden koduyla başarısız olur.

Kalıcı olmayan iletiler için kuyruk yöneticisi, bölümlene gerçekleştirmek için bir iş biriminin kullanılabilir olmasını gerektirmez.

Bölümlenmiş olabilecek iletilerdeki verileri dönüştürürken dikkatli olun:

- Alan uygulama MQGET çağrısındaki verileri dönüştürürse ve MQGMO_COMPLETE_MSG seçeneğini belirtirse, veri dönüştürme çıkışına çıkışın dönüştürülmesi için tam ileti geçirilir ve iletinin bölümlenmiş olması çıkışa görünür.
- Alan uygulama bir kerede bir bölüm alırsa, veri dönüştürme çıkışı bir kerede bir bölümü dönüştürmek için çağrılır. Bu nedenle çıkış, diğer bölümlerdeki verilerden bağımsız olarak bir kesimdeki verileri dönüştürmelidir.

İletideki verilerin niteliği, 16 baytlık sınırlardaki verilerin isteğe bağlı olarak bölümlenmesinin, çıkış tarafından dönüştürülemez bölümlere ya da MQFMT_STRING ve karakter kümesi DBCS ya da SBCS/DBCS ise, gönderen uygulamanın daha fazla bölümlenmeyi engellemek için MQMF_SEGMENTATION_INHIBE edilen bölümleri yaratması ve yerleştirmesi gerekir. Bu şekilde, gönderen uygulama, her bir kesimin veri dönüştürme çıkışının kesimi başarıyla dönüştürmesini sağlamak için yeterli bilgi içerdiğinden emin olabilir.

- Gönderen ileti kanalı aracısı (MCA) için gönderen dönüştürmesi belirtilirse, MCA yalnızca mantıksal ileti kesimleri olmayan iletileri dönüştürür; MCA hiçbir zaman bölüm olan iletileri dönüştürmeye çalışmaz.

Bu işaret, MQPUT ve MQPUT1 çağrılarında bir giriş işareti ve MQGET çağrısında bir çıkış işaretiyle gösterilir. İkinci çağrıda kuyruk yöneticisi, işaretin değerini MQGMO 'daki *Segmentation* alanına yansıtır.

Bu işaretin ilk değeri MQMF_SEGMENTATION_INHIBE.

Durum işaretleri: Fiziksel iletinin bir ileti grubuna ait olup olmadığını, mantıksal iletinin bir parçası olup olmadığını gösteren işaretler. MQPUT ya da MQPUT1 çağrısında aşağıdakilerden biri ya da daha fazlası belirtilebilir ya da MQGET çağrısı tarafından döndürülebilir:

MQMF_MSG_IN_GROUP

İleti bir grubun üyesi.

MQMF_LAST_MSG_IN_GROUP

İleti, bir gruptaki son mantıksal iletidir.

Bu işaret ayarlanırsa, kuyruk yöneticisi iletiyle birlikte gönderilen MQMD kopyasında MQMF_MSG_IN_GROUP 'u açar, ancak MQPUT ya da MQPUT1 çağrısında uygulama tarafından sağlanan MQMD' de bu işaretlerin ayarlarını değiştirmez.

Bir grubun tek bir mantıksal iletiden oluşması geçerlidir. Bu durumda, MQMF_LAST_MSG_IN_GROUP ayarlanır, ancak *MsgSeqNumber* alanı bir değerine sahiptir.

MQMF_SEGMENT

İleti, mantıksal iletinin bir bölümdür.

MQMF_LAST_SEGMENT olmadan MQMF_SEGMENT belirtildiğinde, bölümdeki uygulama iletisi verilerinin uzunluğu (var olabilecek herhangi bir IBM MQ üstbilgi yapısının uzunlukları *hariç*) en az bir olmalıdır. Uzunluk sıfır ise, MQPUT ya da MQPUT1 çağrısı MQRC_SEGMENT_LENGTH_ZERO neden koduyla başarısız olur.

z/OS işletim sisteminde, ileti MQIT_GROUP_ID dizin tipine sahip bir kuyruğa konuluyorsa bu seçenek desteklenmez.

MQMF_LAST_SEGMENT

İleti, mantıksal iletinin son bölümüdür.

Bu işaret ayarlanırsa, kuyruk yöneticisi iletiyle birlikte gönderilen MQMD kopyasında MQMF_SEGMENT 'i açar, ancak MQPUT ya da MQPUT1 çağrısında uygulama tarafından sağlanan MQMD' de bu işaretlerin ayarlarını değiştirmez.

Mantıksal bir ileti yalnızca bir bölümden oluşabilir. Bu durumda, MQMF_LAST_SEGMENT ayarlanır, ancak *Offset* alanı sıfır değerine sahiptir.

MQMF_LAST_SEGMENT belirtildiğinde, bölümdeki uygulama iletisi verilerinin uzunluğu (var olabilecek üstbilgi yapılarının uzunlukları *hariç*) sıfır olabilir.

z/OS işletim sisteminde, ileti MQIT_GROUP_ID dizin tipine sahip bir kuyruğa konuluyorsa bu seçenek desteklenmez.

Uygulama, iletiler yerleştirilirken bu işaretlerin doğru ayarlandığından emin olmalıdır. MQPMO_LOGICAL_ORDER belirtildiyse ya da kuyruk tanıtıcısı için önceki MQPUT çağrısında belirtildiyse, işaretlerin ayarları kuyruk tanıtıcısı için kuyruk yöneticisi tarafından tutulan grup ve bölüm bilgileriyle tutarlı olmalıdır. MQPMO_LOGICAL_ORDER belirtildiğinde kuyruk tanıtıcısı için *ardışık* MQPUT çağrıları için aşağıdaki koşullar geçerlidir:

- Geçerli bir grup ya da mantıksal ileti yoksa, tüm bu işaretler (ve bunların birleşimleri) geçerlidir.
- MQMF_MSG_IN_GROUP belirtildikten sonra, MQMF_LAST_MSG_IN_GROUP belirleninceye kadar açık kalmalıdır. Bu koşul karşılanmazsa, çağrı MQRC_INCOMPLETE_GROUP neden koduyla başarısız olur.
- MQMF_SEGMENT belirtildikten sonra, MQMF_LAST_SEGMENT belirtilinceye kadar açık kalmalıdır. Bu koşul karşılanmazsa, çağrı MQRC_INCOMPLETE_MSG neden koduyla başarısız olur.
- MQMF_MSG_IN_GROUP olmadan MQMF_SEGMENT belirtildikten sonra, MQMF_MSG_IN_GROUP, MQMF_LAST_SEGMENT belirtilinceye kadar *kapalı* olarak kalmalıdır. Bu koşul karşılanmazsa, çağrı MQRC_INCOMPLETE_MSG neden koduyla başarısız olur.

Kuyruktaki fiziksel sıra, işaretlerin geçerli birleşimlerini ve çeşitli alanlar için kullanılan değerleri gösterir.

Bu işaretler, MQPUT ve MQPUT1 çağrılarında ilişkin giriş işaretleri ve MQGET çağrısına ilişkin çıkış işaretleridir. İkinci çağrıda, kuyruk yöneticisi işaretlerin değerlerini MQGMO 'daki *GroupStatus* ve *SegmentStatus* alanlarına da yansıtır.

Yayınlama/Abone Olma ile gruplanmış ya da bölümlenmiş iletileri kullanamazsınız.

Varsayılan işaretler: İletinin varsayılan öznitelikleri olduğunu belirtmek için aşağıdaki belirtilebilir:

MQMF_NONE

İleti işareti yok (varsayılan ileti öznitelikleri).

Bu, kesimlere ayırmayı engeller ve iletinin bir grupta olmadığını ve mantıksal iletinin bir parçası olmadığını gösterir. MQMF_NONE, program belgelerine yardımcı olmak için tanımlanır. Bu işaretin başka bir işaretiyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

MsgFlags alanı alt alanlara bölümlenir; ayrıntılar için bkz. "Rapor seçenekleri ve ileti işaretleri" sayfa 891.

Bu alanın ilk değeri MQMF_NONE. *Version*, MQMD_VERSION_2' den küçükse bu alan yoksayılr.

MQMD için OriginalLength (MQLONG)

Bu alan yalnızca bölüm olan rapor iletileri için geçerlidir. Rapor iletilerinin ilişkili olduğu ileti bölümünün uzunluğunu belirler; kesimin bir parçasını oluşturduğu mantıksal iletinin uzunluğunu ya da rapor iletilerindeki verilerin uzunluğunu belirtmez.

Not: Bölüm olan bir ileti için bir rapor iletilisi oluştururken, kuyruk yöneticisi ve ileti kanalı aracısı, özgün iletideki *GroupId*, *MsgSeqNumber*, *Offset* ve *MsgFlags* alanlarını rapor iletilisi için MQMD 'ye kopyalar. Sonuç olarak, rapor iletilisi de bir bölümdür. Rapor iletileri oluşturan uygulamalar da aynısını yapmalı ve *OriginalLength* alanını doğru şekilde ayarlamalıdır.

Aşağıdaki özel değer tanımlandı:

MQOL_UNDEFINED

İletinin özgün uzunluğu tanımlanmadı.

OriginalLength , MQPUT ve MQPUT1 çağrılarındaki bir giriş alanıdır, ancak uygulamanın sağladığı değer yalnızca belirli durumlarda kabul edilir:

- Konan ileti bir bölümse ve aynı zamanda bir rapor iletilisiyse, kuyruk yöneticisi belirtilen değeri kabul eder. Değer şöyle olmalıdır:
 - Kesim son kesim değilse sıfırdan büyük
 - Kesim son kesim ise sıfırdan küçük değil
 - İletide var olan verilerin uzunluğundan az değil

Bu koşullar karşılanmazsa, çağrı MQRC_ORIGINAL_LENGTH_ERROR neden koduyla başarısız olur.

- Yerleştirilmekte olan ileti bir bölümse, ancak bir rapor iletilisi değilse, kuyruk yöneticisi alanı yoksayar ve bunun yerine uygulama iletilisi verilerinin uzunluğunu kullanır.
- Diğer tüm durumlarda, kuyruk yöneticisi alanı yoksayar ve bunun yerine MQOL_UNDEFINED değerini kullanır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır.

Bu alanın ilk değeri MQOL_UNDEFINED. *Version* , MQMD_VERSION_2' den küçükse bu alan yoksayılr.

MQMDE-İleti tanımlayıcı uzantısı

MQMDE yapısı, bazen uygulama iletilisi verilerinden önce gelen verileri tanımlar. Yapı, version-2 MQMD 'de var olan, ancak version-1 MQMD' de bulunmayan MQMD alanlarını içeriyor.

Kullanılabilirlik

Tüm IBM MQ sistemleri ve bu sistemlere bağlı IBM MQ MQI clients .

Biçim Adı

MQFMT_MD_EXTENSION

Karakter kümesi ve kodlama

MQMDE içindeki veriler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır; bunlar, C programlama dili için **CodedCharSetId** kuyruk yöneticisi özniteliği ve MQENC_NATIVE ile verilir.

MQMDE ' nin karakter kümesini ve kodlamasını aşağıdaki *CodedCharSetId* ve *Encoding* alanlarına ayarlayın:

- MQMD (MQMDE yapısı ileti verilerinin başındadır) ya da
- MQMDE yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

MQMDE, kuyruk yöneticisinin karakter kümesinde ve kodlamasında yoksa, MQMDE kabul edilir, ancak yerine getirilmezse, MQMDE ileti verisi olarak işlenir.

Not: Windows üzerinde, Micro Focus COBOL ile derlenen uygulamalar, kuyruk yöneticisinin kodlamasından farklı bir MQENC_NATIVE değeri kullanır. MQPUT, MQPUT1 ve MQGET çağrılarındaki MQMD yapısındaki

sayısal alanların Micro Focus COBOL kodlamasında olması gerekse de, MQMDE yapısındaki sayısal alanların kuyruk yöneticisinin kodlamasında olması gerekir. Bu ikincisi, C programlama dili için MQENC_NATIVE tarafından verilir ve 546 değerine sahiptir.

Kullanım

version-2 MQMD kullanan uygulamalar bir MQMDE yapısıyla karşılaşmaz. Ancak, version-1 MQMD 'yi kullanmaya devam eden özel uygulamalar ve uygulamalar bazı durumlarda MQMDE ile karşılaşabilir. MQMDE yapısı aşağıdaki durumlarda oluşabilir:

- MQPUT ve MQPUT1 çağrılarında belirtildi
- MQGET çağrısı tarafından döndürüldü
- İletim kuyruklarındaki iletilerde

MQPUT ve MQPUT1 çağrılarında MQMDE belirtildi

MQPUT ve MQPUT1 çağrılarında uygulama bir version-1 MQMD sağlarsa, uygulama isteğe bağlı olarak ileti verilerine bir MQMDE öneki gönderebilir ve MQMDE 'nin var olduğunu göstermek için MQMD' deki *Format* alanını MQFMT_MD_EXTENSION olarak ayarlayabilir. Uygulama bir MQMDE sağlamazsa, kuyruk yöneticisi MQMDE ' deki alanlar için varsayılan değerleri varsayar. Kuyruk yöneticisinin kullandığı varsayılan değerler, yapının ilk değerleriyle aynıdır; bkz. [Çizelge 503 sayfa 470](#).

Uygulama bir version-2 MQMD sağlarsa ve , uygulama iletileri verilerini MQMDE ile öneklerse, yapılar aşağıdaki çizelgede gösterildiği gibi işlenir.

MQMD sürümü	version-2 alanlarının değerleri	MQMDE içindeki ilgili alanların değerleri	Kuyruk yöneticisi tarafından gerçekleştirilen işlem
1	-	Geçerli	MQMDE onurlandırıldı
2	Varsayılan	Geçerli	MQMDE onurlandırıldı
2	Varsayılan değil	Geçerli	MQMDE, ileti verisi olarak kabul edilir
1 ya da 2	Herhangi Biri	Geçerli değil	Arama uygun bir neden koduyla başarısız oluyor
1 ya da 2	Herhangi Biri	MQMDE yanlış karakter kümesinde ya da kodlamada ya da desteklenmeyen bir sürümde	MQMDE, ileti verisi olarak kabul edilir

Not: z/OS sistemlerinde, uygulama MQMDE içeren bir version-1 MQMD 'sini belirtiyorsa, kuyruk yöneticisi MQMDE ' yi yalnızca kuyrukta *IndexType* MQIT_GROUP_ID varsa doğrular.

Özel bir durum var. Uygulama bir bölüm (yani, MQMF_SEGMENT ya da MQMF_LAST_SEGMENT işareti) olan bir iletiyi koymak için version-2 MQMD kullanıyorsa ve MQMD ' deki biçim adı MQFMNT_DEAD_LETTER_HEADER ise, kuyruk yöneticisi bir MQMDE yapısı oluşturur ve bunu MQDLH yapısı ile onu izleyen veriler arasına *ekler* . Kuyruk yöneticisinin iletiyle birlikte sakladığı MQMD ' de version-2 alanları varsayılan değerlerine ayarlanır.

version-2 MQMD 'de var olan, ancak version-1 MQMD' de olmayan alanlar, MQPUT ve MQPUT1' de giriş/çıkış alanlarıdır. Ancak, kuyruk yöneticisi MQPUT ve MQPUT1 çağrılarının çıkışında MQMDE ' deki eşdeğer alanlarda herhangi bir değer döndürmez; uygulama bu çıkış değerlerini gerektiriyorsa, bir version-2 MQMD kullanmalıdır.

MQGET çağrısı tarafından döndürülen MQMDE

MQGET çağrısında, uygulama bir version-1 MQMD sağlarsa, kuyruk yöneticisi MQMDE ile döndürülen iletiyi örnekler, ancak MQMDE 'deki alanlardan birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa. Kuyruk yöneticisi, MQMD 'deki *Format* alanını MQMDE' nin var olduğunu göstermek için MQFMT_MD_EXTENSION değerine ayarlar.

Uygulama **Buffer** değiştirgesinin başında bir MQMDE sağlarsa, MQMDE yoksayıdır. MQGET çağrısından döndüğünüzde, ileti için MQMDE ile değiştirilir (gerekliyse) ya da uygulama ileti verileri tarafından üzerine yazılır (MQMDE gerekli değilse).

MQGET çağrısı bir MQMDE döndürürse, MQMDE 'deki veriler genellikle kuyruk yöneticisinin karakter kümesi ve kodlamasında bulunur. Ancak, MQMDE aşağıdaki durumda başka bir karakter kümesi ve kodlamada olabilir:

- MQMDE, MQPUT ya da MQPUT1 çağrısında veri olarak kabul edildi (buna neden olabilecek durumlar için bkz. Çizelge 502 sayfa 469).
- İleti, TCP bağlantısıyla bağlanan bir uzak kuyruk yöneticisinden alındı ve alan ileti kanalı aracısı (MCA) doğru ayarlanmadı.

Not: Windows üzerinde, Micro Focus COBOL ile derlenen uygulamalar, kuyruk yöneticisinin kodlamasından farklı bir MQENC_NATIVE değeri kullanır (yukarıya bakın).

İletim kuyruklarındaki iletilerde MQMDE

İletim kuyruklarındaki iletilerin başına, içinde version-1 MQMD içeren MQXQH yapısı eklenir. MQMDE, MQXQH yapısı ve uygulama iletileri arasında konumlandırılmış olarak da var olabilir; ancak, MQMDE 'deki alanlardan birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa bu durum genellikle var olur.

MQXQH yapısı ile uygulama iletileri arasında diğer MQ üstbilgi yapıları da oluşabilir. Örneğin, gitmeyen harf üstbilgisi MQDLH varsa ve ileti bir bölüm değilse, sıralama şöyle olur:

- MQXQH (version-1 MQMD 'si içerir)
- MQMDE
- MQDLH
- uygulama iletileri verileri

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 503. MQMDE için MQMDE içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQMDE_STRUC_ID	'MDE~'
<u>Sürüm</u> (yapı sürümü numarası)	MQMDE_VERSION_2	2
<u>StrucLength</u> (MQMDE yapısının uzunluğu)	MQMDE_LENGTH_2	72
Kodlama (MQMDE 'yi izleyen verilerin sayısal kodlaması)	MQENC_NATIVE	Ortama bağlıdır
<u>CodedCharSetId</u> (MQMDE 'yi izleyen verilerin karakter kümesi tanıtıcısı)	MQCCSI_AYRILMAMIŞ	0
<u>Biçim</u> (MQMDE 'yi izleyen verilerin biçim adı)	MQFMT_NONE	Boşluklar
<u>İşaretler</u> (genel işaretler)	MQMDEF_NONE	0

Çizelge 503. MQMDE için MQMDE içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
GroupId (grup tanıtıcısı)	MQGI_NONE	Boş Değerler
MsgSeqNumber (grup içindeki mantıksal iletinin sıra numarası)	Yok	1
Görelî Konum (mantıksal iletinin başlangıcından fiziksel iletideki verilerin görelî konumu)	Yok	0
MsgFlags (ileti işaretleri)	MQMF_NONE	0
OriginalLength (özgün iletinin uzunluğu)	MQOL_UNDEFINED	-1

Notlar:

- simgesi, tek bir boş karakteri temsil eder.
- C programlama dilinde, makro değişkeniMQMDE_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQMDE MyMDE = {MQMDE_DEFAULT};
```

Dil bildirimleri

MQMDE için C bildirim

```
typedef struct tagMQMDE MQMDE;
struct tagMQMDE {
    MQCHAR4    StructId;        /* Structure identifier */
    MQLONG     Version;        /* Structure version number */
    MQLONG     StructLength;    /* 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 */
};
```

MQMDE için COBOL bildirim

```
** 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-STRUCLNGTH  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.

```

MQMDE için PL/I bildirimi

```

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 */

```

MQMDE için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

MQMDE          DSECT
MQMDE_STRUCID  DS   CL4   Structure identifier
MQMDE_VERSION  DS   F     Structure version number
MQMDE_STRUCLNGTH 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)

```

MQMDE için Visual Basic bildirimi

```

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'
  CodedCharSetId As Long   'Character-set identifier of data that'
  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'
  Offset       As Long     'Offset of data in physical message from'
  MsgFlags     As Long     'Message flags'

```


OriginalLength As Long 'Length of original message'
End Type

MQMDE için StrucId (MQCHAR4)

Bu, ileti tanımlayıcı uzantı yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQMDE_STRUC_ID.

Değer şöyle olmalıdır:

MQMDE_STRUC_ID

İleti tanımlayıcı uzantı yapısına ilişkin tanıtıcı.

C programlama dili için, MQMDE_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQMDE_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQMDE için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQMDE_VERSION_2

Version-2 ileti tanımlayıcısı uzantı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQMDE_CURRENT_VERSION

İleti tanımlayıcı uzantı yapısının yürürlükteki sürümü.

Bu alanın ilk değeri MQMDE_VERSION_2' dir.

MQMDE için StrucLength (MQLONG)

Bu, MQMDE yapısının uzunluğudur; aşağıdaki değer tanımlanır:

MQMDE_LENGTH_2

version-2 ileti tanımlayıcısı uzantı yapısının uzunluğu.

Bu alanın ilk değeri MQMDE_LENGTH_2' dir.

MQMDE için kodlama (MQLONG)

MQMDE yapısını izleyen verilerin sayısal kodlamasını belirtir; MQMDE yapısının kendisindeki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Kuyruk yöneticisi, alanın geçerli olup olmadığını denetlemez. Veri kodlamasıyla ilgili ek bilgi için [“MQMD-İleti tanımlayıcı” sayfa 420](#) içinde açıklanan *Encoding* alanına bakın.

Bu alanın ilk değeri MQENC_NATIVE.

MQMDE için CodedCharSetId (MQLONG)

MQMDE yapısını izleyen verilerin karakter kümesi tanıtıcısını belirtir; MQMDE yapısının kendisindeki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Kuyruk yöneticisi bu alanın geçerli olup olmadığını denetlemez. Aşağıdaki özel değer kullanılabilir:

MQCCSI_INHERIT

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı MQCCSI_INHERIT değerini döndürmez.

MQMD 'deki *PutApplType* alanının değeri MQAT_BROKER ise, MQCSI_INHERIT kullanılamaz.

Bu değer aşağıdaki ortamlarda desteklenir:

-  AIX

-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağı IBM MQ istemcileri için.

Bu alanın ilk değeri MQCCSI_UNDEFINED 'dir.

MQMDE için biçim (MQCHAR8)

MQMDE yapısını izleyen verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Kuyruk yöneticisi bu alanın geçerli olup olmadığını denetlemez. Biçim adlarıyla ilgili ek bilgi için [“MQMD-İleti tanımlayıcı” sayfa 420](#) içinde açıklanan *Format* alanına bakın.

Bu alanın ilk değeri MQFMT_NONE.

MQMDE için İşaretler (MQLONG)

Aşağıdaki işaret belirtilebilir:

MQMDEF_NONE

Bayrak yok.

Bu alanın ilk değeri MQMDEF_NONE.

MQMDE için GroupId (MQBYTE24)

[“MQMD-İleti tanımlayıcı” sayfa 420](#) içinde açıklanan *GroupId* alanına bakın. Bu alanın ilk değeri MQGI_NONE.

MsgSeqMQMDE için sayı (MQLONG)

[“MQMD-İleti tanımlayıcı” sayfa 420](#) içinde açıklanan *MsgSeqNumber* alanına bakın. Bu alanın ilk değeri 1 'dir.

MQMDE için görelî konum (MQLONG)

[“MQMD-İleti tanımlayıcı” sayfa 420](#) içinde açıklanan *Offset* alanına bakın. Bu alanın ilk değeri 0 'dir.

MQMDE için MsgFlags (MQLONG)

[“MQMD-İleti tanımlayıcı” sayfa 420](#) içinde açıklanan *MsgFlags* alanına bakın. Bu alanın ilk değeri MQMF_NONE.

MQMDE için OriginalLength (MQLONG)

[“MQMD-İleti tanımlayıcı” sayfa 420](#) içinde açıklanan *OriginalLength* alanına bakın. Bu alanın ilk değeri MQOL_UNDEFINED.

MQMHBO-Arabellek seçeneklerine ileti tanıtıcısı

MQMHBO yapısı, uygulamaların, arabelleklerin ileti tanıtıcılarından nasıl üretildiğini denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQMHBUF çağrısındaki bir giriş değıştirgesidir.

Karakter kümesi ve kodlama

MQMHBO içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında (MQENC_NATIVE) yer almalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 504. MQMHBO 'daki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQMHBO_STRUC_ID	'MHBO'
Sürüm (yapı sürümü numarası)	MQMHBO_VERSION_1	1
Seçenekler (MQMHBUF işlemini denetleyen seçenekler)	MQMHBO_PROPERTIES_I N_MQRFH2	

Notlar:

- Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.
- C programlama dilinde, makro değişkeni MQMHBO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQMHBO MyMHBO = {MQMHBO_DEFAULT};
```

Dil bildirimleri

MQMHBO için C bildirimini

```
typedef struct tagMQMHBO MQMHBO;  
struct tagMQMHBO {  
    MQCHAR4 StrucId;          /* Structure identifier */  
    MQLONG Version;          /* Structure version number */  
    MQLONG Options;          /* Options that control the action of  
                             MQMHBUF */  
};
```

MQMHBO için COBOL bildirimini

```
** 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.
```

MQMHBO için PL/I bildirimini

```
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 */
```

MQMHBO için High Level Assembler (Üst Düzey Çevirici) bildirimini

```
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
MQMHBO_AREA

EQU *-MQMHBO
DS CL (MQMHBO_LENGTH)

MQMHBO için StrucId (MQCHAR4)

Bu, arabellek seçenekleri yapısına ilişkin ileti tanıtıcısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQMHBO_STRUC_ID.

Değer şöyle olmalıdır:

MQMHBO_STRUC_ID

Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısı tanıtıcısı.

C programlama dili için, MQMHBO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQMHBO_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQMHBO için Sürüm (MQLONG)

Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısı-Sürüm alanı

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MQMHBO_VERSION_1

Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısının sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQMHBO_CURRENT_VERSION

Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQMHBO_VERSION_1' dir.

MQMHBO için Seçenekler (MQLONG)

İleti tanıtıcısı-arabellek seçenekleri yapısı-Seçenekler alanı

Bu seçenekler MQMHBUF işlemini denetler.

Aşağıdaki seçeneği belirtmeniz gerekir:

MQMHBO_PROPERTIES_IN_MQRFH2

Özellikleri bir ileti tanıtıcısından arabelleğe dönüştürürken, bunları MQRFH2 biçimine dönüştürün.

İsteğe bağlı olarak, aşağıdaki seçeneği de belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyse).

MQMHBO_DELETE_PROPERTIES

Arabelleğe eklenen özellikler ileti tanıtıcısından silinir. Arama başarısız olursa, hiçbir özellik silinmez.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: MQMHBO_PROPERTIES_IN_MQRFH2.

MQOD-Nesne tanımlayıcı

MQOD yapısı, ada göre bir nesne belirtmek için kullanılır. Yapı, MQOPEN ve MQPUT1 çağrılarında bir giriş/çıkış değiştirgesidir.

Aşağıdaki nesne tipleri geçerlidir:

- Kuyruk ya da dağıtım listesi
- Ad listesi
- Süreç tanımlaması
- Kuyruk yöneticisi
- Konu

Kullanılabilirlik

Tüm IBM MQ sistemleri ve bu sistemlere bağlı IBM MQ MQI clients .

Sürüm

Yürürlükteki MQOD sürümü: MQOD_VERSION_4. Birkaç ortam arasında bağlantı sağlamak istediğiniz uygulamalar, ilgili tüm ortamlarda MQOD ' un gerekli sürümünün desteklendiğinden emin olmalıdır. Yapının yalnızca daha yeni sürümlerinde bulunan alanlar, izleyen açıklamalarda bu şekilde tanımlanır.

Desteklenen programlama dilleri için sağlanan üstbilgi, COPY ve INCLUDE dosyaları, ortam tarafından desteklenen ancak *Version* alanının ilk değeri MQOD_VERSION_1 olarak ayarlanmış MQOD ' un en son sürümünü içerir. version-1 yapısında var olmayan alanları kullanmak için uygulamanın *Version* alanını gerekli sürümün sürüm numarasına ayarlaması gerekir.

Bir dağıtım listesini açmak için *Version* , MQOD_VERSION_2 ya da daha büyük olmalıdır.

Karakter kümesi ve kodlama

MQOD içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında olmalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcisinin karakter kümesinde ve kodlamasında olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQOD_STRUC_ID	' 0D→→ '
<u>Sürüm</u> (yapı sürümü numarası)	MQOD_VERSION_1	1
<u>ObjectType</u> (nesne tipi)	MQOT_Q	1
<u>ObjectName</u> (nesne adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>ObjectQMgrAdı</u> (nesne kuyruğu yöneticisi adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>DynamicQName</u> (dinamik kuyruk adı)	Yok	' CSQ . * ' on z/OS ; ' AMQ . * ' tersi durumda
<u>AlternateUserKimliği</u> (diğer kullanıcı kimliği)	Yok	Boş değerli dizilim ya da boşluklar
Not: <i>Version</i> , MQOD_VERSION_2' den küçükse kalan alanlar yoksayılr.		
<u>RecsPresent</u> (var olan nesne kaydı sayısı)	Yok	0
<u>KnownDestSayı</u> (başarıyla açılan yerel kuyruk sayısı)	Yok	0
<u>UnknownDestCount</u> (başarıyla açılan uzak kuyruk sayısı)	Yok	0
<u>InvalidDestSayı</u> (açılmayan kuyrukların sayısı)	Yok	0
<u>ObjectRecGörelİ Konum</u> (MQOD ' un başlangıcından ilk nesne kaydının görelİ konumu)	Yok	0
<u>ResponseRecGörelİ Konumu</u> (MQOD ' un başlangıcından ilk yanıt kaydının görelİ konumu)	Yok	0
<u>ObjectRecPtr</u> (ilk nesne kaydının adresi)	Yok	Boş değerli gösterge ya da boş değerli byte

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>ResponseRecPtr</u> (ilk yanıt kaydının adresi)	Yok	Boş değerli gösterge ya da boş değerli byte
Not: <i>Version</i> , MQOD_VERSION_3' den küçükse kalan alanlar yoksayılr.		
<u>AlternateSecurityTanıtıcısı</u> (diğer güvenlik tanıtıcısı)	MQSID_YOK	Boş Değerler
<u>ResolvedQName</u> (çözölen kuyruk adı)	Yok	Boş değeri dizilim ya da boşluklar
<u>ResolvedQMgrAd</u> (çözömlenen kuyruk yöneticisi adı)	Yok	Boş değeri dizilim ya da boşluklar
Not: <i>Version</i> , MQOD_VERSION_4' den küçükse kalan alanlar yoksayılr.		
<u>ObjectString</u> (uzun nesne adı)	MQCHARV_DEFAULT	MQCHARV için tanımlandığı şekilde
<u>SelectionString</u> (seçim dizgisi)	MQCHARV_DEFAULT	MQCHARV için tanımlandığı şekilde
<u>ResObjectDizesi</u> (çözömlenen uzun nesne adı)	MQCHARV_DEFAULT	MQCHARV için tanımlandığı şekilde
<u>ResolvedType</u> (çözömlenen nesne tipi)	MQOT_NONE	0
Notlar: <ol style="list-style-type: none"> 1. ↪ simgesi, tek bir boş karakteri temsil eder. 2. Boş değeri dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir. 3. C programlama dilinde, makro değeri MQOD_DEFAULT, çizelgede listelenen değeri içerir. Yapıdaki alanlar için başlangıç değeri sağlamak üzere aşağıdaki şekilde kullanılabilir: <pre>MQOD MyOD = {MQOD_DEFAULT};</pre>		

Dil bildirimleri

MQOD için C bildirim

```
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 */
};

```

MQOD için COBOL bildirimi

```

** 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-OBJECTQMGRNAME        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-OBJECTRECPTTR          POINTER.
** Address of first response record
15 MQOD-RESPONSERECPTTR        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.

```

MQOD için PL/I bildirimi

```

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 */

```

MQOD için High Level Assembler (Üst Düzey Çevirici) bildirimi

MQOD	DSECT		
MQOD_STRUCTID	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_KNOWNDESTCOUNT	DS	F	Number of local queues opened
*			successfully
MQOD_UNKNOWNDSTCOUNT	DS	F	Number of remote queues opened


```

*
MQOD_INVALIDDESTCOUNT      DS    F    successfully
*
MQOD_OBJECTRECOFFSET        DS    F    Number of queues that failed to
*
MQOD_RESPONSERECOFFSET      DS    F    open
*
MQOD_OBJECTRECPTTR          DS    F    Offset of first object record from
*
MQOD_RESPONSERECPTR          DS    F    start of MQOD
*
MQOD_ALTERNATESECURITYID    DS    XL40  Offset of first response record
MQOD_RESOLVEDQNAME          DS    CL48  from start of MQOD
MQOD_RESOLVEDQMGRNAME       DS    CL48  Address of first object record
MQOD_OBJECTSTRING           DS    F    Address of first response record
MQOD_OBJECTSTRING_VSPTR     DS    F    XL40  Alternate security identifier
MQOD_OBJECTSTRING_VSOFFSET  DS    F    CL48  Resolved queue name
MQOD_OBJECTSTRING_VBUFSIZE  DS    F    CL48  Resolved queue manager name
MQOD_OBJECTSTRING_VSLNGTH   DS    F    Object Long name
MQOD_OBJECTSTRING_VSCCSID   DS    F    Address of variable length string
MQOD_OBJECTSTRING_LENGTH    EQU   *- MQOD_OBJECTSTRING
MQOD_OBJECTSTRING_AREA      ORG   MQOD_OBJECTSTRING
*
MQOD_SELECTIONSTRING        DS    F    CL(MQOD_OBJECTSTRING_LENGTH)
*
MQOD_SELECTIONSTRING_VSPTR  DS    F    Message Selector
MQOD_SELECTIONSTRING_VSOFFSET DS    F    Address of variable length string
MQOD_SELECTIONSTRING_VBUFSIZE DS    F    Offset of variable length string
MQOD_SELECTIONSTRING_VSLNGTH DS    F    size of buffer
MQOD_SELECTIONSTRING_VSCCSID DS    F    Length of variable length string
MQOD_SELECTIONSTRING_LENGTH EQU   *- MQOD_SELECTIONSTRING
MQOD_SELECTIONSTRING_AREA   ORG   MQOD_SELECTIONSTRING
*
MQOD_RESOBJECTSTRING        DS    F    CL(MQOD_SELECTIONSTRING_LENGTH)
*
MQOD_RESOBJECTSTRING_VSPTR  DS    F    Resolved Long object name
MQOD_RESOBJECTSTRING_VSOFFSET DS    F    Address of variable length string
MQOD_RESOBJECTSTRING_VBUFSIZE DS    F    Offset of variable length string
MQOD_RESOBJECTSTRING_VSLNGTH DS    F    size of buffer
MQOD_RESOBJECTSTRING_VSCCSID DS    F    Length of variable length string
MQOD_RESOBJECTSTRING_LENGTH EQU   *- MQOD_RESOBJECTSTRING
MQOD_RESOBJECTSTRING_AREA   ORG   MQOD_RESOBJECTSTRING
*
MQOD_RESOBJECTSTRING_AREA   DS    F    CL(MQOD_RESOBJECTSTRING_LENGTH)
MQOD_RESOLVEDTYPE           DS    F    Alias queue object resolved type
*
MQOD_LENGTH                 EQU   *-MQOD
MQOD_AREA                   ORG   MQOD
*
MQOD_AREA                   DS    CL(MQOD_LENGTH)

```

MQOD için Visual Basic bildirimi

```

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'
  UnknownDestCount As Long  'Number of remote queues opened'
  InvalidDestCount As Long  'successfully'
  ObjectRecOffset As Long   'Number of queues that failed to'
  ResponseRecOffset As Long  'open'
  ObjectRecPtr  As MQPTR    'Offset of first object record from'
  ResponseRecPtr As MQPTR    'start of MQOD'
  AlternateSecurityId As MQBYTE40 'Offset of first response record'
  ResolvedQName As String*48  'from start of MQOD'
  ResolvedQMgrName As String*48 'Address of first object record'
  End Type

```

MQOD için StrucId (MQCHAR4)

Bu, nesne tanımlayıcı yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQOD_STRUC_ID.

Değer şöyle olmalıdır:

MQOD_STRUC_ID

Nesne tanımlayıcı yapısına ilişkin tanıttıcı.

C programlama dili için, MQOD_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQOD_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQOD için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer aşağıdakilerden biri olmalıdır:

MQOD_VERSION_1

Version-1 nesne tanımlayıcı yapısı.

MQOD_VERSION_2

Version-2 nesne tanımlayıcı yapısı.

MQOD_VERSION_3

Version-3 nesne tanımlayıcı yapısı.

MQOD_VERSION_4

Version-4 nesne tanımlayıcı yapısı.

Tüm sürümler tüm IBM MQ V7.0 ortamlarında desteklenir.

Yalnızca yapının daha yeni sürümlerinde bulunan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQOD_CURRENT_VERSION

Nesne tanımlayıcı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQOD_VERSION_1' dir.

MQOD için ObjectType (MQLONG)

Nesne tanımlayıcıda adlandırılmakta olan nesnenin tipi. Olası değerler şunlardır:

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı. Nesnenin adı *ObjectName* alanında bulunur.

MQOT_Q

Sıraya girin. Nesnenin adı *ObjectName* alanında bulunur.

MQOT_NAMELIST

-Namelist. Nesnenin adı *ObjectName* alanında bulunur

MQOT_PROCESS

Süreç tanımlaması. Nesnenin adı *ObjectName* alanında bulunur

MQOT_Q_MGR

Kuyruk yöneticisi. Nesnenin adı *ObjectName* alanında bulunur

MQOT_KONU

Konu. Tam konu adı iki farklı alandan oluşturulabilir: *ObjectName* ve *ObjectString*.

Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQOT_Q.

MQOD için ObjectName (MQCHAR48)

Bu, *ObjectQMgrName* ile tanımlanan kuyruk yöneticisinde tanımlandığı şekilde, nesnenin yerel adıdır. Ad aşağıdaki karakterleri içerebilir:

- Büyük harf alfabetik karakterler (A-Z)
- Küçük harf alfabetik karakterler (a-z)
- Sayısal basamaklar (0-9)
- Nokta (.), eğik çizgi (/), altçizgi (_), yüzde (%)

Ad başta ya da gömülü boşluklar içermemelidir, ancak sonda boşluklar içerebilir. Addaki önemli verilerin sonunu göstermek için boş bir karakter kullanın; bu karakteri izleyen karakterler ve boş değer boşluk olarak değerlendirilir. Belirtilen ortamlarda aşağıdaki kısıtlamalar geçerlidir:

- EBCDIC Katakana kullanan sistemlerde küçük harfli karakterler kullanılamaz.
- z/OS'ta:
 - Alt çizgiyle başlayan ya da biten adlardan kaçınınız; bunlar işlemler ve denetim panoları tarafından işlenemez.
 - Yüzde karakterinin RACFiçin özel bir anlamı vardır. Dış güvenlik yöneticisi olarak RACF kullanılırsa, adlar yüzdeyi içermemelidir. Varsa, RACF sosyal tanımları kullanıldığında bu adlar güvenlik denetimlerine dahil edilmez.
- IBM üzerinde, komutlarda belirtildiğinde küçük harf, eğik çizgi ya da yüzde içeren adlar tırnak içine alınmalıdır. Bu tırnak işaretleri, yapılarda alan olarak ya da çağrılarda parametre olarak oluşan adlar için belirlenmemelidir.

Tam konu adı iki farklı alandan oluşturulabilir: *ObjectName* ve *ObjectString*. Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.

Aşağıdaki noktalar, belirtilen nesne tipleri için geçerlidir:

- *ObjectName* bir model kuyruğunun adıysa, kuyruk yöneticisi model kuyruğunun özniteliklerini içeren dinamik bir kuyruk yaratır ve *ObjectName* alanında yaratılan kuyruğun adını döndürür. Model kuyruğu yalnızca MQOPEN çağrısında belirtilebilir; MQPUT1 çağrısında model kuyruğu geçerli değil.
- *ObjectName*, TARGTYPE (TOPIC) ile bir diğer ad kuyruğunun adıysa, önce adı belirtilen diğer ad kuyruğunda bir güvenlik denetimi yapılır; diğer ad kuyrukları kullanıldığında bu olağandır. Güvenlik denetimi başarıyla tamamlandığında, MQOPEN çağrısı devam eder ve bir MQOT_TOPIC; bu, denetim konusu nesnesiyle ilgili bir güvenlik denetimi yapmayı içerir.
- *ObjectName* ve *ObjectQMGrName*, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun sahip olduğu bir paylaşılan kuyruğu tanımlarsa, yerel kuyruk yöneticisinde aynı ada sahip bir kuyruk tanımı da olmamalıdır. Böyle bir tanımlama (yerel kuyruk, diğer ad kuyruğu, uzak kuyruk ya da model kuyruğu) varsa, çağrı MQRC_OBJECT_NOT_UNIQUE neden koduyla başarısız olur.
- Açılmakta olan nesne bir dağıtım listesiye (yani, *RecsPresent* var ve sıfırdan büyük), *ObjectName* boş değerli ya da boş değerli olmalıdır. Bu koşul karşılanmazsa, çağrı MQRC_OBJECT_NAME_ERROR neden koduyla başarısız olur.
- *ObjectType* MQOT_Q_MGR ise, özel kurallar geçerli olur; bu durumda ad, alanın ilk boş karakterine ya da sonuna kadar tamamen boş olmalıdır.

Bu, *ObjectName* bir model kuyruğunun adı ve diğer tüm durumlarda yalnızca giriş alanı olduğunda MQOPEN çağrısına ilişkin bir giriş/çıkış alanıdır. Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

ObjectQMGrMQOD için Ad (MQCHAR48)

Bu, *ObjectName* nesnesinin tanımlandığı kuyruk yöneticisinin adıdır. Adda geçerli olan karakterler, *ObjectName* ile aynıdır (bkz. "[MQOD için ObjectName \(MQCHAR48\)](#)" sayfa 482). İlk boş değere kadar ya da alanın sonuna kadar tümüyle boş bırakılan bir ad, uygulamanın bağlı olduğu kuyruk yöneticisini (yerel kuyruk yöneticisi) gösterir.

Aşağıdaki noktalar, belirtilen nesne tipleri için geçerlidir:

- *ObjectType* MQOT_TOPIC, MQOT_NAMELIST, MQOT_PROCESS ya da MQOT_Q_MGR ise, *ObjectQMGrName* boş olmalı ya da yerel kuyruk yöneticisinin adı olmalıdır.
- *ObjectName* bir model kuyruğunun adıysa, kuyruk yöneticisi model kuyruğunun özniteliklerini içeren dinamik bir kuyruk yaratır ve *ObjectQMGrName* alanında kuyruğun yaratıldığı kuyruk yöneticisinin adını döndürür; bu, yerel kuyruk yöneticisinin adıdır. Model kuyruğu yalnızca MQOPEN çağrısında belirtilebilir; MQPUT1 çağrısında model kuyruğu geçerli değil.

- *ObjectName* bir küme kuyruğunun adıysa ve *ObjectQMgrName* boşsa, MQOPEN çağrısının döndürdüğü kuyruk tanıtıcısı kullanılarak gönderilen iletilerin hedefi, kuyruk yöneticisi (ya da kuruluysa, küme iş yükü çıkışı) tarafından aşağıdaki gibi seçilir:
 - MQOO_BIND_ON_OPEN belirtilirse, kuyruk yöneticisi MQOPEN çağrısını işlerken küme kuyruğunun belirli bir eşgörünümünü seçer ve bu kuyruk tanıtıcısı kullanılarak yerleştirilen tüm iletiler o yönetim ortamına gönderilir.
 - MQOO_BIND_NOT_FIXED belirtilirse, kuyruk yöneticisi bu kuyruk tanıtıcısını kullanan her ardışık MQPUT çağrısı için hedef kuyruğun farklı bir eşgörünümünü seçebilir (kümede farklı bir kuyruk yöneticisinde bulunur).

Uygulamanın bir küme kuyruğunun *belirli* bir eşgörünümüne (yani, kümedeki belirli bir kuyruk yöneticisinde bulunan bir kuyruk eşgörünümüne) ileti göndermesi gerekiyorsa, uygulamanın *ObjectQMgrName* alanında o kuyruk yöneticisinin adını belirtmesi gerekir. Bu, yerel kuyruk yöneticisini iletiyi belirtilen hedef kuyruk yöneticisine göndermeye zorlar.

- *ObjectName* , paylaşılan bir kuyruğun adıysa, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun iyeliğinde, *ObjectQMgrName* kuyruk paylaşım grubunun adı, yerel kuyruk yöneticisinin adı ya da boşluk olabilir; ileti, bu değerlerden herhangi biri belirtilirse aynı kuyruğa yerleştirilir.

Kuyruk paylaşım grupları yalnızca z/OS'ünde desteklenir.

- *ObjectName* uzak kuyruk paylaşım grubunun sahip olduğu bir paylaşılan kuyruğun adıysa (yerel kuyruk yöneticisinin ait olmadığı bir kuyruk paylaşım grubu), *ObjectQMgrName* kuyruk paylaşım grubunun adı olmalıdır. Söz konusu gruba ait bir kuyruk yöneticisinin adını kullanabilirsiniz, ancak ileti kuyruk paylaşım grubuna geldiğinde belirli bir kuyruk yöneticisi kullanılamıyorsa bu işlem iletiyi geciktirebilir.
- Açılmakta olan nesne bir dağıtım listesiye (yani, *RecsPresent* sıfırdan büyükse), *ObjectQMgrName* boş değerli ya da boş değerli bir dizgi olmalıdır. Bu koşul karşılanmazsa, çağrı MQRC_OBJECT_Q_MGR_NAME_ERROR neden koduyla başarısız olur.

Bu, *ObjectName* bir model kuyruğunun adı ve diğer tüm durumlarda yalnızca giriş alanı olduğunda MQOPEN çağrısına ilişkin bir giriş/çıkış alanıdır. Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQOD için DynamicQName (MQCHAR48)

MQOPEN çağrısıyla yaratılacak dinamik kuyruğun adıdır. Bu yalnızca *ObjectName* bir model kuyruğunun adını belirttiğinde geçerlidir; diğer tüm durumlarda *DynamicQName* yoksayılr.

Adda geçerli olan karakterler, yıldız işaretinin de geçerli olması dışında, *ObjectName* için geçerli olan karakterlerle aynıdır. *ObjectName* bir model kuyruğunun adıysa, boş olan bir ad (ya da ilk boş karakterden önce yalnızca boşluk olan bir ad) geçerli değildir.

Addaki son boşluk olmayan karakter bir yıldız işaretiyse (*), kuyruk yöneticisi yıldız işaretini, kuyruk için oluşturulan adın yerel kuyruk yöneticisinde benzersiz olmasını garanti eden bir karakter dizisiyle değiştirir. Bunun için yeterli sayıda karaktere izin vermek için, yıldız işareti yalnızca 1-33 arasındaki konumlarda geçerlidir. Yıldız işaretinden sonra boşluk ya da boş karakter dışında bir karakter olmamalıdır.

Yıldız işaretinin ilk karakter konumunda oluşması için geçerlidir; bu durumda ad yalnızca kuyruk yöneticisi tarafından oluşturulan karakterlerden oluşur.

z/OS' ta, otomatik olarak oluşturulan tam ada sahip bir kuyrukta güvenlik denetimi yapılamayacağı için, ilk karakter konumunda yıldız işareti olan bir ad kullanmayın.

Bu bir giriş alanıdır. Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri ortam tarafından belirlenir:

- z/OS'da değer 'CSQ.*' dir.
- Diğer platformlarda değer: 'AMQ.*' .

Değer, C dilinde boş sonlandırılmış bir dizgi ve diğer programlama dillerinde boşluk doldurulan bir dizedir.

AlternateUserId (MQCHAR12) for MQOD

MQOPEN çağrısı için MQOO_ALTERNATE_USER_AUTHORITY ya da MQPUT1 çağrısı için MQPMO_ALTERNATE_USER_AUTHORITY belirtirseniz, bu alan, uygulamanın çalışmakta olduğu kullanıcı kimliği yerine, açma yetkisini denetlemek için kullanılan alternatif bir kullanıcı kimliği içerir. Ancak, bazı denetimler yürürlükteki kullanıcı kimliğiyle (örneğin, bağlam denetimleri) gerçekleştirilmeye devam eder.

MQOO_ALTERNATE_USER_AUTHORITY ya da MQPMO_ALTERNATE_USER_AUTHORITY belirtilirse ve bu alan ilk boş karakter ya da alanın sonuna kadar tamamen boşsa, açma işlemi ancak belirtilen seçeneklerle bu nesneyi açmak için kullanıcı yetkisi gerekmediğinde başarılı olabilir.

MQOO_ALTERNATE_USER_AUTHORITY ya da MQPMO_ALTERNATE_USER_AUTHORITY belirtilmezse, bu alan yoksayılr.

Belirtilen ortamlarda aşağıdaki farklılıklar vardır:

- z/OS'da, açma yetkisini denetlemek için yalnızca *AlternateUserId* 'un ilk 8 karakteri kullanılır. Ancak, yürürlükteki kullanıcı kimliğinin bu diğer kullanıcı kimliğini belirtme yetkisi olmalıdır; bu denetim için diğer kullanıcı kimliğinin 12 karakterinin tümü de kullanılır. Kullanıcı kimliği yalnızca dış güvenlik yöneticisi tarafından izin verilen karakterleri içermelidir.

Bir kuyruk için *AlternateUserId* belirtilirse, bu değer, iletiler konduğunda kuyruk yöneticisi tarafından kullanılabilir. MQPUT ya da MQPUT1 çağrısında belirtilen MQPMO_*_CONTEXT seçenekleri kuyruk yöneticisinin kimlik bağlamı bilgilerini oluşturmasına neden olursa, kuyruk yöneticisi *AlternateUserId* ögesini iletinin MQMD 'sindeki *UserIdentifier* alanına, yürürlükteki kullanıcı kimliğinin yerine yerleştirir.

- Diğer ortamlarda, *AlternateUserId* yalnızca açılmakta olan nesneye ilişkin erişim denetimi denetimleri için kullanılır. Nesne bir kuyruksa, *AlternateUserId* , o kuyruk tanıtıcısı kullanılarak gönderilen iletilerin MQMD 'deki *UserIdentifier* alanının içeriğini etkilemez.

Bu bir giriş alanıdır. Bu alanın uzunluğu MQ_USER_ID_LENGTH tarafından verilir. Bu alanın ilk değeri C 'deki boş dizgi ve diğer programlama dillerindeki 12 boş karakterdir.

MQOD için RecsPresent (MQLONG)

Bu, uygulama tarafından sağlanan MQOR nesne kayıtlarının sayısıdır. Bu sayı sıfırdan büyükse, bir dağıtım listesinin açılmakta olduğunu ve *RecsPresent* 'un listedeki hedef kuyrukların sayısı olduğunu gösterir. Dağıtım listesi tek bir hedef içerebilir.

RecsPresent değeri sıfırdan küçük olmamalıdır ve sıfırdan büyükse *ObjectType* MQOT_Q olmalıdır; bu koşullar karşılanmazsa, çağrı MQRC_RECS_PRESENT_ERROR neden koduyla başarısız olur.

z/OS' ta bu alan sıfır olmalıdır.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQOD_VERSION_2' den küçükse bu alan yoksayılr.

KnownDestMQOD için sayı (MQLONG)

Bu, dağıtım listesindeki yerel kuyruklara çözülecek ve başarıyla açılan kuyrukların sayısıdır. Bu sayı, uzak kuyruklara çözülecek kuyrukları içermez (iletiyi saklamak için başlangıçta yerel bir iletim kuyruğu kullanılsa bile). Varsa, dağıtım listesinde olmayan tek bir kuyruk açılırken de bu alan ayarlanır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQOD_VERSION_1' den küçükse bu alan yoksayılr.

UnknownDestMQOD için sayı (MQLONG)

Bu, dağıtım listesindeki uzak kuyruklara çözülecek ve başarıyla açılacak kuyrukların sayısıdır. Varsa, dağıtım listesinde olmayan tek bir kuyruk açılırken de bu alan ayarlanır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQOD_VERSION_1' den küçükse bu alan yoksayılr.

InvalidDestMQOD için sayı (MQLONG)

Bu, dağıtım listesindeki, başarıyla açılmayan kuyrukların sayısıdır. Varsa, dağıtım listesinde olmayan tek bir kuyruk açılırken de bu alan ayarlanır.

Not: Varsa, bu alan yalnızca MQOPEN ya da MQPUT1 çağrısındaki **CompCode** değiştirgesi MQCC_OK ya da MQCC_WARNING; **CompCode** değiştirgesi MQCC_FAILED ise ayarlanmaz.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQOD_VERSION_1' den küçükse bu alan yoksayılr.

ObjectRecMQOD için görelı konum (MQLONG)

Bu, MQOD yapısının başlangıcından itibaren ilk MQOR nesne kaydının bayt cinsinden görelı konumudur. Görelı konum pozitif ya da negatif olabilir. *ObjectRecOffset* yalnızca bir dağıtım listesi açıldığında kullanılır. *RecsPresent* sıfırda alan yoksayılr.

Dağıtım listesi açılırken, dağıtım listesindeki hedef kuyrukların adlarını belirtmek için bir ya da daha fazla MQOR nesne kaydı dizisi sağlanmalıdır. Bu iki yoldan biriyle yapılabilir:

- *ObjectRecOffset* görelı konum alanını kullanarak.

Bu durumda, uygulama MQOD içeren kendi yapısını ve ardından MQOR kayıtları dizisini (gerektiği kadar dizi ögesi olan) bildirmeli ve *ObjectRecOffset* ögesini MQOD ' un başlangıcından itibaren dizideki ilk ögenin görelı konumuna ayarlamalıdır. Bu görelı konumun doğru olduğundan ve bir MQLONG içinde tutulabilecek bir değere sahip olduğundan emin olun (en kısıtlayıcı programlama dili COBOL 'dur; bu dilin geçerli aralığı -999 999 999 999-999 999 999 arasındadır).

İşaretçi veri tipini desteklemeyen ya da gösterge veri tipini farklı ortamlara (örneğin, COBOL programlama dili) taşınmayacak şekilde uygulayan programlama dilleri için *ObjectRecOffset* kullanın.

- *ObjectRecPtr* gösterge alanını kullanarak.

Bu durumda, uygulama MQOR yapılarının dizisini MQOD yapısından ayrı olarak bildirebilir ve *ObjectRecPtr* değerini dizinin adresine ayarlayabilir.

İşaretçi veri tipini farklı ortamlara (örneğin, C programlama dili) taşınabilir bir şekilde destekleyen programlama dilleri için *ObjectRecPtr* kullanın.

Hangi tekniği seçerseniz seçin, *ObjectRecOffset* ve *ObjectRecPtr* yöntemlerinden birini kullanın; Her ikisi de sıfırda ya da her ikisi de sıfır değilse, çağrı MQRC_OBJECT_RECORDS_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQOD_VERSION_2' den küçükse bu alan yoksayılr.

ResponseRecMQOD için görelı konum (MQLONG)

Bu, MQOD yapısının başlangıcından itibaren ilk MQRR yanıt kaydının bayt cinsinden görelı konumudur. Görelı konum pozitif ya da negatif olabilir. *ResponseRecOffset* yalnızca bir dağıtım listesi açıldığında kullanılır. *RecsPresent* sıfırda alan yoksayılr.

Bir dağıtım listesi açıldığında, açılmayan kuyrukları (MQRR 'deki *CompCode* alanı) ve her başarısızlığın nedenini (MQRR 'deki *Reason* alanı) saptamak için bir ya da daha çok MQRR yanıt kaydı dizisi sağlayabilirsiniz. Veriler, yanıt kayıtları dizisinde, nesne kayıtları dizisindeki kuyruk adlarıyla aynı sırayla döndürülür. Kuyruk yöneticisi yanıt kayıtlarını yalnızca çağrının sonucu karışık olduğunda (diğer bir deyişle, bazı kuyruklar başarısız olduğunda ya da tümü farklı nedenlerle başarısız olduğunda) ayarlar; çağrıdaki MQRC_MULTIPLE_REASON neden kodu bu durumu gösterir. Aynı neden kodu tüm kuyruklar için de geçerliyse, MQOPEN ya da MQPUT1 çağrısının **Reason** değiştirgesinde bu neden döndürülür ve yanıt kayıtları ayarlanmaz. Yanıt kayıtları isteğe bağlıdır, ancak bunlar sağlanırsa *RecsPresent* olmalıdır.

Yanıt kayıtları, *ResponseRecOffset* içinde bir görelı konum belirtilerek ya da *ResponseRecPtr* içinde bir adres belirtilerek nesne kayıtlarıyla aynı şekilde sağlanabilir; Bunun nasıl yapılacağını öğrenmek için bkz. "ObjectRecMQOD için görelı konum (MQLONG)" sayfa 486. Ancak, birden fazla *ResponseRecOffset* ve *ResponseRecPtr* kullanılamaz; her ikisi de sıfır değilse, çağrı MQRC_RESPONSE_RECORDS_ERROR neden koduyla başarısız olur.

MQPUT1 çağrısına ilişkin bu yanıt kayıtları, ileti dağıtım listesindeki kuyruklara gönderildiğinde ortaya çıkan hatalarla ilgili bilgileri ve kuyruklar açıldığında oluşan hataları döndürmek için kullanılır. Bir kuyruğa ilişkin koyma işleminden gelen tamamlanma kodu ve neden kodu, yalnızca kuyruktaki tamamlanma kodu MQCC_OK ya da MQCC_WARNING ise, o kuyruğa ilişkin açma işleminden gelenlerin yerine geçer.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0'dır. *Version* , MQOD_VERSION_2' den küçükse bu alan yoksayılr.

ObjectRecMQOD için Ptr (MQPTR)

Bu, ilk MQOR nesne kaydının adresidir. *ObjectRecPtr* yalnızca bir dağıtım listesi açıldığında kullanılır. *RecsPresent* sıfırda alan yoksayılr.

Nesne kayıtlarını belirtmek için *ObjectRecPtr* ya da *ObjectRecOffset* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız; *ObjectRecOffset* alanının açıklaması için bkz. "[ObjectRecMQOD için görelî konum \(MQLONG\)](#)" sayfa 486. *ObjectRecPtr* kullanmıyorsanız, boş değeri gösterge ya da boş değeri bayt olarak ayarlayın.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değeri gösterge ve tersi durumda tümü boş değeri bir bayt dizgisidir. *Version* , MQOD_VERSION_2' den küçükse bu alan yoksayılr.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir ve ilk değeri tümüyle boş değeri byte dizgisi olur.

ResponseRecMQOD için Ptr (MQPTR)

Bu, ilk MQRR yanıt kaydının adresidir. *ResponseRecPtr* yalnızca bir dağıtım listesi açıldığında kullanılır. *RecsPresent* sıfırda alan yoksayılr.

Yanıt kayıtlarını belirtmek için *ResponseRecPtr* ya da *ResponseRecOffset* kullanın, ancak her ikisini de belirtmeyin; ayrıntılar için bkz. "[ResponseRecMQOD için görelî konum \(MQLONG\)](#)" sayfa 486. *ResponseRecPtr* kullanmıyorsanız, boş değeri gösterge ya da boş değeri bayt olarak ayarlayın.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değeri gösterge ve tersi durumda tümü boş değeri bir bayt dizgisidir. *Version* , MQOD_VERSION_2' den küçükse bu alan yoksayılr.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir ve ilk değeri tümüyle boş değeri byte dizgisi olur.

MQOD için AlternateSecurityTanıtıcısı (MQBYTE40)

Bu, uygun yetki denetimlerinin gerçekleştirilmesine izin vermek için *AlternateUserId* ile yetkilendirme hizmetine iletilen bir güvenlik tanıtıcısıdır. *AlternateSecurityId* yalnızca aşağıdaki durumda kullanılır:

- MQOPEN çağrısında MQOO_ALTERNATE_USER_AUTHORITY belirtildi ya da
- MQPUT1 çağrısında MQPMO_ALTERNATE_USER_AUTHORITY belirtildi,

ve *AlternateUserId* alanı, alanın ilk boş karakterine ya da sonuna kadar tamamen boş değildir.

Windows işletim sistemlerinde *AlternateSecurityId* , *AlternateUserId* ögesini benzersiz olarak tanımlayan Windows güvenlik tanıtıcısını (SID) sağlamak için kullanılabilir. Bir kullanıcının SID 'si, LookupAccountName () Windows API çağrısı kullanılarak Windows sisteminden alınabilir.

z/OS' da bu alan yoksayılr.

AlternateSecurityId alanı aşağıdaki yapıya sahiptir:

- İlk bayt, izleyen önemli verilerin uzunluğunu içeren ikili bir tamsayıdır; değeri, uzunluk byte 'ının kendisini dışlar. Güvenlik tanıtıcısı yoksa, uzunluk sıfır olur.
- İkinci bayt, var olan güvenlik tanıtıcısının tipini gösterir; aşağıdaki değerler kullanılabilir:

MQSIDT_NT_SECURITY_ID

Windows güvenlik tanıtıcısı.

MQSIDT_NONE

Güvenlik tanıtıcısı yok.

- Birinci baytın tanımladığı uzunluğa kadar olan üçüncü ve sonraki baytlar güvenlik tanıtıcısının kendisini içerir.
- Alandaki kalan bayt sayısı ikili sıfır olarak ayarlanır.

Aşağıdaki özel değeri kullanabilirsiniz:

MQSID_YOK

Güvenlik tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQSID_NONE_ARRAY değişmezi de tanımlanır; bu değer MQSID_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu bir giriş alanıdır. Bu alanın uzunluğu MQ_SECURITY_ID_LENGTH tarafından verilir. Bu alanın ilk değeri MQSID_NONE. *Version* , MQOD_VERSION_3' den küçükse bu alan yoksayılr.

MQOD için ResolvedQName (MQCHAR48)

Bu, yerel kuyruk yöneticisi adı çözdükten sonra hedef kuyruğun adıdır. Döndürülen ad, *ResolvedQMGrName* ile tanımlanan kuyruk yöneticisinde bulunan bir kuyruğun adıdır.

Boş olmayan bir değer, nesne göz atma, giriş ya da çıkış (ya da herhangi bir birleşim) için açılmış tek bir kuyruksa döndürülür. Açılan nesne aşağıdakilerden biriye, *ResolvedQName* boş olarak ayarlanır:

- Kuyruk değil
- Bir kuyruk, ancak göz atma, giriş ya da çıkış için açılmadı
- Dağıtım listesi
- Bir konu nesnesine başvuran bir diğer ad kuyruğu (bunun yerine [ResObjectString](#) dizgisine bakın).
- Bir konu nesnesine çözülen diğer ad kuyruğu.

Bu bir çıkış alanıdır. Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir. *Version* , MQOD_VERSION_3' den küçükse bu alan yoksayılr.

MQOD için ResolvedQMGrAdı (MQCHAR48)

Bu, yerel kuyruk yöneticisi adı çözdükten sonra hedef kuyruk yöneticisinin adıdır. Döndürülen ad, *ResolvedQName* ile tanımlanan kuyruğun iyisi olan kuyruk yöneticisinin adıdır. *ResolvedQMGrName* , yerel kuyruk yöneticisinin adı olabilir.

ResolvedQName , yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun sahip olduğu paylaşılan bir kuyruksa, *ResolvedQMGrName* , kuyruk paylaşım grubunun adıdır. Kuyruk başka bir kuyruk paylaşım grubuna aitse, *ResolvedQName* kuyruk paylaşım grubunun adı ya da kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisinin adı olabilir (döndürülen değer türü, yerel kuyruk yöneticisinde var olan kuyruk tanımlamalarına göre belirlenir).

Boş olmayan bir değer, nesne göz atma, giriş ya da çıkış (ya da herhangi bir birleşim) için açılmış tek bir kuyruksa döndürülür. Açılan nesne aşağıdakilerden biriye, *ResolvedQMGrName* boş olarak ayarlanır:

- Kuyruk değil
- Bir kuyruk, ancak göz atma, giriş ya da çıkış için açılmadı
- MQOO_BIND_NOT_FIXED belirtilmiş bir küme kuyruğu (ya da **DefBind** kuyruk özneliği MQBND_BIND_NOT_FIXED değerine sahip olduğunda MQOO_BIND_AS_Q_DEF ile birlikte)
- Dağıtım listesi

Bu bir çıkış alanıdır. Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir. *Version* , MQOD_VERSION_3' den küçükse bu alan yoksayılr.

MQOD için ObjectString (MQCHARV)

ObjectString alanı, uzun nesne adını belirtir.

Bu, kullanılacak uzun nesne adını belirtir. Bu alana yalnızca belirli *ObjectType* değerleri için başvurulur ve diğer tüm değerler için yoksayıdır. Bu alanın kullanıldığını gösteren değerlerin ayrıntıları için *ObjectType* açıklamasına bakın.

ObjectString yanlış belirtilirse, MQCHARV yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı MQRC_OBJECT_STRING_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu yapıdaki alanların başlangıç değerleri, MQCHARV yapısındaki değerlerle aynıdır.

Tam konu adı iki farklı alandan oluşturulabilir: *ObjectName* ve *ObjectString*. Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.

MQOD için SelectionString (MQCHARV)

Bu, kuyruktan ileti alınırken kullanılan seçim ölçütlerini sağlamak için kullanılan dizedir.

SelectionString aşağıdaki durumlarda sağlanmamalıdır:

- *ObjectType* MQOT_Q değilse
- Açılmakta olan kuyruk MQOO_BROWSE ya da MQOO_INPUT_* seçeneklerinden biri kullanılarak açılmıyorsa

Bu durumlarda *SelectionString* belirtiliyorsa, çağrı MQRC_SELECTOR_INVALID_FOR_TYPE neden koduyla başarısız olur.

SelectionString yanlış belirtilirse, “MQCHARV-Değişken Uzunluklu Dizgi” sayfa 294 yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı MQRC_SELECTION_STRING_ERROR neden koduyla başarısız olur. *SelectionString* uzunluk üst sınırı: MQ_SELECTOR_LENGTH.

SelectionString kullanımı, [Seçicileri](#) içinde açıklanmıştır.

MQOD için ResObjectDizesi (MQCHARV)

ResObjectString alanı, kuyruk yöneticisi *ObjectName* alanında sağlanan adı çözdükten sonra uzun nesne adıdır.

Bu alan yalnızca, bir konu nesnesine gönderme yapan konular ve kuyruk diğer adları için döndürülür.

Uzun nesne adı *ObjectString* içinde sağlanıyorsa ve *ObjectName* içinde hiçbir şey sağlanmıyorsa, bu alanda döndürülen değer *ObjectString* içinde sağlananla aynıdır.

Bu alan atlanırsa (ResObjectString.VSBufSize sıfırdır), *ResObjectString* döndürülmez, ancak uzunluk ResObjectString.VSLength içinde döndürülür.

Arabellek uzunluğu (ResObjectString.VSBufSize içinde sağlanır) tam *ResObjectString* değerinden kısaysa, dizgi kesilir ve sağlanan arabelleğe sığabilecek en sağdaki karakterlerin sayısı döndürülür.

ResObjectString yanlış belirtilirse, MQCHARV yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı MQRC_RES_OBJECT_STRING_ERROR neden koduyla başarısız olur.

MQOD için ResolvedType (MQLONG)

Açılmakta olan çözümlenen (temel) nesnenin tipi.

Olası değerler şunlardır:

MQOT_Q

Çözümlenen nesne bir kuyruk. Bu değer, bir kuyruk doğrudan açıldığında ya da bir kuyruğu gösteren diğer ad kuyruğu açıldığında geçerlidir.

MQOT_KONU

Çözümlenen nesne bir konudur. Bu değer, bir konu doğrudan açıldığında ya da konu nesnesini gösteren bir diğer ad kuyruğu açıldığında geçerlidir.

MQOT_NONE

Çözülen tip ne bir kuyruk ne de bir konu.

MQOR-Nesne kaydı

Tek bir hedef kuyruğun kuyruk adını ve kuyruk yöneticisi adını belirtmek için MQOR yapısını kullanın. MQOR, MQOPEN ve MQPUT1 çağrılarına ilişkin bir giriş yapısıdır.

Kullanılabilirlik

MQOR yapısı aşağıdaki altyapılarda bulunur:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

Karakter kümesi ve kodlama

MQOR içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında olmalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Kullanım

MQOPEN çağrısında bu yapıların bir dizisini sağlayarak bir kuyruk listesi açabilirsiniz; bu listeye dağıtım listesi adı verilir. MQOPEN çağrısının döndürdüğü kuyruk tanıtıcısı kullanılarak konan her ileti, kuyruğun başarıyla açılması koşuluyla, listedeki kuyrukların her birine yerleştirilir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 505. MQOR için MQOR içindeki alanlar		
Alan adı ve açıklaması	Değişmez adı	Değişmez ilk değeri (varsa)
<u>ObjectName</u> (nesne adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>ObjectQMgrAdı</u> (nesne kuyruğu yöneticisi adı)	Yok	Boş değerli dizilim ya da boşluklar
Notlar: <ol style="list-style-type: none">1. Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.2. C programlama dilinde, makro değişkeniMQOR_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir: <pre>MQOR MyOR = {MQOR_DEFAULT};</pre>		

Dil bildirimleri

MQOR için C bildirimini

```
typedef struct tagMQOR MQOR;
struct tagMQOR {
    MQCHAR48  ObjectName;      /* Object name */
    MQCHAR48  ObjectQMgrName; /* Object queue manager name */
};
```

MQOR için COBOL bildirimini

```
** MQOR structure
10 MQOR.
** Object name
15 MQOR-OBJECTNAME PIC X(48).
** Object queue manager name
15 MQOR-OBJECTQMGRNAME PIC X(48).
```

MQOR için PL/I bildirimini

```
dcl
1 MQOR based,
3 ObjectName char(48), /* Object name */
3 ObjectQMgrName char(48); /* Object queue manager name */
```

MQOR için Visual Basic bildirimini

```
Type MQOR
    ObjectName As String*48 'Object name'
    ObjectQMgrName As String*48 'Object queue manager name'
End Type
```

MQOR için ObjectName (MQCHAR48)

Bu, aşağıdaki durumlar dışında, MQOD yapısındaki *ObjectName* alanıyla aynıdır (ayrıntılar için MQOD ' ye bakın):

- Bir kuyruğun adı olmalıdır.
- Bir model kuyruğunun adı olmamalıdır.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

ObjectQMgrMQOR için Ad (MQCHAR48)

Bu, MQOD yapısındaki *ObjectQMgrName* alanıyla aynıdır (ayrıntılar için MQOD ' ye bakın).



Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQPD-Özellik tanımlayıcısı

MQPD yapısı, bir özelliğin özniteliklerini tanımlamak için kullanılır. Yapı, MQSETMP çağrısında bir giriş/çıkış değiştirgesi ve MQINQMP çağrısında bir çıkış değiştirgesidir.

Kullanılabilirlik

MQPD yapısı aşağıdaki platformlarda bulunur:

-  AIX
-  IBM i

-  Linux
-  Windows
-  z/OS

ve IBM MQ MQI clients için.

Karakter kümesi ve kodlama

MQPD içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında olmalıdır (**MQENC_NATIVE**).

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 506. MQPD içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQPD_STRUC_ID	' PD '
<u>Sürüm</u> (yapı sürümü numarası)	MQPD_VERSION_1	1
<u>Seçenekler</u> (seçenekler)	MQPD_NONE	0
<u>Destek</u> (ileti özelliği için gerekli destek)	MQPD_SUPPORT_OPTIONAL	0
<u>Bağlam</u> (özelliğin ait olduğu ileti bağlamı)	MQPD_NO_CONTEXT	0
<u>CopyOptions</u> (özelliğin ait olduğu kopyalama seçenekleri)	MQCOPY_DEFAULT	0

Notlar:

1. C programlama dilinde, MQPD_DEFAULT makro değişkeni çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQPD MyPD = {MQPD_DEFAULT};
```

Dil bildirimleri

MQPD için C bildirimi

```
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 */
};
```

MQPD için COBOL bildirimi

```
** 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.

```

MQPD için PL/I bildirimi

```

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 */

```

MQPD için High Level Assembler (Üst Düzey Çevirici) bildirimi

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)	

MQPD için StrucId (MQCHAR4)

Bu, özellik tanımlayıcı yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQPD_STRUC_ID.

Değer şöyle olmalıdır:

MQPD_STRUC_ID

Özellik tanımlayıcı yapısına ilişkin tanıttıcı.

C programlama dili için, **MQPD_STRUC_ID_ARRAY** sabiti de tanımlanır. Bu, **MQPD_STRUC_ID** ile aynı değere sahiptir, ancak bir dizgi yerine bir karakter dizisidir.

MQPD için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQPD_VERSION_1

Version-1 özellik tanımlayıcı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQPD_CURRENT_VERSION

Özellik tanımlayıcı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **MQPD_VERSION_1**.

MQPD için Seçenekler (MQLONG)

Değer şöyle olmalıdır:

MQPD_NONE

Seçenek belirtilmedi

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQPD_NONE.

MQPD için destek (MQLONG)

Bu alanda, bu özelliği içeren iletinin kuyruğa konması için kuyruk yöneticisinin ileti özelliği için hangi destek düzeyinin gerekli olduğu açıklanır. Bu yalnızca IBM MQtanımlı özellikler için geçerlidir; diğer tüm özellikler için destek isteğe bağlıdır.

IBM MQtanımlı özellik kuyruk yöneticisi tarafından bilindiğinde alan otomatik olarak doğru değere ayarlanır. Özellik tanınmazsa, MQPD_SUPPORT_OPTIONAL atanır. Bir kuyruk yöneticisi, kuyruk yöneticisinin yanlış olarak algıladığı IBM MQtanımlı bir özelliği içeren bir ileti aldığı anda, kuyruk yöneticisi *Support* alanının değerini düzeltir.

MQCMHO_NO_VALIDATION seçeneğinin ayarlandığı bir ileti tanıtıcısında MQSETMP çağrısı kullanılarak IBM MQtanımlı bir özellik ayarlanırken *Support* bir giriş alanı olur. Bu, bir uygulamanın, özelliğin bağlı kuyruk yöneticisi tarafından desteklenmediği, ancak iletinin başka bir kuyruk yöneticisinde işlemek üzere tasarlandığı, doğru değere sahip bir IBM MQtanımlı özellik koymasına olanak sağlar.

MQPD_SUPPORT_OPTIONAL değeri her zaman IBM MQtanımlı olmayan özelliklere atanır.

İleti özelliklerini destekleyen bir IBM WebSphere MQ 7.0 kuyruk yöneticisi, tanınmayan bir *Support* değeri içeren bir özellik alırsa, özellik aşağıdaki gibi işlenir:

- Tanınmayan değerlerden herhangi biri MQPD_REJECT_UNSUP_MASK içinde bulunuyorsa, MQPD_SUPPORT_REQUIRED belirtilir.
- Tanınmayan değerlerden herhangi biri MQPD_ACCEPT_UNSUP_IF_XMIT_MASK içinde bulunuyorsa, MQPD_SUPPORT_REQUIRED_IF_LOCAL belirtilir.
- Ters durumda MQPD_SUPPORT_OPTIONAL belirtilir.

MQINQMP çağrısı aşağıdaki değerlerden birini döndürdü ya da MQCMHO_NO_VALIDATION seçeneğinin ayarlandığı bir ileti tanıtıcısında MQSETMP çağrısı kullanılırken değerlerden biri belirtilebilir:

MQPD_SUPPORT_OPTIONAL

Özellik desteklenmese de bir kuyruk yöneticisi tarafından kabul edilir. İletinin ileti özelliklerini desteklemeyen bir kuyruk yöneticisine akması için özellik atılabilir. Bu değer, IBM MQtanımlı olmayan özelliklere de atanır.

MQPD_SUPPORT_REQUIRED

Özellik için destek gereklidir. İleti, IBM MQtanımlı özelliği desteklemeyen bir kuyruk yöneticisi tarafından reddedildi. MQPUT ya da MQPUT1 çağrısı, tamamlama kodu MQCC_FAILED ve neden kodu MQRC_UNSUPPORTED_PROPERTY ile başarısız oluyor.

MQPD_SUPPORT_REQUIRED_IF_LOCAL

İleti yerel bir kuyruğa gönderildiyse, ileti IBM MQtanımlı özelliği desteklemeyen bir kuyruk yöneticisi tarafından reddedilir. MQPUT ya da MQPUT1 çağrısı, tamamlama kodu MQCC_FAILED ve neden kodu MQRC_UNSUPPORTED_PROPERTY ile başarısız oluyor.

İleti uzak bir kuyruk yöneticisine gönderildiyse, MQPUT ya da MQPUT1 çağrısı başarılı olur.

İleti tanıtıcısı MQCMHO_NO_VALIDATION seçeneği ayarlanarak yaratıldıysa, MQINQMP çağrısında bir çıkış alanı ve MQSETMP çağrısında bir giriş alanıdır. Bu alanın ilk değeri MQPD_SUPPORT_OPTIONAL.

MQPD için Bağlam (MQLONG)

Bu, özelliğin hangi ileti bağlamına ait olduğunu açıklar.

Bir kuyruk yöneticisi, kuyruk yöneticisinin yanlış olarak algıladığı IBM MQtanımlı bir özelliği içeren bir ileti aldığı anda, kuyruk yöneticisi *Context* alanının değerini düzeltir.

Aşağıdaki seçenek belirtilebilir:

MQPD_USER_CONTEXT

Özellik, kullanıcı bağlamıyla ilişkilendirildi.

MQSETMP çağrısını kullanarak kullanıcı bağlamıyla ilişkili bir özellik ayarlayabilmek için özel bir yetkilendirme gerekmez.

Bir IBM WebSphere MQ 7.0 kuyruk yöneticisinde, kullanıcı bağlamıyla ilişkili bir özellik MQOO_SAVE_ALL_CONTEXT için açıklandığı gibi kaydedilir. MQPMO_PASS_ALL_CONTEXT belirtilmiş bir MQPUT çağrısı, özelliğin kaydedilen bağlamdan yeni iletiye kopyalanmasına neden olur.

Daha önce açıklanan seçenek gerekli değilse, aşağıdaki seçenek kullanılabilir:

MQPD_NO_CONTEXT

Özellik bir ileti bağlamıyla ilişkilendirilmemiş.

MQRC_PD_ERROR *Reason* koduyla tanınmayan bir değer reddedildi.

MQSETMP çağrısı için bir giriş/çıkış alanı ve MQINQMP çağrısından bir çıkış alanıdır. Bu alanın ilk değeri MQPD_NO_CONTEXT 'dir.

MQPD için CopyOptions (MQLONG)

Bu, özelliğin hangi tipte iletilere kopyalanması gerektiğini açıklar. Bu yalnızca tanınan IBM MQ tanımlı özellikler için bir çıkış alanıdır; IBM MQ uygun değeri ayarlar.

Bir kuyruk yöneticisi, kuyruk yöneticisinin yanlış olarak algıladığı IBM MQ tanımlı bir özelliği içeren bir ileti aldığı anda, kuyruk yöneticisi *CopyOptions* alanının değerini düzeltir.

Bu seçeneklerden birini ya da birkaçını belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyorsa).

MQCOPY_FORWARD

Bu özellik, iletilmekte olan bir iletiye kopyalanır.

MQCOPY_PUBLISH

Bu özellik, bir ileti yayınlanırken abone tarafından alınan iletiye kopyalanır.

MQCOPY_REPLY

Bu özellik bir yanıt iletisine kopyalanır.

MQCOPY_REPORT

Bu özellik bir rapor iletisine kopyalanır.

MQCOPY_ALL

Bu özellik, sonraki iletilerin tüm tiplerine kopyalanır.

Varsayılan seçenek: Varsayılan kopyalama seçenekleri kümesini sağlamak için aşağıdaki seçenek belirtilebilir:

MQCOPY_DEFAULT

Bu özellik, iletilen bir iletiye, bir rapor iletisine ya da bir ileti yayınlanırken abone tarafından alınan bir iletiye kopyalanır.

Bu, MQCOPY_FORWARD, artı MQCOPY_REPORT ve MQCOPY_PUBLISH seçeneklerinin birleşimini belirtmeye eşdeğerdir.

Daha önce açıklanan seçeneklerden hiçbiri gerekmiyorsa, aşağıdaki seçeneği kullanın:

MQCOPY_NONE

Başka bir kopyalama seçeneği belirtilmediğini belirtmek için bu değeri kullanın; programsal olarak bu özellik ile sonraki iletiler arasında ilişki yoktur. Bu, ileti tanımlayıcı özellikleri için her zaman döndürülür.

MQSETMP çağrısı için bir giriş/çıkış alanı ve MQINQMP çağrısından bir çıkış alanıdır. Bu alanın ilk değeri MQCOPY_DEFAULT.

MQPMO-İleti koyma seçenekleri

MQPMO yapısı, uygulamanın iletilerin kuyruklara nasıl yerleştirileceğini ya da konulara nasıl yayınlanacağını denetleyen seçenekleri belirtmesini sağlar. Yapı, MQPUT ve MQPUT1 çağrılarında bir giriş/çıkış değiştirgesidir.

Sürüm

MQPMO ' nun yürürlükteki sürümü: MQPMO_VERSION_3. Belirli alanlar yalnızca MQPMO ' nun belirli sürümlerinde kullanılabilir. Uygulamaları birden çok ortam arasında bağlamanız gerekirse, MQPMO sürümünün tüm ortamlarda tutarlı olduğundan emin olmanız gerekir. Yapının yalnızca belirli sürümlerinde var olan alanlar, bu konuda ve alan açıklamalarında bu şekilde tanımlanır.

Desteklenen programlama dilleri için sağlanan üstbilgi, COPY ve INCLUDE dosyaları, ortam tarafından desteklenen en son MQPMO sürümünü içerir, ancak *Version* alanının ilk değeri MQPMO_VERSION_1 olarak ayarlanmıştır. *version-1* yapısında var olmayan alanları kullanmak için uygulamanın *Version* alanını gerekli sürümün sürüm numarasına ayarlaması gerekir.

Karakter kümesi ve kodlama

MQPMO içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında yer almalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 507. MQPMO içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQPMO_STRUC_ID	'PMO→'
Sürüm (yapı sürümü numarası)	MQPMO_VERSION_1	1
Seçenekler (MQPUT ve MQPUT1eylemine denetleyen seçenekler)	MQPMO_NONE	0
Zamanaşımı (ayrılmış)	Yok	-1
Bağlam (giriş kuyruğunun nesne tanıtıcısı)	Yok	0
KnownDestSayı (yerel kuyruklara başarıyla gönderilen iletilerin sayısı)	Yok	0
UnknownDestCount (Uzak kuyruklara başarıyla gönderilen ileti sayısı)	Yok	0
InvalidDestSayı (gönderilemeyen iletilerin sayısı)	Yok	0
ResolvedQName (hedef kuyruğun çözülmüş adı)	Yok	Boş değerli dizilim ya da boşluklar
ResolvedQMgrAdı (hedef kuyruk yöneticisinin çözümlenen adı)	Yok	Boş değerli dizilim ya da boşluklar
Not: <i>Version</i> , MQPMO_VERSION_2' den küçükse kalan alanlar yoksayılr.		
RecsPresent (mevcut koyma iletileri kaydı ya da yanıt kaydı sayısı)	Yok	0

Çizelge 507. MQPMO içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
PutMsgRecFields (hangi MQPMR alanlarının var olduğunu gösteren işaretler)	MQPMRF_NONE	0
PutMsgRecOffset (MQPMO ' nun başlangıcından ilk koyma iletisi kaydının görelî konumu)	Yok	0
ResponseRecKayma (MQPMO ' nun başlangıcından ilk yanıt kaydının görelî konumu)	Yok	0
PutMsgRecPtr (ilk koyma iletisi kaydının adresi)	Yok	Boş değeri gösterge ya da boş değeri byte
ResponseRecPtr (ilk yanıt kaydının adresi)	Yok	Boş değeri gösterge ya da boş değeri byte
Not: <i>Version</i> , MQPMO_VERSION_3' den küçükse kalan alanlar yoksayılır.		
OriginalMsgTanıtıcısı (özgün ileti tanıtıcısı)	MQHM_NONE	0
NewMsgHandle (yeni ileti tanıtıcısı)	MQHM_NONE	0
İşlem (gerçekleştirilmekte olan koyma değeri tipi ve <i>OriginalMsgHandle</i> alanı tarafından belirtilen özgün ileti ile <i>NewMsgHandle</i> alanı tarafından belirtilen yeni ileti arasındaki ilişki)	MQACTP_YENI	0
PubLevel (yayın tarafından hedeflenen abonelik düzeyi)	Yok	9
Notlar: <ol style="list-style-type: none"> 1. – simgesi, tek bir boş karakteri temsil eder. 2. Boş değeri dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğeri programlama dillerindeki boş karakterleri gösterir. 3. C programlama dilinde, makro değeri MQPMO_DEFAULT, çizelgede listelenen değeri içerir. Yapıdaki alanlar için başlangıç değeri sağlamak üzere bunu aşağıdaki şekilde kullanın: <pre>MQPMO MyPMO = {MQPMO_DEFAULT};</pre>		

Dil bildirimleri

MQPMO için C bildirimi

```
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 */
};

```

MQPMO için COBOL bildirimi

```

** 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-INVALIDDSTCOUNT 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-NEWMSGHANDLE PIC S9(18) BINARY.
** The action being performed
15 MQPMO-ACTION PIC S9(9) BINARY.
** Publish level
15 MQPMO-PUBLEVEL PIC S9(9) BINARY.

```

MQPMO için PL/I bildirimi

```

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 */

```

MQPMO için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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_UNKNOWNDSTCOUNT 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_PUTMSGRECPtr    DS    F      Address of first put message
*                    *          record
MQPMO_RESPONSERECPtr  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
                    ORG    MQPMO
MQPMO_AREA           DS    CL(MQPMO_LENGTH)

```

MQPMO için Visual Basic bildirimi

```

Type MQPMO
  StrucId      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'

```

InvalidDestCount	As Long	'to remote queues' '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 MQPMP 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		

MQPMO için StrucId (MQCHAR4)

Bu, koyma iletisi seçenekleri yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQPMO_STRUC_ID.

Değer şöyle olmalıdır:

MQPMO_STRUC_ID

Koyma iletisi seçenekleri yapısına ilişkin tanıttıcı.

C programlama dili için, MQPMO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQPMO_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQPMO için Sürüm (MQLONG)

Yapı sürümü numarası.

Değer aşağıdakilerden biri olmalıdır:

MQPMO_VERSION_1

Version-1 koyma iletisi seçenekleri yapısı.

Bu sürüm tüm ortamlarda desteklenir.

MQPMO_VERSION_2

Version-2 koyma iletisi seçenekleri yapısı.

Bu sürüm aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

MQPMO_VERSION_3

Version-3 koyma iletisi seçenekleri yapısı.

Bu sürüm tüm ortamlarda desteklenir.

Yalnızca yapının daha yeni sürümünde var olan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQPMO_CURRENT_VERSION

Koyma iletisi seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQPMO_VERSION_1' dir.

MQPMO için Seçenekler (MQLONG)

Seçenekler alanı, **MQPUT** ve **MQPUT1** çağrılarının çalışmasını denetler.

Kapsam seçeneği. MQPMO seçeneklerinden herhangi birini belirtebilir ya da hiçbirini belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemi kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyse). Geçerli olmayan birleşimler belirtilir; diğer birleşimler geçerlidir.

Aşağıdaki seçenek, gönderilen yayınların kapsamını denetler:

MQPMO_SCOPE_QMGR

Yayın yalnızca bu kuyruk yöneticisine abone olan abonelere gönderilir. Yayın, bu kuyruk yöneticisine abonelik yapan herhangi bir uzak yayınlama/abone olma kuyruk yöneticisine iletilmez; bu, PUBSCOPE konu özniteliği kullanılarak ayarlanan tüm davranışları geçersiz kılar.

Not: Ayarlanmazsa, yayınlama kapsamı PUBSCOPE konu özniteliği tarafından belirlenir.

Yayınlama seçenekleri. Aşağıdaki seçenekler, iletilerin bir konuya yayınlama şeklini denetler:

MQPMO_SUPPRESS_REPLYTO

Bu yayının MQMD 'nin *ReplyToQ* ve *ReplyToQMGR* alanlarında belirtilen bilgiler abonelere iletilmez. Bu seçenek *ReplyToQ* gerektiren bir rapor seçeneğiyle kullanılırsa, çağrı MQRC_MISSING_REPLY_TO_Q ile başarısız olur.

MQPMO_RETAIN

Gönderilmekte olan yayın, kuyruk yöneticisi tarafından alıkonacak. Bu alıkoyma, bir abonenin MQSUBRQ çağrısıyla yayımlandıktan sonra bu yayının bir kopyasını istemesini sağlar. Ayrıca, bir yayının, bu yayın yapıldıktan sonra aboneliğini yapan uygulamalara gönderilmesini sağlar (MQSO_NEW_PUBLICATIONS_ONLY seçeneği kullanılarak gönderilmemeyi seçmedikleri sürece). Bir uygulamaya alıkonan bir yayın gönderilirse, bu, o yayının MQIsRetained ileti özelliğiyle gösterilir.

Konu ağacının her düğümünde yalnızca bir yayın korunabilir. Bu nedenle, bu konu için başka bir uygulama tarafından yayınlanan alıkonmuş bir yayın varsa, bu yayın ile değiştirilir. Bu nedenle, aynı konuda birden fazla yayıncının ileti tutmasını önlemek daha iyidir.

Alıkonan yayınlar bir abone tarafından istendiğinde, kullanılan abonelik konuda genel arama karakteri içerebilir; bu durumda, alıkonan yayınların sayısı (konu ağacındaki çeşitli düğümlerde) eşleşebilir ve istekte bulunan uygulamaya birkaç yayın gönderilebilir. Daha fazla ayrıntı için "[MQSUBRQ-Abonelik isteği](#)" sayfa 786 aramasının açıklamasına bakın.

Alıkonan yayınların abonelik düzeyleriyle nasıl etkileşimde bulunacağına ilişkin bilgi için [Intercepting publications](#) başlıklı konuya bakın.

Bu seçenek kullanılırsa ve yayın alıkonamazsa, ileti yayınlanmaz ve çağrı MQRC_PUT_NOT_ALLOWED ile başarısız olur.

MQPMO_NOT_OWN_SUBS

Kuyruk yöneticisine, uygulamanın yayınlarından hiçbirini sahip olduğu aboneliklere göndermek istemediğini bildirir. Bağlantı tanıtıcıları aynıysa, aboneliklerin aynı uygulamaya ait olduğu varsayılır.

MQPMO_WARN_IF_NO_SUBS_EŞLEŞME

Yayınla herhangi bir abonelik eşleşmezse, MQCC_WARNING ve MQRC_NO_SUBS_MATCHES neden kodunun tamamlanma kodunu (*CompCode*) döndürün.

Koyma işlemi tarafından MQRC_NO_SUBS_EŞLEŞME döndürülürse, yayın herhangi bir aboneliğe teslim edilmedi. Ancak, koyma işleminde MQPMO_RETAIN seçeneği belirtilirse, ileti alıkonur ve daha sonra tanımlanan herhangi bir eşleşen aboneliğe teslim edilir.

Aşağıdaki koşullardan herhangi biri karşılandıysa, konuyla ilgili bir abonelik yayınlama eşleşir:

- İleti abonelik kuyruğuna teslim edilir
- İleti abonelik kuyruğuna teslim edilecekti, ancak kuyrukla ilgili bir sorun, iletinin kuyruğa konamayacağı ve sonuç olarak gitmeyen iletiler kuyruğuna yerleştirileceği ya da atılacağı anlamına gelir.

- İletinin aboneliğe teslimini gizleyen bir yöneltme çıkışı tanımlandı

Aşağıdaki koşullardan herhangi biri karşılandıysa, konuyla ilgili bir abonelik yayınıla eşleşmiyor:

- Abonelikte, yayınıla eşleşmeyen bir seçim dizgisi var
- Abonelik MQSO_PUBLICATION_ON_REQUEST seçeneğini belirtti
- Koyma işleminde MQPMO_NOT_OWN_SUBS seçeneği belirtildiğinden ve abonelik, yayınlayıcının kimliğiyle eşleştiğinden yayını teslim edilmedi.

Eşitleme noktası seçenekleri. Aşağıdaki seçenekler, bir iş birimi içindeki MQPUT ya da MQPUT1 çağrısının katılımıyla ilgilidir:

MQPMO_SYNCPOINT

İstek, normal iş birimi protokolleri içinde çalışmaktadır. İleti, iş birimi kesinleştirilinceye kadar iş biriminin dışında görünmez. İş birimi geriletildiyse, ileti silinir.

MQPMO_SYNCPOINT ve MQPMO_NO_SYNCPOINT belirtilmezse, iş birimi protokollerine koyma isteğinin dahil edilmesi, uygulamayı çalıştıran ortam değil, kuyruk yöneticisini çalıştıran ortam tarafından belirlenir. z/OS üzerinde, koyma isteği bir iş birimi içindedir. Diğer tüm ortamlarda, koyma isteği bir iş birimi içinde değildir.

Bu farklılıklar nedeniyle, bağlantı vermek istediğiniz bir uygulama varsayılan olarak bu seçeneğe izin vermemelidir; MQPMO_SYNCPOINT ya da MQPMO_NO_SYNCPOINT belirtilerek belirtilmelidir.

MQPMO_SYNCPOINT ile MQPMO_NO_SYNCPOINT belirtmeyin.

MQPMO_NO_SYNCPOINT

İstek, normal iş birimi protokollerinin dışında çalışmaktadır. İleti hemen kullanılabilir ve bir iş birimi yedeklenerek silinemez.

MQPMO_NO_SYNCPOINT ve MQPMO_SYNCPOINT belirtilmezse, iş birimi protokollerine koyma isteğinin eklenmesi, uygulamayı çalıştıran ortam değil, kuyruk yöneticisini çalıştıran ortam tarafından belirlenir. z/OS üzerinde, koyma isteği bir iş birimi içindedir. Diğer tüm ortamlarda, koyma isteği bir iş birimi içinde değildir.

Bu farklılıklar nedeniyle, bağlantı vermek istediğiniz bir uygulama varsayılan olarak bu seçeneğe izin vermemelidir; MQPMO_SYNCPOINT ya da MQPMO_NO_SYNCPOINT belirtilerek belirtilmelidir.

MQPMO_SYNCPOINT ile MQPMO_NO_SYNCPOINT belirtmeyin.

İleti tanıtıcısı ve ilinti tanıtıcısı seçenekleri. Aşağıdaki seçenekler, kuyruk yöneticisinden yeni bir ileti tanıtıcısı ya da ilinti tanıtıcısı oluşturmasını ister:

MQPMO_NEW_MSG_ID

Kuyruk yöneticisi, MQMD ' deki *MsgId* alanının içeriğini yeni bir ileti tanıtıcısıyla değiştirir. Bu ileti tanıtıcısı iletiyle birlikte gönderilir ve MQPUT ya da MQPUT1 çağrısının çıkışında uygulamaya döndürülür.

İleti bir dağıtım listesine konurken MQPMO_NEW_MSG_ID seçeneği de belirtilebilir; ayrıntılar için MQPMR yapısındaki *MsgId* alanının tanımına bakın.

Bu seçeneğin kullanılması, her MQPUT ya da MQPUT1 çağrısının öncesinde *MsgId* alanını MQMI_NONE olarak sıfırlama gereksinmesini ortadan kaldırır.

MQPMO_NEW_CORREL_ID

Kuyruk yöneticisi, MQMD ' deki *CorrelId* alanının içeriğini yeni bir ilinti tanıtıcısıyla değiştirir. Bu ilinti tanıtıcısı iletiyle birlikte gönderilir ve MQPUT ya da MQPUT1 çağrısının çıkışında uygulamaya döndürülür.

İleti bir dağıtım listesine konurken MQPMO_NEW_CORREL_ID seçeneği de belirtilebilir; ayrıntılar için MQPMR yapısındaki *CorrelId* alanının tanımına bakın.

MQPMO_NEW_CORREL_ID, uygulamanın benzersiz bir ilinti tanıtıcısı gerektirdiği durumlarda kullanışlıdır.

Grup ve bölüm seçenekleri. Aşağıdaki seçenekler, mantıksal ileti grupları ve kesimleri içindeki iletilerin işlenmesiyle ilgilidir. Seçeneği anlamınıza yardımcı olması için aşağıdaki tanımları okuyun.



Uyarı: Yayınlama/Abone Olma ile bölümlenmiş ya da gruplanmış iletileri kullanamazsınız.

Fiziksel ileti

Bir kuyruğa yerleştirilebilecek ya da kuyruktan kaldırılacak en küçük bilgi birimidir; genellikle tek bir MQPUT, MQPUT1 ya da MQGET çağrısında belirtilen ya da alınan bilgilere karşılık gelir. Her fiziksel iletinin kendi ileti tanımlayıcısı (MQMD) vardır. Genellikle, fiziksel iletiler ileti tanıtcısı (MQMD 'deki *MsgId* alanı) için farklı değerlerle ayırt edilir, ancak bu kuyruk yöneticisi tarafından zorlanmaz.

Mantıksal ileti

Mantıksal ileti, yalnızca z/OS dışı platformlar için tek bir uygulama bilgisi birimidir. Sistem kısıtlamaları olmadığında, mantıksal bir ileti fiziksel bir iletiyle aynıdır. Ancak, mantıksal iletilerin çok büyük olduğu durumlarda, sistem kısıtlamaları mantıksal bir iletinin *segment* adı verilen iki ya da daha çok fiziksel iletiye bölünmesini önerebilir ya da gerekli kılabilir.

Bölümlenmiş bir mantıksal ileti, boş olmayan aynı grup tanıtcısına (MQMD 'deki *GroupId* alanı) ve aynı ileti sıra numarasına (MQMD 'deki *MsgSeqNumber* alanı) sahip iki ya da daha fazla fiziksel iletiden oluşur. Kesimler, mantıksal iletidaki verilerin başlangıcından fiziksel iletidaki verilerin görelî konumunu veren kesim görelî konumu (MQMD 'deki *Offset* alanı) için farklı değerlerle ayırt edilir. Her bir bölüm fiziksel bir ileti olduğu için, mantıksal bir iletidaki bölümler genellikle farklı ileti tanıtcılarına sahiptir.

Bölümlenmemiş, ancak gönderen uygulama tarafından bölümlenmeye izin verilen mantıksal bir ileti, boş olmayan bir grup tanıtcısına da sahiptir; ancak bu durumda, mantıksal ileti bir ileti grubuna ait değilse, bu grup tanıtcısına sahip tek bir fiziksel ileti vardır. Mantıksal ileti bir ileti grubuna ait değilse, gönderen uygulama tarafından segmentasyonun engellendiği mantıksal iletiler boş değerli bir grup tanıtcısına (MQGI_NONE) sahiptir.

İleti grubu

İleti grubu, aynı boş olmayan grup tanıtcısına sahip bir ya da daha fazla mantıksal iletiden oluşan bir kümedir. Gruptaki mantıksal iletiler, 1- n aralığında bir tamsayı olan ileti sıra numarasına ilişkin farklı değerlerle ayırt edilir; burada n , gruptaki mantıksal iletilerin sayısıdır. Mantıksal iletilerden biri ya da daha fazlası bölümlenmişse, grupta n ' den fazla fiziksel ileti vardır.

MQPMO_LOGICAL_ORDER

Bu seçenek, kuyruk yöneticisine uygulamanın iletileri gruplar ve mantıksal ileti kesimleri halinde nasıl yerleştirdiğini bildirir. Yalnızca MQPUT çağrısında belirtilebilir; MQPUT1 çağrısında geçerli değildir.

MQPMO_LOGICAL_ORDER belirtilirse, uygulamanın aşağıdaki öğeler için ardışık MQPUT çağrıları kullandığını gösterir:

1. Her mantıksal iletidaki bölümleri, 0 'dan başlayarak, aralıksız olarak, kesim görelî konumunun artırılmasına göre sırgörüntüleyebilirsiniz?.., ??? ...
2. Bölümleri sonraki mantıksal iletiye yerleştirmeden önce tüm bölümleri tek bir mantıksal iletiye koyun.
3. Mantıksal iletileri, her ileti grubundaki ileti sıra numarasının artma sırasına göre, 1 'den başlayarak, boşluk olmadan koyun. IBM MQ , ileti sıra numarasını otomatik olarak artırır.
4. Mantıksal iletileri sonraki ileti grubuna koymadan önce, tüm mantıksal iletileri bir ileti grubuna koyun.

MQPMO_LOGICAL_ORDER ile ilgili ayrıntılı bilgi için [Mantıksal ve fiziksel sıralama](#) başlıklı konuya bakın.

Bağlam seçenekleri. Aşağıdaki seçenekler ileti bağlamının işlenmesini denetler:

MQPMO_NO_CONTEXT

Hem kimlik hem de kaynak bağlamı, bağlam olmadığını gösterecek şekilde ayarlanır. Bu, MQMD 'deki bağlam alanlarının şu değere ayarlandığı anlamına gelir:

- Karakter alanları için boşluklar
- Bayt alanları için boş değerler

- Sayısal alanlar için sıfırlar

MQPMO_DEFAULT_CONTEXT

İleti, hem kimlik hem de kaynak için, kendisiyle ilişkilendirilmiş varsayılan bağlam bilgisine sahip olmaktadır. Kuyruk yöneticisi, ileti tanımlayıcısındaki bağlam alanlarını aşağıdaki gibi ayarlar:

Çizelge 508. MQMD alanları için varsayılan bağlam bilgisi değerleri

MQMD 'deki Alan	Kullanılan değer
<i>UserIdentifier</i>	Olanaklıysa ortamdan belirlenir; tersi durumda boşluk değerine ayarlanır.
<i>AccountingToken</i>	Olanaklıysa ortamdan saptanır; tersi durumda MQACT_NONE olarak ayarlanır.
<i>AppIdentityData</i>	Boşluk olarak ayarlayın.
<i>PutApplType</i>	Çevreden belirlenmiştir.
<i>PutApplName</i>	Olanaklıysa ortamdan belirlenir; tersi durumda boşluk değerine ayarlanır.
<i>PutDate</i>	İletinin konma tarihine ayarlayın.
<i>PutTime</i>	İletinin konma zamanına ayarlayın.
<i>AppOriginData</i>	Boşluk olarak ayarlayın.

İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

Bağlam seçeneği belirtilmezse, bunlar varsayılan değerler ve işlemlerdir.

MQPMO_PASS_IDENTITY_CONTEXT

İleti, kendisiyle ilişkilendirilmiş bağlam bilgisine sahip olacak. Kimlik bağlamı, *Context* alanında belirtilen kuyruk tanıtıcısından alınır. Kaynak bağlamı bilgileri, kuyruk yöneticisi tarafından, MQPMO_DEFAULT_CONTEXT ile aynı şekilde üretilir (değerler için önceki çizelgeye bakın). İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

MQPUT çağrısı için, kuyruk MQOO_PASS_IDENTITY_CONTEXT seçeneğiyle (ya da bunu gösteren bir seçenekle) açılmış olmalıdır. MQPUT1 çağrısı için, MQOO_PASS_IDENTITY_CONTEXT seçeneğiyle MQOPEN çağrısıyla aynı yetki denetimi gerçekleştirilir.

MQPMO_PASS_ALL_CONTEXT

İleti, kendisiyle ilişkilendirilmiş bağlam bilgisine sahip olacak. Bağlam, *Context* alanında belirtilen kuyruk tanıtıcısından alınır. İleti bağlamıyla ilgili daha fazla bilgi için [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

MQPUT çağrısı için, kuyruk MQOO_PASS_ALL_CONTEXT seçeneğiyle (ya da bunu gösteren bir seçenekle) açılmış olmalıdır. MQPUT1 çağrısı için, MQOO_PASS_ALL_CONTEXT seçeneğiyle MQOPEN çağrısıyla aynı yetki denetimi gerçekleştirilir.

MQPMO_SET_IDENTITY_CONTEXT

İleti, kendisiyle ilişkilendirilmiş bağlam bilgisine sahip olacak. Uygulama, MQMD yapısında kimlik bağlamını belirtir. Kaynak bağlamı bilgileri, kuyruk yöneticisi tarafından, MQPMO_DEFAULT_CONTEXT ile aynı şekilde üretilir (değerler için önceki çizelgeye bakın). İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

MQPUT çağrısı için, kuyruk MQOO_SET_IDENTITY_CONTEXT seçeneğiyle (ya da bunu gösteren bir seçenekle) açılmış olmalıdır. MQPUT1 çağrısı için, MQOO_SET_IDENTITY_CONTEXT seçeneğiyle MQOPEN çağrısıyla aynı yetki denetimi gerçekleştirilir.

MQPMO_SET_ALL_CONTEXT

İleti, kendisiyle ilişkilendirilmiş bağlam bilgisine sahip olacak. Uygulama, MQMD yapısındaki kimliği, kökeni ve kullanıcı bağlamını belirtir. İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

MQPUT çağrısı için, kuyruk MQOO_SET_ALL_CONTEXT seçeneğiyle açılmış olmalıdır. MQPUT1 çağrısı için, MQOO_SET_ALL_CONTEXT seçeneğiyle MQOPEN çağrısıyla aynı yetki denetimi gerçekleştirilir.

MQPMO_*_CONTEXT bağlam seçeneklerinden yalnızca birini belirtebilirsiniz. Hiçbirini belirtmezseniz, MQPMO_DEFAULT_CONTEXT varsayılır.

Özellik seçenekleri. Aşağıdaki seçenek, iletinin özellikleriyle ilgilidir:

YALNIZCA MQPMO_MD_FOR_OUTPUT_YALNIZCA

İleti tanımlayıcı parametresi yalnızca, konan iletinin ileti tanımlayıcısını döndürmek için kullanılmalıdır. **MQPMO** yapısının *NewMsgHandle*, *OriginalMsgHandle*ya da her iki alanla ilişkili ileti tanımlayıcı alanları giriş için kullanılmalıdır.

Geçerli bir ileti tanıtıcısı sağlanmazsa, arama **MQRC_MD_ERROR**neden koduyla başarısız olur.

Yanıt seçeneklerini koyun. Aşağıdaki seçenekler, bir MQPUT ya da MQPUT1

çağırısına döndürülen yanıtı denetler. Bu seçeneklerden yalnızca birini belirleyebilirsiniz.

MQPMO_ASYNC_RESPONSE ve MQPMO_SYNC_RESPONSE belirtilmezse, MQPMO_RESPONSE_AS_Q_DEF ya da MQPMO_RESPONSE_AS_TOPIC_DEF varsayılır.

MQPMO_ASYNC_RESPONSE (mqpmo_async_yanıtı)

MQPMO_ASYNC_RESPONSE seçeneği, uygulama kuyruk yöneticisinin çağırısı tamamlamasını beklemeden bir MQPUT ya da MQPUT1 işleminin tamamlanmasını ister. Bu seçeneğin kullanılması, özellikle istemci bağ tanımlarını kullanan uygulamalar için ileti sistemi başarımını artırabilir. Bir uygulama, MQSTAT komutunu kullanarak, önceki zamanuyumsuz çağrılar sırasında bir hata oluşup oluşmadığını düzenli aralıklarla denetleyebilir.

Bu seçenekle, MQMD ' de yalnızca aşağıdaki alanların doldurulacağı garanti edilir;

- ApplIdentityVerileri
- PutApplTipi
- PutApplAdı
- ApplOriginVerileri

Buna ek olarak, seçenekler olarak MQPMO_NEW_MSG_ID ya da MQPMO_NEW_CORREL_ID ya da her ikisi de belirtilirse, döndürülen MsgId ve CorrelId de tamamlanır. (MQPMO_NEW_MSG_ID, boş bir MsgId alanı belirtilerek örtük olarak belirtilebilir).

Yalnızca önce belirtilen alanlar doldurulmuştur. Olağan durumda MQMD ya da MQPMO yapısında döndürülecek diğer bilgiler tanımsız.

MQPUT1için zamanuyumsuz koyma yanıtı istenirken, MQOD yapısında döndürülen ResolvedQName ve ResolvedQMgradı tanımlanmadı.

MQPUT ya da MQPUT1için zamanuyumsuz koyma yanıtı istenirken, CompCode ve MQCC_OK ve MQRC_NONE nedenleri iletinin başarıyla kuyruğa konduğunu göstermez. Zamanuyumsuz koyma yanıtı kullanan ve iletilerin kuyruğa konduğunun doğrulanmasını gerektiren bir MQI uygulaması geliştirirken, koyma işlemlerinden hem CompCode , hem de neden kodlarını denetlemeli ve zamanuyumsuz hata bilgilerini sorgulamak için MQSTAT kullanmalısınız.

Her bir MQPUT ya da MQPUT1 çağırısının başarılı ya da başarısız olması hemen döndürülmesine de, zamanuyumsuz bir çağrı altında oluşan ilk hata daha sonra MQSTAT çağırısıyla saptanabilir.

Syncpoint altındaki kalıcı bir ileti zamanuyumsuz koyma yanıtı kullanılarak teslim edilemezse ve işlemi kesinleştirme girişiminde bulunursanız, kesinleştirme başarısız olur ve hareket MQCC_FAILED tamamlanma kodu ve MQRC_BACKED_OUT nedeni ile geri çekilir. Uygulama, önceki bir MQPUT ya da MQPUT1 başarısızlığının nedenini saptamak için MQSTAT ' ye çağrı yapabilir.

MQPMO_SYNC_RESPONSE

Bu koyma yanıtı tipinin belirtilmesi, MQPUT ya da MQPUT1 işleminin her zaman zamanuyumlu olarak yayınlanmasını sağlar. Koyma işlemi başarılı olursa, MQMD ve MQPMO ' daki tüm alanlar tamamlanır.

Bu seçenek, kuyrukta ya da konu nesnesinde tanımlanan varsayılan koyma yanıtı değerinden bağımsız olarak zamanuyumlu bir yanıt sağlar.

MQPMO_RESPONSE_AS_Q_DEF

Bir MQPUT çağırısı için bu değer belirtilirse, kullanılan koyma yanıtı tipi, uygulama tarafından ilk açıldığında kuyrukta belirtilen DEFRESP değerinden alınır.

- Kuyruk bir küme kuyruğuysa ve bir MQPUT çağrısı için bu değer belirtilirse, kullanılan koyma yanıtı tipi, iletinin yerleştirildiği kuyruğun belirli eşgörünümüne sahip olan *hedef* kuyruk yöneticisinde tanımlanan **DEFPRESP** özniteliğinden alınır.

Küme kuyruğunun birden çok eşgörünümü varsa ve bu öznitelikte farklılık gösterdiklerinde, bunlardan birinin tane seçkullanım seçtiği değer tahmin edilemez. Bu nedenle, bu özniteliği tüm eşgörünümelerde aynı değere ayarlamalısınız. Böyle bir durumda değilse, kuyruk yöneticisi günlüklerine AMQ9407 hata iletisi gönderilir. Ayrıca bkz. [Diğer adlar, uzak kuyruklar ve küme kuyrukları için hedef nesne öznitelikleri nasıl çözülür?](#)

- Kuyruk bir küme kuyruğu değilse ve bir MQPUT çağrısı için bu değer belirtilirse, kullanılan koyma yanıtı tipi, hedef kuyruk yöneticisi uzak olsa bile, *yemel* kuyruk yöneticisinde tanımlanan **DEFPRESP** özniteliğinden alınır.

Bir istemci uygulaması IBM WebSphere MQ 7.0düzeyinden önceki bir düzeyde bir kuyruk yöneticisine bağlıysa, MQPMO_SYNC_RESPONSE belirtilmiş gibi davranır.

Bir MQPUT1 çağrısı için bu seçenek belirtilirse, istek sunucuya gönderilmeden önce DEFPRESP özniteliğinin değeri bilinmez. Varsayılan olarak, MQPUT1 çağrısı MQPMO_SYNCPOINT kullanıyorsa, MQPMO_ASYNC_RESPONSE için olduğu gibi davranır ve MQPMO_NO_SYNCPOINT kullanıyorsa, MQPMO_SYNC_RESPONSE için olduğu gibi davranır. Ancak, istemci yapılandırma dosyasında Put1DefaultAlwaysSync özelliğini ayarlayarak bu varsayılan davranışı geçersiz kılabilirsiniz; bkz. [İstemci yapılandırma dosyasının KANAL kısmı.](#)

MQPMO_RESPONSE_AS_TOPIC_DEF

MQPMO_RESPONSE_AS_TOPIC_DEF, konu nesneleriyle kullanılmak üzere MQPMO_RESPONSE_AS_Q_DEF ile eşanlamdır.

Diğer seçenekler. Aşağıdaki seçenekler yetki denetimini, kuyruk yöneticisi susturulduğunda ve kuyruk ve kuyruk yöneticisi adları çözüldüğünde ne olacağını denetler:

MQPMO_ALTERNATE_USER_AUTHORITY

MQPMO_ALTERNATE_USER_AUTHORITY, MQPUT1 çağrısının **ObjDesc** değiştirgesindeki *AlternateUserId* alanının, iletileri kuyruğa koymak için yetkinin geçerliliğini denetlemek için kullanılacak bir kullanıcı kimliği içerdiğini gösterir. Çağrı yalnızca *AlternateUserId* 'in belirtilen seçeneklerle kuyruğu açma yetkisi varsa, uygulamanın altında çalıştığı kullanıcı kimliğinin bunu yapma yetkisi olup olmadığına bakılmaksızın başarılı olabilir. (Bu, belirtilen bağlam seçenekleri için geçerli değildir; ancak, her zaman uygulamanın çalıştığı kullanıcı kimliğiyle karşılaştırılarak denetlenir.)

Bu seçenek yalnızca MQPUT1 çağrısıyla geçerlidir.

MQPMO_FAIL_IF QUIESCING

Bu seçenek, kuyruk yöneticisi susturma durumundaysa, MQPUT ya da MQPUT1 çağrısına başarısız olmasına neden olur.

z/OSsistemlerinde bu seçenek, bağlantı (bir CICS ya da IMS uygulaması için) susturma durumundaysa, MQPUT ya da MQPUT1 çağrısına da başarısız olmasına neden olur.

Çağrı, MQRC_Q_MGR QUIESCING ya da MQRC_CONNECTION QUIESCING neden koduyla MQCC_FAILED tamamlanma kodunu döndürür.

MQPMO_RESOLVE_LOCAL_Q

MQPMO yapısındaki *ResolvedQName* ögesini, iletinin konduğu yerel kuyruğun adıyla ve *ResolvedQMgrName* yerel kuyruğu barındıran yerel kuyruk yöneticisinin adıyla doldurmak için bu seçeneği kullanın. MQPMO_RESOLVE_LOCAL_Q ile ilgili daha fazla bilgi için [MQOO_RESOLVE_LOCAL_Q](#) konusuna bakın.

Kuyruğa koyma yetkiniz varsa, MQPUT çağrısında bu işareti belirtmek için gerekli yetkiye sahipsiniz; özel yetkiye gerek yoktur.

Varsayılan seçenek. Açıklanan seçeneklerin hiçbirine gerek duymazsanız, aşağıdaki seçeneği kullanın:

MQPMO_NONE

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar. MQPMO_NONE, program belgelerine yardımcı olmak üzere tanımlandı; bu seçeneğin başka

bir ögeyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

MQPMO_NONE bir giriş alanıdır. *Options* alanının ilk değeri MQPMO_NONE.

MQPMO için zamanaşımı (MQLONG)

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri şudur: -1.

MQPMO için Bağlam (MQHOBJ)

MQPMO_PASS_IDENTITY_CONTEXT ya da MQPMO_PASS_ALL_CONTEXT belirtilirse, bu alan, konan iletiyle ilişkilendirilecek bağlam bilgilerinin alındığı giriş kuyruğu tanıtıcısını içermelidir.

MQPMO_PASS_IDENTITY_CONTEXT ya da MQPMO_PASS_ALL_CONTEXT belirtilmezse, bu alan yoksayıdır.

Bu bir giriş alanıdır. Bu alanın ilk değeri şudur: 0.

MQPMO için KnownDestSayısı (MQLONG)

Bu, yürürlükteki MQPUT ya da MQPUT1 çağrısının yerel kuyruklar olan dağıtım listesindeki kuyruklara başarıyla gönderdiği iletilerin sayısıdır. Bu sayı, uzak kuyruklara çözülecek kuyruklara gönderilen iletileri içermez (iletiyi saklamak için başlangıçta yerel bir iletim kuyruğu kullanılsa bile). Bu alan, dağıtım listesinde olmayan tek bir kuyruğa ileti yerleştirilirken de ayarlanır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dir. *Version* , MQPMO_VERSION_1' den küçükse bu alan ayarlanmaz.

Dağıtım listeleri desteklenmediği için bu alan z/OS üzerinde tanımlı değil.

UnknownDestMQPMO için sayı (MQLONG)

Bu, yürürlükteki MQPUT ya da MQPUT1 çağrısının dağıtım listesindeki uzak kuyruklara çözülecek kuyruklara başarıyla gönderdiği iletilerin sayısıdır. Kuyruk yöneticisinin dağıtım listesi formunda geçici olarak tuttuğu iletiler, bu dağıtım listelerinin içerdiği tek tek hedeflerin sayısı olarak sayılır. Bu alan, dağıtım listesinde olmayan tek bir kuyruğa ileti yerleştirilirken de ayarlanır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dir. *Version* , MQPMO_VERSION_1' den küçükse bu alan ayarlanmaz.

Dağıtım listeleri desteklenmediği için bu alan z/OS üzerinde tanımlı değil.

InvalidDestMQPMO için sayı (MQLONG)

Bu, dağıtım listesindeki kuyruklara gönderilemeyen iletilerin sayısıdır. Sayı, açılmayan kuyrukların yanı sıra, başarıyla açılan, ancak koyma işleminin başarısız olduğu kuyrukları da içeriyor. Bu alan, dağıtım listesinde olmayan tek bir kuyruğa ileti yerleştirilirken de ayarlanır.

Not: Bu alan, MQPUT ya da MQPUT1 çağrısındaki **CompCode** değiştirgesi MQCC_OK ya da MQCC_WARNING; **CompCode** değiştirgesi MQCC_FAILED ise, ancak uygulama kodunda buna güvenmiyorsanız ayarlanabilir.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dir. *Version* , MQPMO_VERSION_1' den küçükse bu alan ayarlanmaz.

Dağıtım listeleri desteklenmediği için bu alan z/OS üzerinde tanımlı değil.

MQPMO için ResolvedQName (MQCHAR48)

Ad çözme işlemi yerel kuyruk yöneticisi tarafından gerçekleştirildikten sonra hedef kuyruğun adıdır. Döndürülen ad, *ResolvedQMgrName* ile tanımlanan kuyruk yöneticisinde bulunan bir kuyruğun adıdır.

Boş olmayan bir değer, nesne tek bir kuyruksa döndürülür; nesne bir dağıtım listesi ya da konu ise, döndürülen değer tanımsız olur.

Bu bir çıkış alanıdır. Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

ResolvedQMGrMQPMO için Ad (MQCHAR48)

Bu, ad çözme işlemi yerel kuyruk yöneticisi tarafından gerçekleştirildikten sonra hedef kuyruk yöneticisinin adıdır. Döndürülen ad, *ResolvedQName* ile tanımlanan kuyruğa sahip olan kuyruk yöneticisinin adıdır ve yerel kuyruk yöneticisinin adı olabilir.

ResolvedQName, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun sahip olduğu paylaşılan bir kuyruksa, *ResolvedQMGrName*, kuyruk paylaşım grubunun adıdır. Kuyruk başka bir kuyruk paylaşım grubuna aitse, *ResolvedQName* kuyruk paylaşım grubunun adı ya da kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisinin adı olabilir (döndürülen değer türü, yerel kuyruk yöneticisinde var olan kuyruk tanımlamalarına göre belirlenir).

Boş olmayan bir değer, nesne tek bir kuyruksa döndürülür; nesne bir dağıtım listesi ya da konu ise, döndürülen değer tanımsız olur.

Bu bir çıkış alanıdır. Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQPMO için RecsPresent (MQLONG)

Bu, uygulama tarafından sağlanan MQPMR koyma iletisi kayıtlarının ya da MQRR yanıt kayıtlarının sayısıdır. Bu sayı, ileti bir dağıtım listesine konuluyorsa sıfırdan büyük olabilir. Konan ileti kayıtları ve yanıt kayıtları isteğe bağlıdır; uygulamanın herhangi bir kayıt sağlaması gerekmez ya da yalnızca bir tip kayıt sağlamayı seçebilir. Ancak, uygulama her iki tipten de kayıt sağlarsa, her tipten *RecsPresent* kayıt sağlamalıdır.

RecsPresent değeri, dağıtım listesindeki hedef sayısı ile aynı olmalıdır. Çok fazla kayıt sağlanırsa, fazla kayıt kullanılmaz; çok az kayıt sağlanırsa, koyma iletisi kayıtları olmayan hedeflere ilişkin ileti özellikleri için varsayılan değerler kullanılır (bkz. *PutMsgRecOffset*).

RecsPresent sıfırdan küçükse ya da sıfırdan büyükse, ancak ileti dağıtım listesine konmuyorsa, çağrı başarısız olur; neden kodu MQRC_RECS_PRESENT_ERROR.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0'dır. *Version*, MQPMO_VERSION_2' den küçükse bu alan yoksayılr.

PutMsgRecFields (MQLONG)

Bu alan, uygulama tarafından sağlanan koyma iletisi kayıtlarında hangi MQPMR alanlarının bulunduğunu gösteren işaretleri içerir. *PutMsgRecFields* seçeneğini yalnızca ileti bir dağıtım listesine konurken kullanın. *RecsPresent* sıfırsa ya da *PutMsgRecOffset* ve *PutMsgRecPtr* her ikisi de sıfırsa alan yoksayılr.

Varolan alanlar için, kuyruk yöneticisi her hedef için karşılık gelen koyma iletisi kaydındaki alanlardaki değerleri kullanır. Var olmayan alanlar için, kuyruk yöneticisi MQMD yapısındaki değerleri kullanır.

Koyma iletisi kayıtlarında hangi alanların bulunduğunu belirtmek için aşağıdaki işaretlerden birini ya da birkaçını kullanın:

MQPMRF_MSG_ID

İleti tanıtıcısı alanı var.

MQPMRF_CORREL_ID

İlinti tanıtıcısı alanı var.

MQPMRF_GROUP_ID

Grup tanıtıcısı alanı var.

MQPMRF_FEEDBACK

Geribildirim alanı var.

MQPMRF_ACCOUNTING_TOKEN

Muhasebe simgesi alanı var.

Bu işareti belirtirseniz, *Options* alanında `MQPMO_SET_IDENTITY_CONTEXT` ya da `MQPMO_SET_ALL_CONTEXT` belirtin; bu koşul karşılanmazsa, çağrı `MQRC_PMO_RECORD_FLAGS_ERROR` neden koduyla başarısız olur.

`MQPMR` alanı yoksa, aşağıdaki belirtilebilir:

`MQPMRF_NONE`

Bir koyma iletisi kayıt alanı yok.

Bu değer belirtilirse, *RecsPresent* sıfır ya da her ikisi de *PutMsgRecOffset* ve *PutMsgRecPtr* sıfır olmalıdır.

`MQPMRF_NONE`, program belgelerine yardımcı olmak için tanımlandı. Bu değişimin başka bir değişimle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

PutMsgRecFields geçerli olmayan işaretler içeriyorsa ya da koyma iletisi kayıtları sağlanıyorsa, ancak *PutMsgRecFields* `MQPMRF_NONE` değerine sahipse, çağrı `MQRC_PMO_RECORD_FLAGS_ERROR` neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri `MQPMRF_NONE`. *Version*, `MQPMO_VERSION_2`'den küçükse bu alan yoksayılır.

***PutMsgRecOffset* (MQPMO için MQLONG)**

Bu, `MQPMO` yapısının başlangıcından itibaren ilk `MQPMR` koyma iletisi kaydının bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir. *PutMsgRecOffset* yalnızca ileti bir dağıtım listesine konurken kullanılır. *RecsPresent* sıfırsa alan yoksayılır.

İleti bir dağıtım listesine konduğunda, her hedef için iletinin belirli özelliklerini tek tek belirtmek üzere bir ya da daha fazla `MQPMR` koyma iletisi kaydı dizisi sağlanabilir; bu özellikler şunlardır:

- İleti Tanıtıcısı
- İlti tanıtıcısı
- Grup tanıtıcısı
- Geribildirim değeri
- Muhasebe simgesi

Tüm bu özellikleri belirtmenize gerek yoktur, ancak seçtiğiniz altküme her neyse, alanları doğru sırada belirtin. Daha fazla ayrıntı için `MQPMR` yapısının tanımına bakın.

Genellikle, dağıtım listesi açıldığında `MQOD` tarafından belirtilen nesne kayıtları kadar çok koyma iletisi kaydı olmalıdır; her bir koyma iletisi kaydı, ilgili nesne kaydı tarafından tanımlanan kuyruk için ileti özelliklerini sağlar. Dağıtım listesindeki açılmayan kuyruklar, bu durumda ileti özellikleri yoksayılsa da, dizideki uygun konumlara kendilerine ayrılmış ileti kayıtlarını yerleştirmeye devam etmelidir.

Koyma iletisi kaydı sayısı, nesne kaydı sayısından farklı olabilir. Nesne kayıtlarından daha az koyma iletisi kaydı varsa, konan ileti kayıtları olmayan hedeflere ilişkin ileti özellikleri, ileti tanımlayıcısı `MQMD` 'deki ilgili alanlardan alınır. Nesne kayıtlarından daha fazla koyma iletisi kaydı varsa, fazlalık kullanılmaz (bunlara erişmek yine de mümkün olmalıdır). Koyma iletisi kayıtları isteğe bağlıdır, ancak bunlar sağlanırsa *RecsPresent* olmalıdır.

PutMsgRecOffset içinde bir görelî konum belirterek ya da *PutMsgRecPtr* içinde bir adres belirterek `MQOD` 'daki nesne kayıtlarına benzer bir şekilde koyma iletisi kayıtlarını sağlayın; Bunun nasıl yapılacağını öğrenmek için "[MQOD-Nesne tanımlayıcı](#)" sayfa 476 içinde açıklanan *ObjectRecOffset* alanına bakın.

Birden fazla *PutMsgRecOffset* ve *PutMsgRecPtr* kullanılamaz; her ikisi de sıfır değilse, çağrı `MQRC_PUT_MSG_RECORDS_ERROR` neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *Version*, `MQPMO_VERSION_2`'den küçükse bu alan yoksayılır.

***ResponseRecMQPMO* için görelî konum (MQLONG)**

Bu, MQPMO yapısının başlangıcından itibaren ilk MQRR yanıt kaydının bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir. *ResponseRecOffset* yalnızca ileti bir dağıtım listesine konurken kullanılır. *RecsPresent* sıfırda alan yoksayılr.

İletiyi bir dağıtım listesine koyarken, iletinin başarıyla gönderilmediği kuyrukları (MQRR 'deki *CompCode* alanı) ve her başarısızlığın nedenini (MQRR' deki *Reason* alanı) saptamak için bir ya da daha çok MQRR yanıt kaydı dizisi sağlayabilirsiniz. Kuyruk açılmadığı için ya da koyma işlemi başarısız olduğu için ileti gönderilmemiş olabilir. Kuyruk yöneticisi yanıt kayıtlarını yalnızca çağrı sonucu karışık olduğunda (yani, bazı iletiler diğerleri başarısız olduğunda ya da tümü farklı nedenlerle başarısız olduğunda) ayarlar; çağrıdaki MQRC_MULTIPLE_REASON neden kodu bu durumu gösterir. Aynı neden kodu tüm kuyruklar için de geçerliyse, MQPUT ya da MQPUT1 çağrısının **Reason** değıştirgesinde bu neden döndürölür ve yanıt kayıtları ayarlanmaz.

Genellikle, dağıtım listesi açıldığında MQOD tarafından belirtilen nesne kaydı sayısı kadar yanıt kaydı vardır; gerektiğinde, her yanıt kaydı, ilgili nesne kaydı tarafından belirlenen kuyruğa konmaya ilişkin tamamlanma koduna ve neden koduna ayarlanır. Dağıtım listesindeki açılmayan kuyrukların, dizideki uygun konumlarda kendilerine ayrılmış yanıt kayıtları olmalıdır; ancak bunlar, koyma işlemi yerine, açık işlemden kaynaklanan tamamlanma kodu ve neden koduna ayarlı olmalıdır.

Yanıt kaydı sayısı, nesne kaydı sayısından farklı olabilir. Nesne kayıtlarından daha az yanıt kaydı varsa, uygulama, koyma işleminin başarısız olduğu tüm hedefleri ya da başarısızlığın nedenlerini belirleyemeyebilir. Nesne kayıtlarından daha fazla yanıt kaydı varsa, fazlalık kullanılmaz (bunlara erişmek yine de mümkün olmalıdır). Yanıt kayıtları isteğe bağlıdır, ancak bunlar sağlanırsa *RecsPresent* olmalıdır.

ResponseRecOffset içinde bir görelî konum belirterek ya da *ResponseRecPtr* içinde bir adres belirterek MQOD 'daki nesne kayıtlarına benzer bir şekilde yanıt kayıtlarını sağlayın; Bunun nasıl yapılacağını öğrenmek için “MQOD-Nesne tanımlayıcı” sayfa 476 içinde açıklanan *ObjectRecOffset* alanına bakın. Ancak, *ResponseRecOffset* ve *ResponseRecPtr* ürünlerinden en fazla birini kullanın; Her ikisi de sıfır değilse, çağrı MQRC_RESPONSE_RECORDS_ERROR neden koduyla başarısız olur.

MQPUT1 çağrısı için bu alan sıfır olmalıdır. Bunun nedeni, yanıt bilgilerinin (istenirse) nesne tanımlayıcısı MQOD tarafından belirtilen yanıt kayıtlarında döndürölmesi olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *Version* , MQPMO_VERSION_2' den küçükse bu alan yoksayılr.

***PutMsgRecPtr* (MQPMO için MQPTR)**

Bu, ilk MQPMR koyma iletisi kaydının adresidir. *PutMsgRecPtr* seçeneğini yalnızca ileti bir dağıtım listesine konurken kullanın. *RecsPresent* sıfırda alan yoksayılr.

Koyma iletisi kayıtlarını belirtmek için *PutMsgRecPtr* ya da *PutMsgRecOffset* kullanılabilir, ancak her ikisini birden kullanamazsınız; ayrıntılar için bkz. “*PutMsgRecOffset* (MQPMO için MQLONG)” sayfa 509. *PutMsgRecPtr* kullanmıyorsanız, boş değeri gösterge ya da boş değeri bayt olarak ayarlayın.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değeri gösterge ve tersi durumda tümü boş değeri bir bayt dizisidir. *Version* , MQPMO_VERSION_2' den küçükse bu alan yoksayılr.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizisi olarak bildirilir ve ilk değeri tümüyle boş değeri byte dizisi olur.

***ResponseRecMQPMO* için *Ptr* (MQPTR)**

Bu, ilk MQRR yanıt kaydının adresidir. *ResponseRecPtr* yalnızca ileti bir dağıtım listesine konurken kullanılır. *RecsPresent* sıfırda alan yoksayılr.

Yanıt kayıtlarını belirtmek için *ResponseRecPtr* ya da *ResponseRecOffset* kullanın, ancak her ikisini de belirtmeyin; ayrıntılar için bkz. “*ResponseRecMQPMO* için görelî konum (MQLONG)” sayfa 509. *ResponseRecPtr* kullanmıyorsanız, boş değeri gösterge ya da boş değeri bayt olarak ayarlayın.

MQPUT1 çağrısı için bu alan boş değeri gösterge ya da boş değeri byte olmalıdır. Bunun nedeni, yanıt bilgilerinin (istenirse) nesne tanımlayıcısı MQOD tarafından belirtilen yanıt kayıtlarında döndürölmesi olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizgisidir. *Version* , MQPMO_VERSION_2' den küçükse bu alan yoksayılr.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir ve ilk değer tümüyle boş değerli byte dizgisi olur.

OriginalMsgMQPMO için tanıtıcısı (MQHMSG)

Bu, bir iletinin isteğe bağlı bir tanıtıcısıdır. Önceden bir kuyruktan alınmış olabilir. Bu tutamacın kullanılması *Action* alanının değerine tabidir; ayrıca bkz. [NewMsgHandle](#).

Özgün ileti tanıtıcısının içeriği, **MQPUT** ya da **MQPUT1** çağrısı tarafından değiştirilmez.

Bu bir giriş alanıdır. Bu alanın ilk değeri şudur: **MQHM_NONE**. Sürüm **MQPMO_VERSION_3**değerinden küçükse bu alan yoksayılr.

NewMsgMQPMO için tanıtıcısı (MQHMSG)

Bu, İşlem alanının değerine tabi olarak konmakta olan iletinin isteğe bağlı bir tanıtıcısıdır. İletinin özelliklerini tanımlar ve belirtildiyse, *OriginalMsgHandle* değerlerini geçersiz kılar.

MQPUT ya da **MQPUT1** çağrısından döndüğünüzde, tutamacın içeriği gerçekte yerleştirilen iletiyi yansıtır.

Bu bir giriş alanıdır. Bu alanın ilk değeri şudur: **MQHM_NONE**. Sürüm **MQPMO_VERSION_3**değerinden küçükse bu alan yoksayılr.

MQPMO için İşlem (MQLONG)

Bu, gerçekleştirilmekte olan koyma değeri tipini ve OriginalMsgHandle alanı tarafından belirtilen özgün ileti ile NewMsgHandle alanı tarafından belirtilen yeni ileti arasındaki ilişkiyi belirtir. İletinin özellikleri, kuyruk yöneticisi tarafından belirtilen işlemin değerine göre seçilir.

MQPUT ya da MQPUT1 çağrılarında MsgDesc değiştirgesini kullanarak ileti tanımlayıcısının içeriğini belirtmeyi seçebilirsiniz. Diğer bir seçenek olarak, MsgDesc değiştirgesi sağlanamaz ya da MQPMO yapısının Options (Seçenekler) alanına MQPMO_MD_FOR_OUTPUT_ONLY eklenerek yalnızca çıkış olarak belirtilebilir.

MsgDesc değiştirgesi sağlanmazsa ya da yalnızca çıkış olarak belirtildiyse, yeni iletiye ilişkin ileti tanımlayıcı, bu konuda açıklanan kurallara göre MQPMO ' nun ileti tanıtıcısı alanlarından doldurulur.

Denetim bağlamı bilgileri kısmında açıklanan bağlam ayarı ve aktarma etkinlikleri, ileti tanımlayıcısı oluşturulduktan sonra yürürlüğe girer.

Yanlış bir işlem değeri belirtilirse, çağrı MQRC_ACTION_ERROR neden koduyla başarısız olur.

Aşağıdaki işlemlerden herhangi biri belirtilebilir:

MQACTP_YENI

Yeni bir ileti konuluyor ve program tarafından önceki bir iletiyle ilişki belirlenmiyor. İleti tanımlayıcısı aşağıdaki gibi oluşturulur:

- MQPUT ya da MQPUT1 çağrısında MsgDesc belirtilirse ve MQPMO_MD_FOR_OUTPUT_ONLY MQPMO.Options, değiştirilmemiş ileti tanımlayıcısı olarak kullanılır.
- MsgDesc sağlanmazsa ya da MQPMO_MD_FOR_OUTPUT_ONLY MQPMO.Options daha sonra, kuyruk yöneticisi OriginalMsgHandle ve NewMsgHand'indeki özelliklerin birleşimini kullanarak ileti tanımlayıcısını oluşturur. Yeni ileti tanıtıcısında belirttik olarak ayarlanan tüm ileti tanımlayıcı alanları, özgün ileti tanıtıcısında belirlenenlerden önceliklidir.

İleti verileri MQPUT ya da MQPUT1 Arabellek değiştirgesinden alınır.

MQACTP_FORWARD

Önceden alınan bir ileti iletiliyor. Özgün ileti tanıtıcısı, önceden alınan iletiyi belirler.

Yeni ileti tanıtıcısı, özgün ileti tanıtıcısındaki özelliklerde (ileti tanımlayıcısı da içinde olmak üzere) yapılan değişiklikleri belirler.

İleti tanımlayıcısı aşağıdaki gibi oluşturulur:

- MQPUT ya da MQPUT1 çağrısında MsgDesc belirtilirse ve MQPMO_MD_FOR_OUTPUT_ONLY MQPMO.Options, değiştirilmemiş ileti tanımlayıcısı olarak kullanılır.
- MsgDesc sağlanmazsa ya da MQPMO_MD_FOR_OUTPUT_ONLY MQPMO.Options daha sonra, kuyruk yöneticisi OriginalMsgHandle ve NewMsgHand'indeki özelliklerin bir birleşimini kullanarak ileti tanımlayıcısını oluşturur. Yeni ileti tanıtıcısında belirtik olarak ayarlanan tüm ileti tanımlayıcı alanları, özgün ileti tanıtıcısında belirlenenlerden önceliklidir.
- MQPMO.Options, daha sonra bunlar onurlandırıldı.

İleti özellikleri aşağıdaki gibi oluşturulur:

- MQPD.CopyOptions
- Yeni ileti tanıtıcısından tüm özellikler. Yeni ileti tanıtıcısındaki, özgün ileti tanıtıcısındaki bir özellik aynı adı taşıyan her özellik için, değer yeni ileti tanıtıcısından alınır. Bu kuralın tek kural dışı durumu, yeni ileti tanıtıcısındaki özellik, özgün ileti tanıtıcısındaki bir özellik aynı ada sahip olduğunda, ancak özelliğin değeri boş olduğunda özel durumdur. Bu durumda özellik iletiden kaldırılır.

İletilecek ileti verileri MQPUT ya da MQPUT1 Arabellek değiştirgesinden alınır.

MQACTP_REPLY

Daha önce alınan bir iletiye yanıt gönderiliyor. Özgün ileti tanıtıcısı, önceden alınan iletiyi belirler.

Yeni ileti tanıtıcısı, özgün ileti tanıtıcısındaki özelliklerde (ileti tanımlayıcısı da içinde olmak üzere) yapılan değişiklikleri belirler.

İleti tanımlayıcısı aşağıdaki gibi oluşturulur:

- MQPUT ya da MQPUT1 çağrısında MsgDesc belirtilirse ve MQPMO_MD_FOR_OUTPUT_ONLY MQPMO.Options, değiştirilmemiş ileti tanımlayıcısı olarak kullanılır.
- MsgDesc sağlanmazsa ya da MQPMO_MD_FOR_OUTPUT_ONLY MQPMO.Options, ardından ilk ileti tanımlayıcı alanları aşağıdaki gibi seçilir:

<i>Çizelge 509. Yanıt iletisi tanıtıcısı dönüşümü</i>	
MQMD 'deki Alan	Kullanılan değer
Rapor	MQRO_PASS_DISCARD_AND_EXPIRY ise ve MQRO_DISCARD_MSG ayarlandı: MQRO_DISCARD_MSG tersi durumda MQRO_NONE
MsgType	MQMT_REPLY
Son kullanma tarihi	MQRO_PASS_DISCARD_AND_EXPIRY ise ayarlı: Giriş iletisinden kopyalandı tersi durumda MQEI_UNLIMITED
Geri bildirim	MQFB_NONE

Çizelge 509. Yanıt iletisi tanıtıcısı dönüşümü (devamı var)	
MQMD 'deki Alan	Kullanılan değer
MsgId	MQPMO_NEW_MSG_ID ayarlandıysa: Yeni bir ileti tanıtıcısı oluşturuldu MQRO_PASS_MSG_ID ayarlandıysa: Giriş iletisinden kopyalandı tersi durumda MQMI_NONE
CorrelId	MQPMO_NEW_CORREL_ID ayarlandıysa: Yeni bir ilinti tanıtıcısı oluşturuldu MQRO_COPY_MSG_ID_TO_CORREL_ID ayarlandıysa: Kopyalandığı alan: MsgId giriş iletisi MQRO_PASS_CORREL_ID ayarlandıysa: Kopyalandığı alan: CorrelId giriş iletisi tersi durumda MQCI_NONE
BackoutCount	0
ReplyToQ	Boşluklar
ReplyToQMgr	Boşluklar
GroupId	MQGI_NONE
MsgSeqNumarası	1
Görelî Konum	0
MsgFlags	MQMF_NONE
OriginalLength	MQOL_UNDEFINED

- İleti tanımlayıcı, yeni ileti tanıtıcısı tarafından değiştirilir-yeni ileti tanıtıcısında belirtik olarak özellikler olarak belirlenen tüm ileti tanımlayıcı alanları, daha önce açıklandığı gibi, ileti tanımlayıcı alanlarından önceliklidir.

İleti özellikleri aşağıdaki gibi oluşturulur:

- MQPD.CopyOptions
- Yeni ileti tanıtıcısından tüm özellikler. Yeni ileti tanıtıcısındaki, özgün ileti tanıtıcısındaki bir özellik ile aynı adı taşıyan her özellik için, değer yeni ileti tanıtıcısından alınır. Bu kuralın tek kural dışı durumu, yeni ileti tanıtıcısındaki özellik, özgün ileti tanıtıcısındaki bir özellik ile aynı adı sahip olduğunda, ancak özelliğin değeri boş olduğunda özel durumdur. Bu durumda özellik iletiden kaldırılır.

İletilecek ileti verileri MQPUT/MQPUT1 Arabellek değiştirilmesinden alınır.

MQACTP_REPORT

Daha önce alınan bir iletinin sonucu olarak bir rapor oluşturuluyor. Özgün ileti tanıtıcısı, raporun oluşturulmasına neden olan iletiyi belirtir.

Yeni ileti tanıtıcısı, özgün ileti tanıtıcısındaki özelliklerde (ileti tanımlayıcısı da içinde olmak üzere) yapılan değişiklikleri belirler.

İleti tanımlayıcısı aşağıdaki gibi oluşturulur:

- MQPUT ya da MQPUT1 çağrısında MsgDesc belirtilirse ve MQPMO_MD_FOR_OUTPUT_ONLY MQPMO.Options, değiştirilmemiş ileti tanımlayıcısı olarak kullanılır.

- MsgDesc sağlanmazsa ya da MQPMO_MD_FOR_OUTPUT_ONLY MQPMO.Options ve ilk ileti tanımlayıcı alanları aşağıdaki gibi seçilir:

<i>Çizelge 510. Rapor ileti tanıtıcısı dönüşümü</i>	
MQMD 'deki Alan	Kullanılan değer
Rapor	MQRO_PASS_DISCARD_AND_EXPIRY ve MQRO_DISCARD_MSG ayarlandı: MQRO_DISCARD_MSG tersi durumda MQRO_NONE
MsgType	MQMT_REPORT
Son kullanma tarihi	MQRO_PASS_DISCARD_AND_EXPIRY ise ayarlı: Giriş iletisinden kopyalandı tersi durumda MQEI_UNLIMITED
MsgId	MQPMO_NEW_MSG_ID ayarlandıysa: Yeni bir ileti tanıtıcısı oluşturuldu MQRO_PASS_MSG_ID ayarlandıysa: Giriş iletisinden kopyalandı tersi durumda MQMI_NONE
CorrelId	MQPMO_NEW_CORREL_ID ayarlandıysa: Yeni bir ilinti tanıtıcısı oluşturuldu MQRO_COPY_MSG_ID_TO_CORREL_ID ayarlandıysa: Kopyalandığı alan: MsgId giriş iletisi MQRO_PASS_CORREL_ID ayarlandıysa: Kopyalandığı alan: CorrelId giriş iletisi tersi durumda MQCI_NONE
BackoutCount	0
ReplyToQ	Boşluklar
ReplyToQMgr	Boşluklar
OriginalLength	<i>BufferLength</i> olarak ayarlayın

- İleti tanımlayıcı, yeni ileti tanıtıcısı tarafından değiştirilir-yeni ileti tanıtıcısında belirttik olarak özellikler olarak belirlenen tüm ileti tanımlayıcı alanları, daha önce açıklandığı gibi, ileti tanımlayıcı alanlarından önceliklidir.

İleti özellikleri aşağıdaki gibi oluşturulur:

- MQPD.CopyOptions
- Yeni ileti tanıtıcısından tüm özellikler. Yeni ileti tanıtıcısındaki, özgün ileti tanıtıcısındaki bir özellik ile aynı adı taşıyan her özellik için, değer yeni ileti tanıtıcısından alınır. Bu kuralın tek kural dışı durumu, yeni ileti tanıtıcısındaki özellik, özgün ileti tanıtıcısındaki bir özellik ile aynı adı sahip olduğunda, ancak özelliğin değeri boş olduğunda özel durumdur. Bu durumda özellik iletiden kaldırılır.

Sonuçtaki MQMD 'deki Geribildirim alanı, oluşturulacak raporu temsil eder. MQFB_NONE geribildirim değeri, MQPUT ya da MQPUT1 çağrısının MQRC_FEEDBACK_ERROR neden koduyla başarısız olmasına neden olur.

Rapor iletisinin kullanıcı verilerini seçmek için IBM MQ , sonuçtaki MQMD 'deki Rapor ve Geribildirim alanlarına ve MQPUT ya da MQPUT1 çağrısının Arabellek ve BufferLength değerlerine danışır.

- Geribildirim MQFB_COA, MQFB_COD ya da MQFB_EXPIRATION ise, Rapor değeri incelenir.
- Aşağıdaki durumlardan herhangi biri doğruysa, Arabellek 'ten gelen tüm ileti verileri BufferLength uzunluğuna göre kullanılır.
 - Geribildirim MQFB_SÜRE sonu ve Rapor MQRO_EXPIRATION_WITH_FULL_DATA içeriyor
 - Geribildirim MQFB_COD ve Rapor MQRO_COD_WITH_FULL_DATA içeriyor
 - Geribildirim MQFB_COA ve Rapor MQRO_COA_WITH_FULL_DATA içeriyor
- Aşağıdaki durumlardan herhangi biri true ise, iletinin ilk 100 baytı (ya da 100 'den küçükse BufferLength) kullanılır.
 - Geribildirim MQFB_SÜRE bitimi ve Rapor MQRO_EXPIRATION_WITH_DATA içeriyor
 - Geribildirim MQFB_COD ve Rapor MQRO_COD_WITH_DATA içeriyor
 - Geribildirim MQFB_COA ve Rapor MQRO_COA_WITH_DATA içeriyor
- Geribildirim MQFB_SÜRE sonu, MQFB_COD ya da MQFB_COA ise ve Rapor, o Feedback değeriyle ilgili *_WITH_FULL_DATA ya da *_WITH_DATA seçeneklerini içermiyorsa, iletiye kullanıcı verileri dahil edilmez.
- Feedback yukarıda listelenenlerden farklı bir değer alırsa, Arabellek ve BufferLength olağan olarak kullanılır.

Önceki listede açıklanan kullanıcı verilerinin türetilmesi de aşağıdaki çizelgede gösterilmiştir:

<i>Çizelge 511. Kullanıcı verilerinin kaynağı</i>			
	MQFB_COA	MQFB_COD	MQFB_SÜRE SONU
MQRO_SÜRE SONU _WITH_FULL_DATA	Yok	Yok	Arabellek (Arabellek uzunluğu)
_FULL_DATA İLE MQRO_COD_WITH	Yok	Arabellek (Arabellek uzunluğu)	Yok
_FULL_DATA İLE MQRO_COA_WITH	Arabellek (Arabellek uzunluğu)	Yok	Yok
MQRO_SÜRE SONU _WITH_DATA	Yok	Yok	Arabellek (İlk 100 bayt)
MQRO_COD_WITH_DA TA	Yok	Arabellek (İlk 100 bayt)	Yok
MQRO_COA_WITH_DA TA	Arabellek (İlk 100 bayt)	Yok	Yok

MQPMO için PubLevel (MQLONG)

Bu alanın ilk değeri 9 'dur. Bu yayın tarafından hedeflenen abonelik düzeyi. Yalnızca bu değerden küçük ya da bu değere eşit en yüksek SubLevel 'e sahip abonelikler bu yayını alır. Bu değer sıfır ile 9 aralığında olmalıdır; sıfır en düşük düzeydir. Ancak, bir yayını alıkonduysa, PubLevel 1 'de yeniden yayınlandığından, artık daha yüksek düzeylerde aboneler tarafından kullanılamaz.

Bilgi için bkz. [Intercepting publications](#).

MQPMR-Put-message kaydı

Bir iletiyi dağıtım listesine koyarken tek bir hedefe ilişkin çeşitli ileti özelliklerini belirtmek için MQPMR yapısını kullanın. MQPMR, MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş/çıkış yapısıdır.

Kullanılabilirlik

MQPMR yapısı aşağıdaki altyapılarda kullanılabilir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

Karakter kümesi ve kodlama

MQPMR içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında yer almalıdır. Ancak, uygulama bir MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Kullanım

MQPUT ya da MQPUT1 çağrısında bu yapılardan oluşan bir dizi sağlayarak, dağıtım listesindeki her hedef kuyruk için farklı değerler belirtebilirsiniz. Alanlardan bazıları yalnızca giriş, diğerleri giriş/çıkış.

Not: Bu yapı, sabit bir düzene sahip olmadığı için alışılmadık bir yapıdır. Bu yapıdaki alanlar isteğe bağlıdır ve her alanın varlığı ya da yokluğu, MQPMO 'daki *PutMsgRecFields* alanındaki işaretlerle gösterilir. **olan alanlar aşağıdaki sırayla oluşmalıdır :**

- *MsgId*
- *CorrelId*
- *GroupId*
- *Feedback*
- *AccountingToken*

Eksik olan alanlar, kayıta yer kaplamıyor.

MQPMR sabit bir yerleşim düzenine sahip olmadığı için, üstbilgide, COPY ve INCLUDE dosyalarında desteklenen programlama dillerine ilişkin bir tanımlama sağlanmaz. Uygulama programcısı, uygulamanın gerektirdiği alanları içeren bir bildirim yaratmalı ve var olan alanları belirtmek için *PutMsgRecFields* içindeki işaretleri ayarlamalıdır.

Alanlar

Desteklenen programlama dilleri için üstbilgi, COPYve INCLUDE dosyalarında yapı bildirimleri sağlanmadığı için bu yapı için tanımlanmış başlangıç değeri yoktur. Örnek bildirimler, tüm alanlar gerekliyse yapının nasıl bildirileceğini gösterir.

Alan adı	Alan açıklaması
<u>MsgId</u>	İleti Tanıtıcısı
<u>CorrelId</u>	İlinti tanıtıcısı

Çizelge 512. MQPMM içindeki alanlar (devamı var)

Alan adı	Alan açıklaması
GroupId	Grup tanıtıcısı
Geri bildirim	Geribildirim ya da neden kodu
AccountingToken	Muhasebe simgesi

Dil bildirimleri

MQPMM için C bildirim

```
typedef struct tagMQPMM MQPMM;  
struct tagMQPMM {  
    MQBYTE24  MsgId;           /* Message identifier */  
    MQBYTE24  CorrelId;        /* Correlation identifier */  
    MQBYTE24  GroupId;         /* Group identifier */  
    MQLONG    Feedback;        /* Feedback or reason code */  
    MQBYTE32  AccountingToken; /* Accounting token */  
};
```

MQPMM için COBOL bildirim

```
** MQPMM structure  
10 MQPMM.  
** Message identifier  
15 MQPMM-MSGID PIC X(24).  
** Correlation identifier  
15 MQPMM-CORRELID PIC X(24).  
** Group identifier  
15 MQPMM-GROUPID PIC X(24).  
** Feedback or reason code  
15 MQPMM-FEEDBACK PIC S9(9) BINARY.  
** Accounting token  
15 MQPMM-ACCOUNTINGTOKEN PIC X(32).
```

MQPMM için PL/I bildirim

```
dcl  
1 MQPMM 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 */
```

MQPMM için Visual Basic bildirim

```
Type MQPMM  
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
```

MQPMM için MsgId (MQBYTE24)

Bu, kuyruğa gönderilen ileti için, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen bir adla kullanılacak ileti tanıtıcısıdır. Tek bir kuyruğa koyma değeri için MQMD 'deki *MsgId* alanıyla aynı şekilde işlenir.

Bu alan MQPMM kaydında yoksa ya da hedeflerden daha az MQPMM kaydı varsa, *MsgId* alanı içeren bir MQPMM kaydı olmayan hedefler için MQMD 'deki değeri kullanılır. Bu değeri MQMI_NONE ise, bu

hedeflerin *her biri* için yeni bir ileti tanıtıcısı oluşturulur (yani, bu hedeflerin ikisi aynı ileti tanıtıcısına sahip değildir).

MQPMO_NEW_MSG_ID belirtilirse, MQPMR kayıtları olup olmadığına bakılmaksızın, dağıtım listesindeki tüm hedefler için yeni ileti tanıtıcıları oluşturulur. Bu, MQPMO_NEW_CORREL_ID ' nin işlenmesinden farklıdır (bkz. *CorrelId* alanı).

Bu bir giriş/çıkış alanıdır.

MQPMR için CorrelId (MQBYTE24)

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen bir adla kuyruğa gönderilen ileti için kullanılacak ilinti tanıtıcısıdır. Tek bir kuyruğa koyma değeri için MQMD ' deki *CorrelId* alanıyla aynı şekilde işlenir.

Bu alan MQPMR kaydında yoksa ya da hedeflerden daha az MQPMR kaydı varsa, *CorrelId* alanı içeren bir MQPMR kaydı olmayan hedefler için MQMD ' deki değer kullanılır.

MQPMO_NEW_CORREL_ID belirtilirse, MQPMR kayıtları olup olmadığına bakılmaksızın, dağıtım listesindeki tüm hedefler için *tek* yeni bir ilinti tanıtıcısı oluşturulur ve kullanılır. Bu, MQPMO_NEW_MSG_ID ' nin işleniş biçiminden farklıdır (bkz. *MsgId* alanı).

Bu bir giriş/çıkış alanıdır.

MQPMR için GroupId (MQBYTE24)

GroupId , MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen adı taşıyan kuyruğa gönderilen ileti için kullanılacak grup tanıtıcısıdır. Tek bir kuyruğa koyma değeri için MQMD ' deki *GroupId* alanıyla aynı şekilde işlenir.

Bu alan MQPMR kaydında yoksa ya da hedeflerden daha az MQPMR kaydı varsa, *GroupId* alanı içeren bir MQPMR kaydı olmayan hedefler için MQMD ' deki değer kullanılır. Değer, Kuyruktaki fiziksel sıralama içinde belgelendiği şekilde işlenir, ancak aşağıdaki farklılıklar gösterir:

- GroupId , QMName ve bir zaman damgasından yaratılır. Bu nedenle, bir GroupId ' nı benzersiz olarak tutmak için kuyruk yöneticisi adlarını da benzersiz tutun. Ayrıca, kuyruk yöneticileri makinesinde saatleri yeniden ayarlamayın.
- Yeni bir grup tanıtıcısının kullanılacağı durumlarda, kuyruk yöneticisi her hedef için farklı bir grup tanıtıcısı oluşturur (yani, iki hedef aynı grup tanıtıcısına sahip değildir).
- Alandaki değer kullanılacağı durumlarda, çağrı MQRC_GROUP_ID_ERROR neden koduyla başarısız olur.

Bu bir giriş/çıkış alanıdır.

MQPMR için geribildirim (MQLONG)

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen adla kuyruğa gönderilen ileti için kullanılacak geribildirim kodudur. Tek bir kuyruğa koyma değeri için MQMD ' deki *Feedback* alanıyla aynı şekilde işlenir.

Bu alan yoksa, MQMD ' deki değer kullanılır.

Bu bir giriş alanıdır.

MQPMR için AccountingToken (MQBYTE32)

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen adla kuyruğa gönderilen ileti için kullanılacak hesap simgesidir. Tek bir kuyruğa koyma değeri için MQMD ' deki *AccountingToken* alanıyla aynı şekilde işlenir. Bu alanın içeriğine ilişkin bilgi için "MQMD-İleti tanımlayıcı" sayfa 420 içindeki *AccountingToken* açıklamasına bakın.

Bu alan yoksa, MQMD ' deki değer kullanılır.

Bu bir giriş alanıdır.

MQRFH-Kurallar ve biçimlendirme üstbilgisi

MQRFH yapısı, kuralların ve biçimlendirme üstbilgisinin düzenini tanımlar. Dizilim verilerini ad-değer çiftleri biçiminde göndermek için bu üstbilgiyi kullanın.

Kullanılabilirlik

Tüm IBM MQ sistemleri ve bu sistemlere bağlı IBM MQ MQI clients .

Biçim Adı

MQFMT_RF_HEADER

Karakter kümesi ve kodlama

MQRFH yapısındaki alanlar (*NameValueString*dahil), MQRFH ' den önce gelen üstbilgi yapısındaki *CodedCharSetId* ve *Encoding* alanları tarafından verilen karakter kümesinde ve kodlamasında ya da MQRFH uygulama iletisi verilerinin başlangıcında ise MQMD yapısındaki bu alanlar tarafından bulunur.

Karakter takımı, kuyruk adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 513. MQRFH için MQRFH ' deki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQRFH_STRUC_ID	'RFH↵'
<u>Sürüm</u> (yapı sürümü numarası)	MQRFH_VERSION_1	1
<u>StrucLength</u> (MQRFH yapısının bayt cinsinden uzunluğu)	MQRFH_STRUC_LENGTH_FIXED	32
<u>Kodlama</u> (<i>NameValueString</i> ' i izleyen verilerin sayısal kodlaması)	MQENC_NATIVE	Ortama bağlıdır
<u>CodedCharSetId</u> (<i>NameValueString</i> ' u izleyen verilerin karakter kümesi tanıtıcısını belirtir)	MQCCSI_AYRILMAMIŞ	0
<u>Biçim</u> (aşağıdaki verilerin biçim adı <i>NameValueString</i>)	MQFMT_NONE	Boşluklar
<u>İşaretler</u> (işaretler)	MQRFH_NONE	0
<u>NameValueString</u> (ad-değer çiftlerini içeren değişken uzunluklu karakter dizisi)	yok	yok
Notlar:		
1. ↵ simgesi, tek bir boş karakteri temsil eder.		
2. C programlama dilinde, makro değişkeniMQRFH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:		
<pre>MQRFH MyRFH = {MQRFH_DEFAULT};</pre>		

Dil bildirimleri

MQRFH için C bildirimi

```
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 */
};
```

MQRFH için COBOL bildirimi

```
** 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.
```

MQRFH için PL/I bildirimi

```
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 */
```

MQRFH için High Level Assembler bildirimi

```
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
MQRFH_AREA DS CL(MQRFH_LENGTH)
```


MQRFH için Visual Basic bildirimi

```
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
```

MQRFH için StrucId (MQCHAR4)

Bu, kuralların yapı tanıtcısı ve biçimlendirme üstbilgi yapısıdır. Her zaman bir giriş alanıdır. Değeri MQRFH_STRUC_ID.

Değer şöyle olmalıdır:

MQRFH_STRUC_ID

Kurallar ve biçimlendirme üstbilgi yapısına ilişkin tanıtcı.

C programlama dili için, değişmez MQRFH_STRUC_ID_ARRAY da tanımlanır. Bu değer MQRFH_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQRFH için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQRFH_VERSION_1

Version-1 kuralları ve üstbilgi yapısını biçimlendirme.

Bu alanın ilk değeri MQRFH_VERSION_1' dir.

MQRFH için StrucLength (MQLONG)

Bu, yapının sonundaki *NameValueString* alanı da içinde olmak üzere MQRFH yapısının bayt cinsinden uzunluğudur. Uzunluk, *NameValueString* alanını izleyen kullanıcı verilerini içermez.

Bazı ortamlardaki kullanıcı verilerini dönüştürürken ortaya çıkan sorunları önlemek için *StrucLength* , dörtlü bir çarpı olmalıdır.

Aşağıdaki değişmez, *NameValueString* alanı dışında, yapının *fixed* kısmının uzunluğunu gösterir:

MQRFH_STRUC_LENGTH_FIXED

MQRFH yapısının sabit kısmının uzunluğu.

Bu alanın ilk değeri MQRFH_STRUC_LENGTH_FIXED.

MQRFH için kodlama (MQLONG)

Bu, *NameValueString* ; ' yi izleyen verilerin sayısal kodlamasını belirtir. MQRFH yapısının kendisindeki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri MQENC_NATIVE.

CodedCharSetId (MQRFH için MQLONG)

Bu, *NameValueString* ; ' den sonra gelen verilerin karakter kümesi tanıtcısını belirtir. MQRFH yapısının kendisindeki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değer kullanılabilir:

MQCCSI_INHERIT

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı MQCCSI_INHERIT değerini döndürmez.

MQMD 'deki *PutAppLType* alanının değeri MQAT_BROKER ise, MQCSI_INHERIT kullanılamaz.

Bu alanın ilk değeri MQCCSI_UNDEFINED 'dir.

MQRFH için biçim (MQCHAR8)

Bu, *NameValueString*'i izleyen verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD 'deki *Format* alanıyla aynıdır.

Bu alanın ilk değeri MQFMT_NONE.

MQRFH için İşaretler (MQLONG)

Aşağıdaki belirtilebilir:

MQRFH_NONE

Bayrak yok.

Bu alanın ilk değeri MQRFH_NONE.

MQRFH için NameValueDizesi (MQCHARn)

Bu, formda ad-değer çiftlerini içeren değişken uzunluklu bir karakter dizisidir:

```
name1 value1 name2 value2 name3 value3 ...
```

Her ad ya da değer, bitişik addan ya da değerden bir ya da daha çok boşluk karakteriyle ayrılmalıdır; bu boşluklar önemli değildir. Bir ad ya da değer, adı ya da değeri çift tırnak işareti içine alarak ve son ek olarak ekleyerek anlamlı boşluklar içerebilir; açık çift tırnak işareti ile eşleşen çift tırnak işareti arasındaki tüm karakterler önemli olarak değerlendirilir. Aşağıdaki örnekte ad FAMOUS_WORDS ve değer Hello World' dir:

```
FAMOUS_WORDS "Hello World"
```

Bir ad ya da değer, boş karakter dışında herhangi bir karakter içerebilir (*NameValueString* için sınırlayıcı görevi görür). Ancak, bir uygulamanın birlikte çalışabilmesini sağlamak için adları aşağıdaki karakterlerle sınırlayabilir:

- İlk karakter: büyük ya da küçük harf alfabetik (A-Z ya da a-z) ya da alt çizgi.
- Sonraki karakterler: büyük ya da küçük harf alfabetik, ondalık basamak (0-9), alt çizgi, tire ya da nokta.

Bir ad ya da değer bir ya da daha çok çift tırnak işareti içeriyorsa, ad ya da değer çift tırnak içine alınmalı ve dizedeki her çift tırnak işareti çift tırnak içine alınmalıdır:

```
Famous_Words "The program displayed ""Hello World"""
```

Adlar ve değerler büyük ve küçük harfe duyarlıdır; diğer bir değişle, küçük harfler büyük harflerle aynı olarak kabul edilmez. Örneğin, FAMOUS_WORDS ve Famous_Words iki farklı addır.

NameValueString byte cinsinden uzunluk *StrucLength* eksi MQRFH_STRUC_LENGTH_FIXED değerine eşittir. Bazı ortamlardaki kullanıcı verilerini dönüştürürken ortaya çıkan sorunları önlemek için, bu uzunluğu dörtlü bir çarpı olarak belirleyin. Bu uzunlukta boşluklarla *NameValueString* 'i takın ya da dizilimdeki son anlamlı karakterin ardından boş bir karakter yerleştirerek daha önce sonlandırın. Belirtilen *NameValueString* uzunluğuna kadar olan boş karakter ve ardından gelen bayt dikkate alınmaz.

Not: Bu alanın uzunluğu değişmez olmadığı için, alan, desteklenen programlama dilleri için sağlanan yapının bildirimlerinden çıkarılır.

MQRFH2 -Kurallar ve biçimlendirme üstbilgisi 2

MQRFH2 üstbilgisi MQRFH üstbilgisine dayalıdır, ancak Unicode dizgilerinin çeviri olmadan taşınmasına izin verir ve sayısal veri tiplerini taşıyabilir. MQRFH2 yapısı, version-2 kurallarının ve biçimlendirme üstbilgisinin biçimini tanımlar. XML benzeri bir sözdizimi kullanılarak kodlanmış verileri göndermek için bu üstbilgiyi kullanırsınız. Bir ileti seri olarak iki ya da daha fazla MQRFH2 yapısı içerebilir; kullanıcı verileri isteğe bağlı olarak serideki son MQRFH2 yapısını takip edebilir.

Kullanılabilirlik

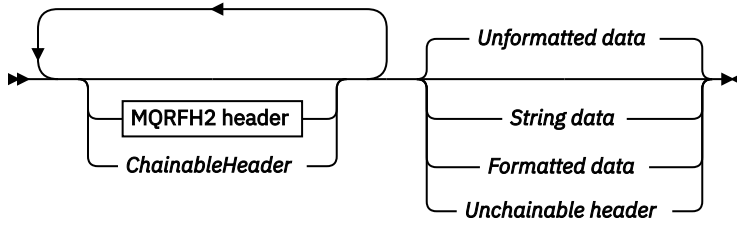
Tüm IBM MQ sistemleri ve bu sistemlere bağlı IBM MQ MQI clients .

Biçim Adı

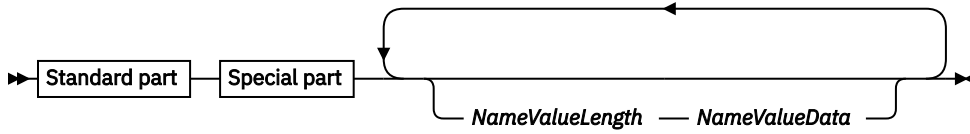
MQFMT_RF_HEADER_2

Syntax

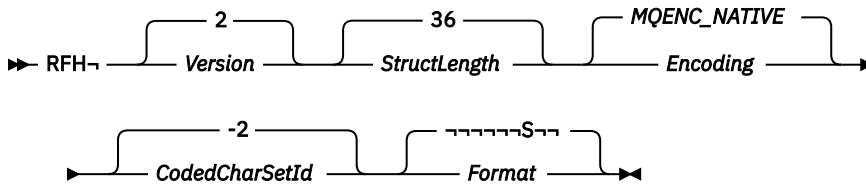
IBM MQ Message



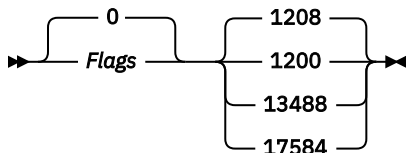
MQRFH2 header



Standard part



Special part



Karakter kümesi ve kodlama

Özel kurallar, MQRFH2 yapısı için kullanılan karakter kümesi ve kodlama için geçerlidir:

- *NameValueData* dışındaki alanlar, MQRFH2' den önce gelen üstbilgi yapısındaki *CodedCharSetId* ve *Encoding* alanları tarafından verilen karakter kümesinde ve kodlamasında ya da MQRFH2 uygulama iletisi verilerinin başındaki ise MQMD yapısındaki bu alanlar tarafından bulunur.

Karakter takımı, kuyruk adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır.

MQGET çağrısında MQGMO_CONVERT belirtildiğinde, kuyruk yöneticisi MQRFH2 alanlarını *NameValueData*da istenen karakter kümesine ve kodlamaya dönüştürür.

- *NameValueData* , *NameValueCCSID* alanı tarafından verilen karakter kümesinde bulunur. *NameValueCCSID* ; için yalnızca listelenen Unicode karakter kümeleri geçerlidir. Ayrıntılar için *NameValueCCSID* açıklamasına bakın.

Bazı karakter kümelerinin kodlamaya bağlı bir gösterimi vardır. *NameValueCCSID* bu karakter kümelerinden biriye, *NameValueData* , MQRFH2içindeki diğer alanlarla aynı kodlamada olmalıdır.

MQGET çağrısında MQGMO_CONVERT belirtildiğinde, kuyruk yöneticisi *NameValueData* ' yi istenen kodlamaya dönüştürür, ancak karakter kümesini değiştirmez.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 514. MQRFH2 için MQRFH2 içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
<u>StrucId</u> (yapı tanıtıcısı)	MQRFH_STRUC_ID	' RFH↵ '
<u>Sürüm</u> (yapı sürümü numarası)	MQRFH_VERSION_2	2
<u>StrucLength</u> (MQRFH2 yapısının bayt cinsinden uzunluğu)	MQRFH_STRUC_LENGTH_FIXED_2	36
<u>Kodlama</u> (son <i>NameValueData</i> alanını izleyen verilerin sayısal kodlaması)	MQENC_NATIVE	Ortama bağlıdır
<u>CodedCharSetId</u> (son <i>NameValueData</i> alanını izleyen verilerin karakter kümesi tanıtıcısı)	MQCCSI_INHERIT	-2
<u>Biçim</u> (son <i>NameValueData</i> alanını izleyen verilerin biçim adı)	MQFMT_NONE	Boşluklar
<u>İşaretler</u> (işaretler)	MQRFH_NONE	0
<u>NameValueCCSID</u> (<i>NameValueData</i> alanındaki verilerin kodlanmış karakter takımı tanıtıcısı)	Yok	1208
<u>NameValueLength</u> (<i>NameValueData</i> alanındaki verilerin bayt cinsinden uzunluğu)	Yok	None

Çizelge 514. MQRFH2 için MQRFH2 içindeki alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
NameValueVerileri (ileti özelliklerinin ad-değer çiftleri)	Yok	Yok

Notlar:

- ↪ simgesi, tek bir boş karakteri temsil eder.
- C programlama dilinde, makro değişkeniMQRFH2_DEFAULT çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQRFH2 MyRFH2 = {MQRFH2_DEFAULT};
```

Dil bildirimleri

MQRFH2 için C bildirimini

```
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 */
};
```

MQRFH2 için COBOL bildirimini

```
** 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-STRUCLNGTH 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.
```

MQRFH2 için PL/I bildirimini

```
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 */

```

MQRFH2 için High Level Assembler bildirimi

```

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
MQRFH_AREA           ORG    MQRFH
MQRFH_AREA           DS    CL(MQRFH_LENGTH)

```

MQRFH2 için Visual Basic bildirimi

```

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 (MQCHAR4) for MQRFH2

Bu, kuralların yapı tanıttıcısıdır ve üstbilgi iki yapısını biçimlendirmektedir. Her zaman bir giriş alanıdır. Değeri: MQRFH2_STRUC_ID.

Değer şöyle olmalıdır:

MQRFH2_STRUC_ID

Kurallar ve biçimlendirme üstbilgisi iki yapısına ilişkin tanıttıcı.

C programlama dili için, MQRFH2_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu, MQRFH2_STRUC_ID ile aynı değere sahip, ancak bir dizgi yerine bir karakter dizisi.

MQRFH2 için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQRFH_VERSION_2

Version-2 kuralları ve üstbilgi yapısını biçimlendirme.

Bu alanın ilk değeri MQRFH_VERSION_2' dir.

StrucLength (MQLONG) for MQRFH2

Bu, yapının sonundaki *NameValueLength* ve *NameValueData* alanları da içinde olmak üzere MQRFH2 yapısının bayt cinsinden uzunluğudur. Yapının sonunda aşağıdaki sırayla birden çok *NameValueLength* ve *NameValueData* alanı çifti olması geçerlidir:

```
length1, data1, length2, data2, ...
```

StrucLength , yapının sonundaki son *NameValueData* alanını izleyebilecek kullanıcı verilerini içermez.

Bazı ortamlarda kullanıcı verilerinin dönüştürülmesiyle ilgili sorunları önlemek için *StrucLength* , dörtlü bir çarpı olmalıdır.

Aşağıdaki değişmez, yapının *fixed* kısmının uzunluğunu, yani *NameValueLength* ve *NameValueData* alanlarını dışlamayan uzunluğunu verir:

MQRFH_STRUC_LENGTH_FIXED_2

MQRFH2 yapısının sabit kısmının uzunluğu.

Bu alanın ilk değeri MQRFH_STRUC_LENGTH_FIXED_2' dir.

MQRFH2 için kodlama (MQLONG)

Bu, son *NameValueData* alanını izleyen verilerin sayısal kodlamasını belirtir; MQRFH2 yapısındaki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri MQENC_NATIVE.

CodedCharSetId (MQLONG) for MQRFH2

Bu, son *NameValueData* alanını izleyen verilerin karakter kümesi tanıtıcısını belirtir; MQRFH2 yapısındaki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Aşağıdaki özel değer kullanılabilir:

MQCCSI_INHERIT

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı MQCCSI_INHERIT değerini döndürmez.

MQMD ' deki *PutApplType* alanının değeri MQAT_BROKER ise, MQCSI_INHERIT kullanılamaz.

Bu alanın ilk değeri MQCCSI_INHERIT.

MQRFH2 için biçim (MQCHAR8)

Bu, son *NameValueData* alanını izleyen verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *Format* alanıyla aynıdır.

Bu alanın ilk değeri MQFMT_NONE.

MQRFH2 için işaretler (MQLONG)

Bu alanın ilk değeri şudur: MQRFH_NONE. MQRFH_NONE belirtilmelidir.

MQRFH_NONE

Bayrak yok.

MQRFH_INTERNAL

MQRFH2 üstbilgisi, dahili olarak ayarlanan özellikleri içerir.

MQRFH_INTERNAL , kuyruk yöneticisinin kullanımı içindir.

En üstteki 16 bit, MQRFH_FLAGS_RESTRICTED_MASK, kuyruk yöneticisinin ayarladığı işaretler için ayrılmıştır. Bir kullanıcının ayarlayabileceği işaretler, en alttaki 16 bitte tanımlanır.

NameValue MQRFH2

Bu, *NameValueData* alanındaki verilerin kodlanmış karakter takımı tanıtıcısını belirtir. Bu, MQRFH2 yapısındaki diğer dizgilerin karakter kümesinden farklıdır ve yapının sonundaki son *NameValueData* alanını izleyen verilerin karakter kümesinden (varsa) farklı olabilir.

NameValueCCSID aşağıdaki değerlerden birine sahip olmalıdır:

CCSID	Anlamı
1200	UTF-16, desteklenen en son Unicode sürümü
13488	UTF-16, Unicode sürümü 2.0 altkümesi
17584	UTF-16, Unicode sürümü 3.0 altkümesi (Euro simgesini içerir)
1208	UTF-8, desteklenen en son Unicode sürümü

UTF-16 karakter kümeleri için *NameValueData* kodlaması (byte sırası), MQRFH2 yapısındaki diğer alanların kodlamasıyla aynı olmalıdır.

Unicode Temel Çok Dilli Plane 'nin (üstteki U + FFFF) ötesindeki karakterler, UTF-16 olarak vekil kod noktalarıyla (X'D800'-X'DFFF') ya da UTF-8 içindeki dört bayt ile temsil edilir.

Not: *NameValueCCSID* yukarıda listelenen değerlerden birine sahip değilse ve MQRFH2 yapısı MQGET çağrısında dönüştürme gerektiriyorsa, çağrı MQRC_SOURCE_CCSID_ERROR neden koduyla tamamlanır ve ileti dönüştürülmeden döndürülür.

Bu alanın ilk değeri 1208 'dir.

NameValueLength (MQLONG) for MQRFH2

İlgili *NameValueData* alanının uzunluğu

Bu, *NameValueData* alanındaki verilerin bayt cinsinden uzunluğunu belirtir. *NameValueLength* , dörtlü bir çarpan olmalıdır.

Not: *NameValueLength* ve *NameValueData* alanları isteğe bağlıdır, ancak varsa çift olarak oluşmaları ve bitişik olmaları gerekir. Alan çifti gerektiği kadar yinelenebilir, örneğin:

```
length1 data1 length2 data2 length3 data3
```

Bu alanlar isteğe bağlı olduğundan, desteklenen çeşitli programlama dilleri için sağlanan yapının bildirimlerinden çıkarılır.

NameValue MQRFH2 için veri (MQCHARn)

NameValueData , ileti özelliklerinin ad-değer çiftlerini içeren bir klasörü içeren değişken uzunluklu bir alandır. Klasör, XML benzeri bir sözdizimi kullanılarak kodlanmış verileri içeren değişken uzunluklu bir karakter dizgisidir. Karakter dizgisinin bayt cinsinden uzunluğu, *NameValueData* alanından önce gelen *NameValueLength* alanı tarafından verilir. Uzunluk, dört çarpı olmalıdır.

NameValueLength ve *NameValueData* alanları isteğe bağlıdır, ancak varsa çift olarak oluşmaları ve bitişik olmaları gerekir. Alan çifti gerektiği kadar yinelenebilir, örneğin:

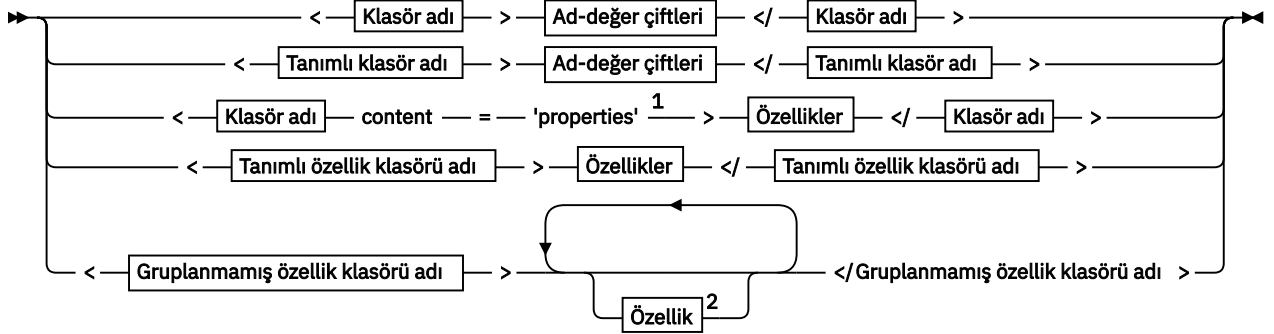
```
length1 data1 length2 data2 length3 data3
```


NameValueData , MQGET çağrısında belirtilen karakter kümesine dönüştürülmez. İleti MQGMO_CONVERT seçeneğiyle alınmış olsa da *NameValueData* , özgün karakter kümesinde kalır. Ancak *NameValueData* , MQGET çağrısında belirtilen kodlamaya dönüştürülür.

Notlar:

- Bu alanlar isteğe bağlı olduğundan, desteklenen çeşitli programlama dilleri için sağlanan yapının bildirimlerinden çıkarılır.
- Sözdizimi şemasında "tanımlı" ve "ayrılmış" terimleri kullanılır. "Tanımlı" , adın IBM MQ tarafından kullanıldığı anlamına gelir. "Ayrılmış" , adın ileride IBM MQ tarafından kullanılmak üzere ayrıldığı anlamına gelir.

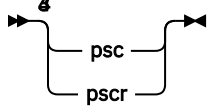
NameValueData sözdizimi



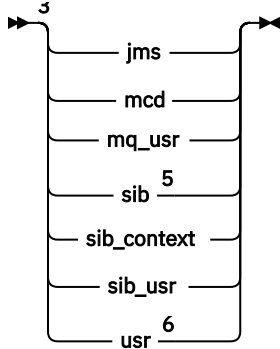
Klasör adı



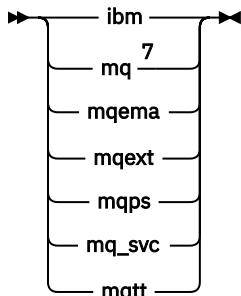
Tanımlı klasör adı



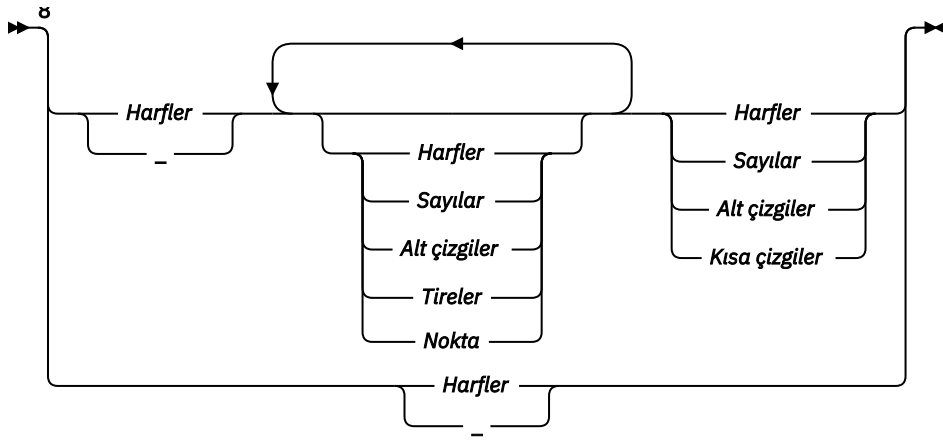
Tanımlı özellik klasörü adı



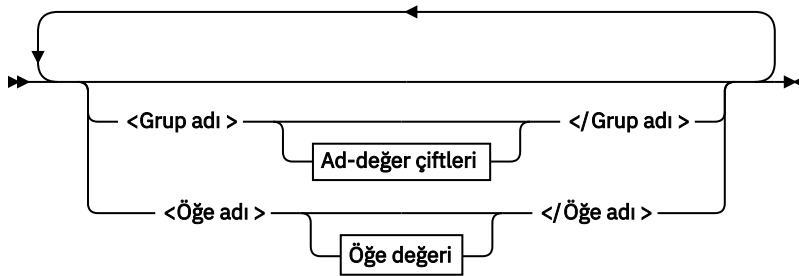
Gruplanmamış özellik klasörü adı



Ad



Ad-değer çiftleri



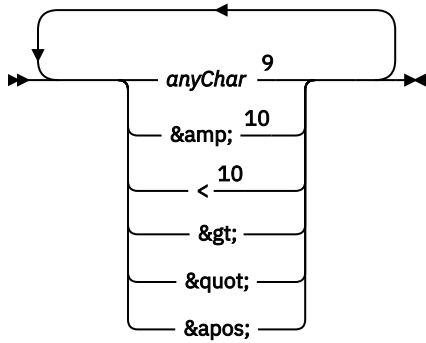
Grup adı

NAME

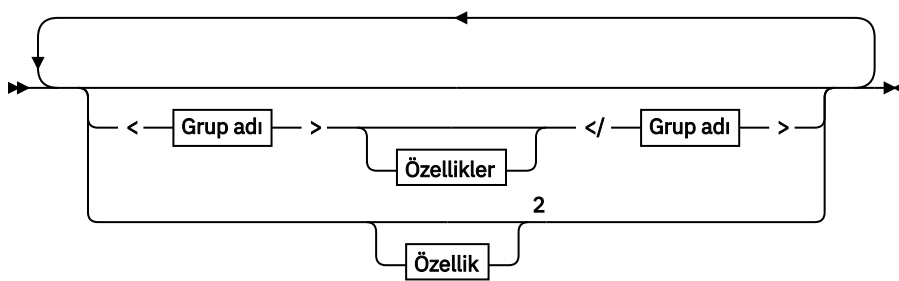
Öğe adı

NAME

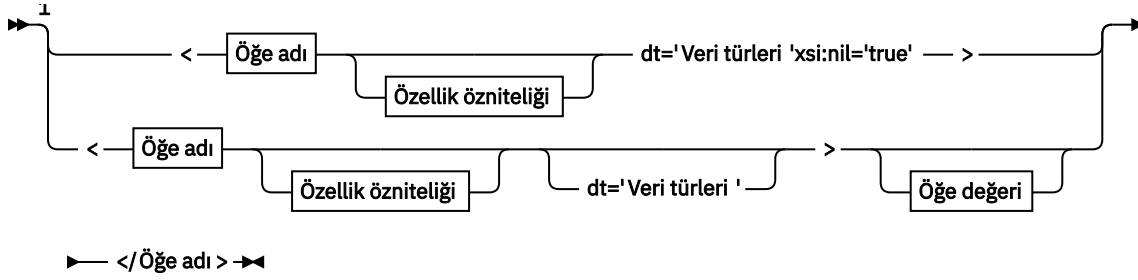
Öğe değeri



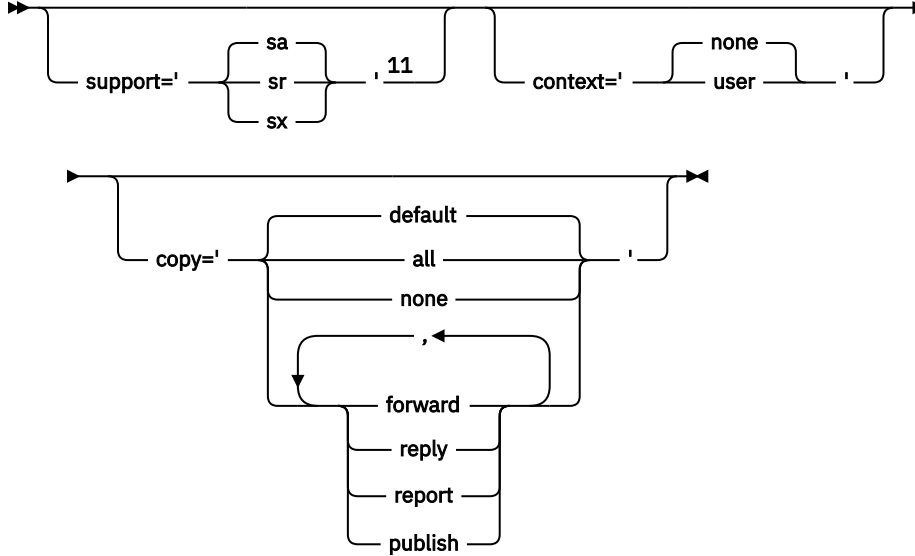
Özellikler



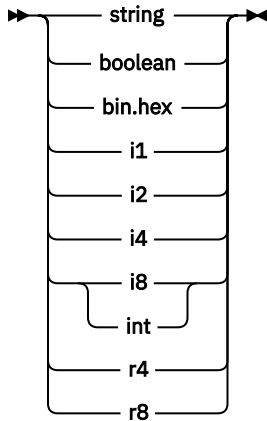
Özellik



Özellik özneliği



Veri türleri



Notlar:

- 1 Çift tırnak işaretleri ya da tek tırnak işaretleri geçerlidir.
- 2 Geçersiz bir özellik adı kullanmayın; bkz. "Özellik adı geçersiz" sayfa 542. Ayrılmış bir özellik adını yalnızca tanımlı amacı için kullanın; bkz. "Tanımlı özellik adları" sayfa 542.
- 3 Ad küçük harfli olmalıdır.
- 4 Yalnızca bir psc ve pscı klasörü desteklenir.
- 5 WebSphere Application Server Service Integration Bus , sonraki MQRFH2 üstbilgilerinde sib, sib_contextve sib_usr klasörlerini yoksayar ve yalnızca ilk MQRFH2 üstbilgisindeki özellikler önemlidir.
- 6 Bir MQRFH2içinde birden çok usı klasörü bulunmamalıdır. usı klasöründeki özellikler bir kereden fazla oluşmamalıdır.
- 7 Yalnızca ilk mq klasöründeki özellikler önemlidir. Klasör UTF-8ise, yalnızca tek bayt UTF-8 karakterleri desteklenir. Tek beyaz alan karakteri Unicode U+0020' dur.

⁸ Geçerli karakterler W3C XML belirtiminde tanımlanır ve esasen Unicode kategorilerinden oluşur Ll, Lu, Lo, Lt, Nl, Mc, Mn, Lm, ve Nd ; Bkz. Unicode karakter kategorileri.

⁹ Tüm karakterler anlamlıdır. Baştaki ve sondaki boşluklar öge değerinin bir parçasıdır.

¹⁰ Geçersiz bir karakter kullanmayın; bkz. “Geçersiz karakterler” sayfa 542. Bu geçersiz karakterler yerine bir çıkış sırası kullanın.

¹¹ Destek özelliği özniteliği yalnızca mq klasöründe geçerlidir.

Klasör adı

NameValueData tek bir klasör içerir. Birden çok klasör oluşturmak için birden çok *NameValueData* alanı oluşturun. Bir ileti içindeki tek bir MQRFH2 üstbilgisinde birden çok *NameValueData* alanı oluşturabilirsiniz. Alternatif olarak, her biri birden çok *NameValueData* alanı içeren birden çok zincirli MQRFH2 üstbilgisi oluşturabilirsiniz.

MQRFH2 üstbilgilerinin sırası ve *NameValueData* alanlarının sırası, bir klasörün mantıksal içeriği için fark yaratmaz. Aynı klasör bir iletide birden çok kez bulunursa, klasör bir bütün olarak ayrıştırılır. Aynı özellik aynı klasörün birden çok örneğindeyse, bir liste olarak ayrıştırılır.

MQRFH2 ögesinin doğru ayrıştırması, bir klasörün iletide fiziksel olarak saklanabileceği diğer yöntemlerden etkilenmez.

Dört klasör bu kuralı izlemiyor. Yalnızca mq, sib, sib_contextve sib_usı klasörünün ilk örneği ayrıştırılır.

Aynı özellik zincirleme MQRFH2 üstbilgilerinin birleşik içeriğinde bir kereden fazla ortaya çıkarsa, özelliğin yalnızca ilk eşgörünümü ayrıştırılır. MQSETMPgibi bir API çağrısı kullanılarak bir özellik ayarlanırsa ve doğrudan bir uygulama tarafından MQRFH2 ' a eklenirse, API çağrısı önceliklidir.

Klasör adı, ad-değer çiftlerini ya da gruplarını içeren bir klasörün adıdır. Gruplar ve ad-değer çiftleri, klasör ağacında aynı düzeyde karıştırılabilir; bkz. Şekil 1 sayfa 532. Bir grup adı ile öge adını birleştirmeyin; bkz. Şekil 2 sayfa 532

```
<group1><nvp1>value</nvp1></group1><group2><nvp2>value</nvp2></group2>
<group3><nvp1>value</nvp1></group3><nvp3>value</nvp3>
```

Şekil 1. Grupların ve ad-değer çiftlerinin doğru kullanımları

```
<group1><nvp1> value </nvp1> value </group1>
```

Şekil 2. Grupların ve ad-değer çiftlerinin yanlış kullanımı

Geçersiz ya da ayrılmış bir klasör adı kullanmayın; bkz. “Yol adı geçersiz” sayfa 542 ve “Ayrılmış klasör ya da özellik klasörü adı” sayfa 541. Tanımlı bir klasör adını yalnızca tanımlı amacı için kullanın; bkz. “Tanımlı klasör adı” sayfa 533.

Klasör adı etiketine ' content=properties ' özniteliğini eklerseniz, klasör bir özellik klasörü olur; bkz. Şekil 3 sayfa 532.

```
<myFolder></myFolder>
<myPropertyFolder contents=' properties '></myPropertyFolder>
```

Şekil 3. Bir klasör ve özellik klasörü örneği

Klasör adları büyük ve küçük harfe duyarlıdır. Klasör adları ve özellik klasörü adları aynı ad alanını paylaşır. Farklı isimleri olmalı. [Şekil 4 sayfa 533](#) içindeki Folder1 , [Şekil 5 sayfa 533](#) içindeki Folder2 ile farklı bir ad olmalıdır.

```
< Folder1 ><NVP1> value </NVP1></ Folder1 >
```

Şekil 4. Folder1 ad alanı

```
< Folder2 content='properties'>< Property1 > value </ Property1 ></ Folder2 >
```

Şekil 5. Folder2 ad alanı

Farklı klasörlerdeki gruplar, özellikler ve ad-değer çiftleri farklı ad alanlarına sahiptir. [Şekil 5 sayfa 533](#) içindeki Property1 , [Şekil 6 sayfa 533](#) içindeki Property1 ile farklı bir özelliktir.

```
<Folder3 content='properties'>< Property1 > value </ Property1 ></Folder3>
```

Şekil 6. Folder3 ad alanı

Özellik klasörleri, iki önemli açıdan özellik dışı klasörlerden farklıdır:

1. Özellik klasörleri özellikleri içerir ve özellik olmayan klasörler ad-değer çiftlerini içerir. Klasörler, sözdizimsel olarak biraz farklıdır.
2. İleti özelliklerine erişmek için MQI ya da JMS ileti özellikleri gibi tanımlı arabirimleri kullanın. Arabirimler, MQRFH2 içindeki özellik klasörlerinin doğru biçimlendirildiğinden emin olur. Doğru biçimlendirilmiş bir özellik klasörü, farklı platformlardaki kuyruk yöneticileri ile farklı yayınlarda birlikte çalışabilir.

İleti özelliği MQI, MQRFH2'ı okumak ve yazmak için sağlam bir yoldur ve bir MQRFH2 ' nin doğru olarak ayrıştırılmasının zorluklarını önler.

Tanımlı klasör adı

Tanımlı klasör adı, IBM MQya da başka bir ürün tarafından kullanılmak üzere ayrılmış bir klasörün adıdır. Aynı ada sahip bir klasör oluşturmayın ve klasörlere kendi ad-değer çiftlerinizi eklemeyin. Tanımlı klasörler: psc ve pscı.

psc ve pscı , kuyruğa alınan yayınlama/abone olma tarafından kullanılır.

MQMF_SEGMENT ya da MQMF_SEGMENTATION_ALLOWED ile yerleştirilen bölümlenmiş bir ileti, tanımlı klasör adına sahip bir MQRFH2 içeremez. MQPUT , 2443, MQRC_SEGMENTATION_NOT_ALLOWEDneden koduyla başarısız olur.

Tanımlı özellik klasörü adı

Tanımlı özellik klasörü adı, IBM MQya da başka bir ürün tarafından kullanılan bir özellik klasörünün adıdır. Klasörlerin ve içeriklerinin adları için bkz. [Özellik klasörleri](#). Tanımlı özellik klasörü adları, IBM MQtarafından ayrılmış tüm klasör adlarının bir alt kümesidir; bkz. [“Ayrılmış klasör ya da özellik klasörü adı” sayfa 541](#).

Tanımlı bir özellik klasöründe saklanan her öge bir özelliktir. Tanımlı bir özellik klasöründe saklanan bir ögenin content=' properties ' özniteliği olmamalıdır.

Özellikleri yalnızca tanımlı özellik klasörlerine `usr`, `mq_usrve` `sib_usr` ekleyebilirsiniz. `mq` ve `sib` gibi diğer özellik klasörlerinde IBM MQ , tanımadığı özellikleri yoksayar ya da atar.

Tanımlanan her özellik klasörünün tanımı, uygulama programları tarafından kullanılacak IBM MQ ' in tanımladığı özellikleri listeler. Bazı özelliklere bir JMS özelliği ayarlanarak ya da alınırken dolaylı olarak erişilir ve bazılarında MQSETMP ve MQINQMP MQI çağruları kullanılarak doğrudan erişilir.

Tanımlanan özellik klasörleri, IBM MQ ' in ayırdığı, ancak hangi uygulamalara erişemeyen diğer özellikleri de içerir. Ayrılmış özelliklerin adları listelenmez. `usr`, `mq_usrve` `sib_usr` özellik klasörlerinde ayrılmış bir özellik yoktur. Ancak, özellik adları geçersiz olan özellikler oluşturmayın; bkz. [“Özellik adı geçersiz” sayfa 542.](#)

Özellik klasörleri

jms

`jms` , MQMDiçinde tam olarak ifade edilemeyen JMS üstbilgi alanlarını ve JMSX özelliklerini içerir. `jms` klasörü her zaman bir JMS MQRFH2içinde bulunur.

Çizelge 515. <i>jms</i> özellik adı, eşanlamlı, veri tipi ve klasör			
Özellik eşanlamlısı	Özellik adı	Veri tipi	Klasör
JMSDestination (JMS Hedefi)	<code>jms.Dst</code>	string	<code><jms><Dst> destination </Dst></jms></code>
JMSExpiration	<code>jms.Exp</code>	i8	<code><jms><Exp> expiration </Exp></jms></code>
JMSCorrelation	<code>jms.Cid</code>	string	<code><jms><Cid> correlationId </Cid></jms></code>
JMSDelivery	<code>jms.Dlv</code>	i4	<code><jms><Dlv> delivery </Dlv></jms></code>
JMSPriority (JMSPriority)	<code>jms.Pri</code>	i4	<code><jms><Pri> priority </Pri></jms></code>
JMSReplyTo	<code>jms.Rto</code>	string	<code><jms><Rto> replyToURI </Rto></jms></code>
JMSTimestamp	<code>jms.Tms</code>	i8	<code><jms><Tms> timestamp </Tms></jms></code>
JMSXGroupID	<code>jms.Gid</code>	string	<code><jms><Gid> groupId </Gid></jms></code>
JMSXGroupSeq	<code>jms.Seq</code>	i4	<code><jms><Seq> messageSequenceNo </Seq></jms></code>

`jms` klasörüne kendi özelliklerinizi eklemeyin.

mcd

`mcd` , iletinin biçimini açıklayan özellikleri içerir. Örneğin, ileti hizmeti etki alanı `Msd` özelliği bir JMS iletisini `JMSTextMessage`, `JMSByteMessage`, `JMSStreamMessage`, `JMSMapMessage`, `JMSObjectMessage`ya da boş değer olarak tanımlar.

`mcd` klasörü, MQRFH2içeren bir JMS iletisinde her zaman bulunur.

Bu ileti, her bana yılında IBM Integration Bus' dan gönderilen bir MQRFH2 iletisini içeren iletide bulunur. Bir iletinin etki alanını, biçimini, tipini ve ileti kümesini açıklar.

Çizelge 516. mcd özellik adı, eşanlamlı, veri tipi ve klasör			
Özellik eşanlamlısı	Özellik adı	Veri tipi	Klasör
	mcd.Msd	string	<mcd><Msd>messageDomain</Msd></mcd>
	mcd.Set	string	<mcd><Set>messageDomain</Set></mcd>
	mcd.Type	string	<mcd><Type>messageDomain</Type></mcd>
	mcd.Fmt	string	<mcd><Fmt>messageDomain</Fmt></mcd>

mcd klasörüne kendi özelliklerinizi eklemeyin.

mq_usr

mq_usr , JMS kullanıcı tanımlı özellikleri olarak gösterilmeyen uygulama tanımlı özellikleri içerir. JMS gereksinimlerini karşılamayan özellikler bu klasöre yerleştirilebilir.

mq_usr klasöründe özellikler oluşturabilirsiniz. mq_usr içinde oluşturduğunuz özellikler, content='properties' özniteliğiyle yeni klasörlerde oluşturduğunuz özellikler gibidir.

sib

sib , WebSphere Application Server hizmet bütünleştirme veriyolu (WAS/SIB) sistem iletisi özelliklerini içerir. sib özellikleri, desteklenen tiplerde olmadığından IBM MQ JMS uygulamalarına JMS özellikleri olarak gösterilmez. Örneğin, bazı sib özellikleri bayt dizileri olduğundan JMS özellikleri olarak gösterilemez. Bazı sib özellikleri WAS/SIB uygulamalarına JMS_IBM_* özellikleri olarak gösterilir; bunlar ileriye doğru ve ters yöneltme yolları özelliklerini içerir.

sib klasörüne kendi özelliklerinizi eklemeyin.

sib_context

sib_context , WAS/SIB kullanıcı uygulamalarına ya da JMS özellikleri olarak gösterilmeyen WAS/SIB sistem iletisi özelliklerini içerir. sib_context , web hizmetleri için kullanılan güvenlik ve hareket özelliklerini içerir.

sib_context klasörüne kendi özelliklerinizi eklemeyin.

sib_usr

sib_usr , desteklenen tipte olmadığından JMS kullanıcı özellikleri olarak gösterilmeyen WAS/SIB kullanıcı iletisi özelliklerini içerir. sib_usr , SImessage arabiriminde WAS/SIB uygulamalarına gösterilir; bkz. [Developing Service Integration](#).

sib_usr özelliğinin tipi bin . hex olmalı ve değer doğru biçimde olmalıdır. Bir IBM MQ uygulaması bin . hex tipinde bir öğeyi klasöre yanlış biçimde yazarsa, uygulama bir IOException alır. Özelliğin veri tipi bin . hex değilse, uygulama bir ClassCastException alır.

Bu klasörü kullanarak JMS kullanıcı özelliklerini WAS/SIB için kullanılabilir kılmayı denemeyin; bunun yerine usr klasörünü kullanın.

sib_usr klasöründe özellikler oluşturabilirsiniz.

usr

usr , iletiyle ilişkili uygulama tanımlı JMS özelliklerini içerir. usr klasörü, yalnızca bir uygulama uygulama tanımlı bir özellik ayarladıysa bulunur.

usr varsayılan özellik klasörüdür. Bir özellik klasör adı olmadan ayarlanırsa, usr klasörüne yerleştirilir.

Çizelge 517. usr özellik adı, eşanlamlı, veri tipi ve klasör			
Özellik eşanlamlısı	Özellik adı	Veri tipi	Klasör
	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>

usr klasöründe özellikler oluşturabilirsiniz.

MQMF_SEGMENT ya da MQMF_SEGMENTATION_ALLOWED içeren bölümlenmiş bir ileti, tanımlı bir özellik klasörü adına sahip bir MQRFH2 içeremez. MQPUT , 2443, MQRC_SEGMENTATION_NOT_ALLOWEDneden koduyla başarısız olur.

Gruplanmamış özellik klasörü adı

ibm

ibm , yalnızca IBM MQtarafından kullanılan özellikleri içerir.

Çizelge 518. ibm özellik adı, eşanlamlı, veri tipi ve klasör			
Özellik eşanlamlısı	Özellik adı	Veri tipi	Klasör
	ibm.rfp	string	<ibm><rfp>fingerprint</rfp></ibm>

ibm klasörüne kendi özelliklerinizi eklemeyin.

mq

mq , yalnızca IBM MQtarafından kullanılan özellikleri içerir.

mq klasöründeki özellikler için aşağıdaki kısıtlamalar geçerlidir:

- Yalnızca iletideki ilk önemli mq klasöründeki özellikler MQtarafından işlendi; iletideki başka bir mq klasöründeki özellikler yoksayılır.
- Klasörde yalnızca tek baytlık UTF-8 karakterlerine izin verilir. Klasördeki çok baytlı bir karakter, ayrıştırmanın başarısız olmasına ve iletinin reddedilmesine neden olabilir.
- Klasörde çıkış dizgilerini kullanmayın. Bir çıkış dizgisi, ögenin gerçek değeri olarak kabul edilir.
- Klasör içinde yalnızca Unicode karakteri U+0020 beyaz alan olarak kabul edilir. Diğer tüm karakterler önemli olarak değerlendirilir ve klasör ayrıştırmasının başarısız olmasına ve iletinin reddedilmesine neden olabilir.

mq klasörünün ayrıştırılması başarısız olursa ya da klasör bu kısıtlamalara uymazsa, ileti 2527, MQRC_RFH_RESTRICTED_FORMAT_ERRneden koduyla reddedilir.

mq klasörüne kendi özelliklerinizi eklemeyin.

mqema

mqema , yalnızca WebSphere Application Servetarafından kullanılan özellikleri içerir. Klasör mqextile değiştirildi.

mqema klasörüne kendi özelliklerinizi eklemeyin.

mqext

mqext aşağıdaki özellik tiplerini içerir:

- Yalnızca WebSphere Application Servetarafından kullanılan özellikler.
- İletilerin gecikmeli teslimiyle ilgili özellikler.

Uygulama, IBM tarafından tanımlanan özelliklerden en az birini ya da kullanılan teslim gecikmesini ayarladıysa klasör vardır.

Çizelge 519. mqext özellik adı, eşanlamlı, veri tipi ve klasör			
Özellik eşanlamlısı	Özellik adı	Veri tipi	Klasör
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>

mqext klasörüne kendi özelliklerinizi eklemeyin.

mqps

mqps , yalnızca IBM MQ yayınlama/abone olma tarafından kullanılan özellikleri içerir. Klasör, yalnızca uygulama, tümleşik yayınlama/abone olma özelliklerinden en az birini ayarladıysa bulunur.

Çizelge 520. mqps özellik adı, eşanlamlı, veri tipi ve klasör			
Özellik eşanlamlısı	Özellik adı	Veri tipi	Klasör
MQTopicString	mqps.Top	string	<mqps><Top>topicString</Top></mqps>
MQSubscriberleri	mqps.Sud	string	<mqps><Sud>subscriberUserData...</Sud></mqps>
MQIsRetained	mqps.Ret	boolean	<mqps><Ret>isRetained</Ret></mqps>
MQPublicationOptions	mqps.Pub	i8	<mqps><Pub>publicationOptions</Pub></mqps>
MQPublicationLevel	mqps.Pbl	i8	<mqps><Pbl>publicationLevel</Pbl></mqps>
MQPublicationTime	mqpse.Pts	string	<mqps><Pts>publicationTime</Pts></mqps>
MQPublicationSequenceNumber	mqpse.Seq	i8	<mqps><Seq>publicationSequenceNumber</Seq></mqps>
MQPublicationData	mqpse.Sid	string	<mqps><Sid>publicationData</Sid></mqps>

Çizelge 520. mqps özellik adı, eşanlamlı, veri tipi ve klasör (devamı var)			
Özellik eşanlamlısı	Özellik adı	Veri tipi	Klasör
MQPubFormat	mqpse.Pfmt	i8	<mqps><Pfmt>messageFormat</Pfmt></mqps>

mqps klasörüne kendi özelliklerinizi eklemeyin.

mq_svc

mq_svc , SupportPac MA93tarafından kullanılan özellikleri içerir.

mq_svc klasörüne kendi özelliklerinizi eklemeyin.

mqtt

mqtt , MQ Telemetry tarafından kullanılan özellikleri içerir

Çizelge 521. mqtt özellik adı, eşanlamlı, veri tipi ve klasör			
Özellik eşanlamlısı	Özellik adı	Veri tipi	Klasör
	mqtt.clientId	string	<mqtt><clientId> topicString </clientId></mqtt>
	mqtt.qos	i4	<mqtt><qos> qualityOfService </qos></mqtt>
	mqtt.msgid	string	<mqtt><msgid> messageIdentifier </msgid></mqtt>

mqtt klasörüne kendi özelliklerinizi eklemeyin.

MQMF_SEGMENT ya da MQMF_SEGMENTATION_ALLOWED ile yerleştirilen bölümlenmiş bir ileti, gruplanmamış özellik klasörü adına sahip bir MQRFH2 içeremez. MQPUT , 2443, MQRC_SEGMENTATION_NOT_ALLOWEDneden koduyla başarısız olur.

Ad-değer çiftleri

Sözdizimi şemasında, "Ad-değer çiftleri" olağan bir klasörün içeriğini açıklar. Sıradan bir klasör grupları ve öğeleri içerir. Öğeler bir ad-değer çiftidir. Bir grup öğeleri ve diğer grupları içerir.

Ağaçlar açısından, öğeler yaprak düğümlerdir ve gruplar iç düğümlerdir. Bir iç düğüm ve kök düğüm olan klasör, iç düğümler ve yaprak düğümlerin bir karışımını içerebilir. Bir düğüm aynı anda hem bir iç düğüm hem de bir yaprak düğüm olamaz; bkz. [Şekil 2 sayfa 532](#).

Özellikler

Sözdizimi şemasında "Özellikler" seçeneği, bir özellik klasörünün içeriğini açıklar. Bir özellik klasörü grupları ve özellikleri içerir. Özellik, isteğe bağlı veri tipi özniteliği olan bir ad-değer çiftidir. Bir grup özellikleri ve diğer grupları içerir.

Ağaçlar açısından, özellikler yaprak düğümlerdir ve gruplar iç düğümlerdir. Bir iç düğüm ve kök düğüm olan özellik klasörü, iç düğümler ve yaprak düğümlerin bir karışımını içerebilir. Bir düğüm aynı anda hem bir iç düğüm hem de bir yaprak düğüm olamaz; bkz. [Şekil 2 sayfa 532](#).

Özellik

İleti özelliği, bir özellik klasöründeki ad-değer çiftidir. İsteğe bağlı olarak bir veri tipi özniteliği ve bir özellik özniteliği içerebilir; örneğin, aşağıdaki koda bakın. Veri tipi özniteliği atlanırsa, özellik tipi string olur.

```
<pf><p1 dt='i8' > value </p1></pf>
```

Bir ileti özelliğinin adı, XML benzeri <> sözdizimiyle tam yol adıdır ve nokta ile değiştirilir. Örneğin, myPropertyFolder1.myGroup1.myGroup2.myProperty1 aşağıdaki gibi bir *NameValueData* dizisiyle eşlenir. Dizgi, daha kolay okunmak için biçimlendirilir.

```
<myPropertyFolder1>
  <myGroup1>
    <myGroup2>
      <myProperty1>value</myProperty1>
    </myGroup2>
  </myGroup1>
</myPropertyFolder1>
```

Bir özellik klasörü birden çok özellik içerebilir. Örneğin, Şekil 7 sayfa 539 içindeki özellikler Yeni bir kümedeki satış kuyruğunu yalıtım ve ağ geçidi kümesi iletim kuyruklarını ayırmak için yapılan değişikliklerindeki özellik klasörüyle eşlenir.

```
myPropertyFolder1.myProperty4
myPropertyFolder1.myGroup1.myGroup2.myProperty1
myPropertyFolder1.myGroup1.myGroup2.myProperty2
myPropertyFolder1.myGroup1.myProperty3
```

Şekil 7. Aynı kök ada sahip birden çok özellik

```
<myPropertyFolder1>
  <myProperty4>value</myProperty4>
  <myGroup1>
    <myGroup2>
      <myProperty1>value</myProperty1>
      <myProperty2>value</myProperty2>
    </myGroup2>
    <myProperty3>value</myProperty3>
  </myGroup1>
</myPropertyFolder1>
```

Şekil 8. Birden çok özellik adı eşlemesi

Ad

Bir ad *Harf* ya da *Altçizgi*le başlamalıdır. *İki nokta* içermemelidir, *Nokta* ile bitmemelidir ve yalnızca *Harfler*, *Sayılar*, *Alt çizgiler*, *Kısa çizgiler* ve *Nokta* içermelidir. Geçerli karakterler W3C XML belirtiminde tanımlanır ve esasen Unicode kategorilerinden oluşur L1, Lu, Lo, Lt, N1, Mc, Mn, Lm, ve Nd ; Bkz. [Unicode karakter kategorileri](#).

Bir özelliğin ya da ad-değer çiftinin tam yolu, “Yol adı geçersiz” sayfa 542 içinde açıklanan kuralı bozmamalıdır. Yollar 4095 byte ile sınırlıdır, Unicode uyumluluk karakterleri içermemelidir ve XML dizisiyle başlamamalıdır.

Grup adı

Bir grup adı, bir adla aynı sözdizimine sahip. Grup adları isteğe bağlıdır. Özellikler ve ad-değer çiftleri bir klasörün köküne yerleştirilebilir. Özellikleri ve ad-değer çiftlerini düzenlemeye yardımcı olacaksa grupları kullanın.

Öğe adı

Bir öğe adı, adla aynı sözdizimine sahip.

Öge değeri

Bir öge değeri, < *Element name* > etiketi ile < /*Element name* > arasındaki tüm beyaz alanı içerir. Bir değerde < ve & karakterlerini kullanmayın. < ve & ile değiştirin.

Özellik öznitelikleri

Özellik öznitelikleri özellik tanımlayıcı alanlarını eşler: Eşlemeler aşağıdaki gibidir:

Destek

sa (varsayılan)

MQPD_SUPPORT_OPTIONAL

sr

MQPD_SUPPORT_REQUIRED

sx

MQPD_SUPPORT_REQUIRED_IF_LOCAL

Bağlam

none (varsayılan)

MQPD_NO_CONTEXT

kullanıcı

MQPD_USER_CONTEXT

CopyOptions

ileri

MQPD_COPY_FORWARD

Yanıt

MQPD_COPY_REPLY

rapor

MQPD_COPY_REPORT

yayınlamak

MQPD_COPY_PUBLISH

tümü

MQPD_COPY_ALL

Diğer seçeneklerle birlikte tümünü kullanmayın.

varsayılan

MQPD_COPY_DEFAULT

Diğer seçeneklerle birlikte default kullanmayın. default , forward + report + publish ile aynıdır.

yok

MQPD_COPY_NONE

Diğer seçeneklerle birlikte none kullanmayın.

Destek özellik öznitelikleri yalnızca mq klasöründeki özellikler için geçerlidir.

Bağlam ve CopyOptions özellik öznitelikleri tüm özellik klasörleri için geçerlidir.

Veri türleri

MQRFH2 veri tipleri, aşağıdaki gibi ileti özelliği tipleriyle eşlenir:

Çizelge 522. Veri tipi eşlemeleri

MQRFH2 veri tipi	İleti özelliği tipi
bin.hex	MQBYTE []
boolean	MQBOOL
i1	MQINT8
i2	MQINT16
i4	MQINT32
i8	MQINT64
r4	MQFLOAT32
r8	MQFLOAT64
string	MQCHAR []

Veri tipi olmayan herhangi bir öğenin stringtipinde olduğu varsayılır.

`xsi:nil='true'` öge özniteliğiyle boş değer gösterilir. Boş olmayan değerler için `xsi:nil='false'` özniteliğini kullanmayın. Örneğin, aşağıdaki özelliğin boş değeri vardır:

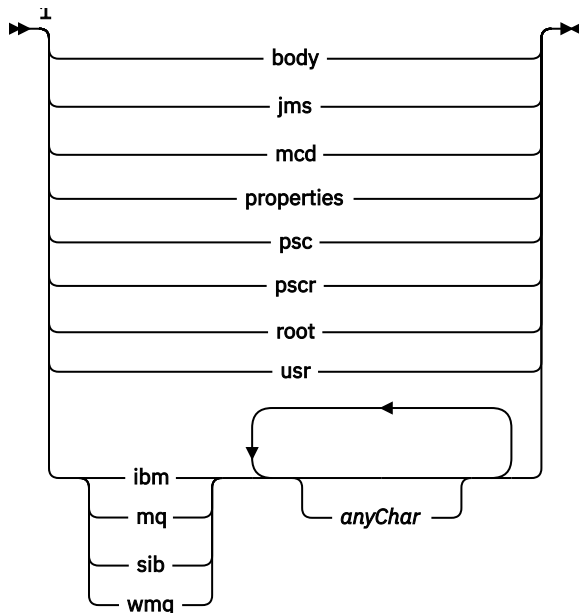
```
<NullProperty  
xsi:nil='true'></NullProperty>
```

Bir byte ya da karakter dizisi özelliğinin değeri boş olabilir. Boş bir değer, sıfır uzunluklu öge değeri olan bir MQRFH2 öğesiyle gösterilir. Örneğin, aşağıdaki özelliğin boş bir değeri vardır:

```
<EmptyProperty></EmptyProperty>
```

Ayrılmış klasör ya da özellik klasörü adı

Bir klasörün ya da özellik klasörünün adını, aşağıdaki dizgilerden biriyle başlamayacak şekilde kısıtlayın. Önekler, IBM tarafından yaratılan klasör ya da özellik adları için ayrılır.

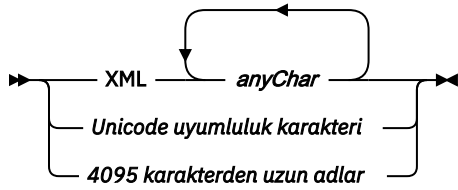


Notlar:

¹ Ayrılmış bir klasör ya da özellik adı, küçük ve büyük harflerin karışımını içerir.

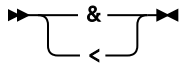
Yol adı geçersiz

Bir ad-değer çiftinin ya da bir özelliğin tam yolunu, aşağıdaki dizgilerden herhangi birini içermeyecek şekilde kısıtlayın.



Geçersiz karakterler

"&" ve "<" hazır bilgileri yerine her zaman & ; ve < çıkış dizilerini kullanın.

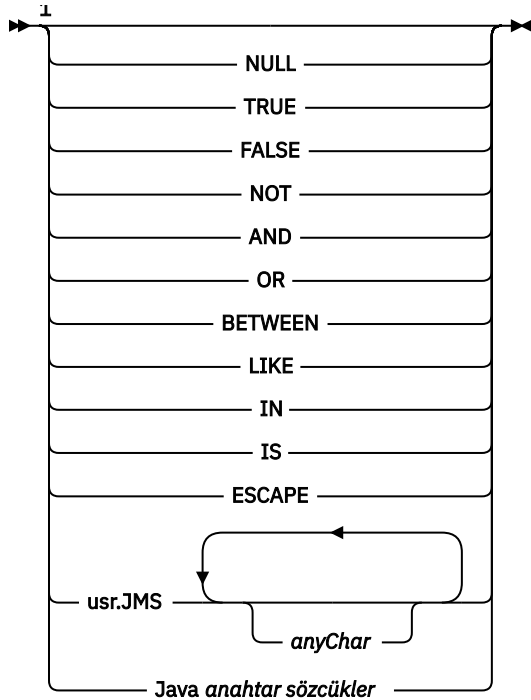


Tanımlı özellik adları

Tanımlı özellik adları, IBM MQ'ya da diğer ürünler tarafından tanımlanan ve IBM MQ ve kullanıcı uygulamaları tarafından kullanılan özelliklerin adlarıdır. Tanımlı özellikler yalnızca tanımlı özellik klasörlerinde var. Tanımlanan özellik adları, özellik klasörlerinin tanımında açıklanır; bkz. [Özellik klasörleri](#).

Özellik adı geçersiz

Aşağıdaki kuralla eşleşen özellik adları oluşturmayın. Kural, yalnızca özellik ögesi adı için değil, bir özelliği adlandıran tam özellik yolu için de geçerlidir.



Notlar:

¹ Geçersiz bir özellik adı, herhangi bir büyük ve küçük harf birleşimi içerebilir.

Geçersiz öznitelikler

Özellik klasörleri ve özellikleri yalnızca desteklenen “Özellik öznitelikleri” sayfa 540 ve “Veri türleri” sayfa 540 içerebilir.

Özellik klasörlerinde ya da özelliklerinde bulunan, desteklenmeyen XML benzeri öznitelikler (örneğin, tırnak içine alınmış dizgi değerleri olan adlar) kaldırılabilir.

Özellik olmayan klasörlerde ya da özellik olmayan öğelerde bulunan ve MQRFH2 üstbilgilerinde kalan XML benzeri öznitelikler.

MQRMH-Başvuru iletisi üstbilgisi

MQRMH yapısı, başvuru iletisi üstbilgisinin biçimini tanımlar. Bu üstbilgi, çok büyük miktarlarda veri (*toplu veri* olarak adlandırılır) göndermek için kullanıcı tarafından yazılan ileti kanalı çıkışlarıyla birlikte kullanılır. bir kuyruk yöneticisinden diğerine. Normal ileti sistemiyle karşılaştırıldığında, toplu veriler bir kuyrukta saklanmaz; bunun yerine, kuyrukta yalnızca toplu verilere ilişkin bir *başvuru* saklanır. Bu, IBM MQ kaynaklarının çok az sayıda çok büyük ileti tarafından tüketilebilme olasılığını azaltır.

Kullanılabilirlik

MQRMH yapısı aşağıdaki platformlarda kullanılabilir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

Biçim Adı

MQFMT_REF_MSG_HEADER

Karakter kümesi ve kodlama

MQRMH ' deki karakter verileri ve görelî konum alanları tarafından adreslenen dizgiler, yerel kuyruk yöneticisinin karakter kümesinde olmalıdır; bu, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından verilir. MQRMH ' deki sayısal veriler yerli makine kodlamasında olmalıdır; bu, C programlama dili için MQENC_NATIVE değeriyle verilir.

MQRMH ' nin karakter takımını ve kodlamasını aşağıdaki *CodedCharSetId* ve *Encoding* alanlarına ayarlayın:

- MQMD (MQRMH yapısı ileti verilerinin başındadır) ya da
- MQRMH yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Kullanım

Bir uygulama, MQRMH ' den oluşan bir ileti koyar, ancak toplu verileri atlar. Bir ileti kanalı aracı (MCA) iletiyi iletim kuyruğundan okuduğunda, başvuru iletisi üstbilgisini işlemek için kullanıcı tarafından sağlanan bir ileti çıkışı çağırılır. Çıkış, MCA iletiyi kanal üzerinden sonraki kuyruk yöneticisine göndermeden önce, MQRMH yapısıyla tanımlanan toplu verilerin başvuru iletisinin sonuna eklenebilir.

Alma sonunda, başvuru iletisini bekleyen bir ileti çıkışı var olmalıdır. Bir başvuru iletisi alındığında, çıkışın nesneyi iletideki MQRMH ' yi izleyen toplu verilerden yaratması ve daha sonra, başvuru iletisini toplu veriler olmadan aktarması gerekir. Başvuru iletisi, daha sonra bir kuyruktan başvuru iletisini (toplu veri olmadan) okuyan bir uygulama tarafından alınabilir.

Normalde, MQRMH yapısı iletideki tek yapıdır. Ancak, ileti bir iletim kuyruğuyorsa, MQRMH yapısından önce bir ya da daha çok ek üstbilgi gelir.

Dağıtım listesine bir başvuru iletisi de gönderilebilir. Bu durumda, ileti bir iletim kuyruğunda olduğunda MQDH yapısı ve ilgili kayıtları MQRMH yapısından önce gelir.

Not: Bir başvuru iletisini bölümlenmiş ileti olarak göndermeyin; ileti çıkışı iletiyi doğru olarak işleyemez.

Veri dönüştürme

Veri dönüştürme amacıyla, MQRMH yapısının dönüştürülmesi, kaynak ortam verilerinin, kaynak nesne adının, hedef ortam verilerinin ve hedef nesne adının dönüştürülmesini içerir. Yapının başlangıcını izleyen *StrucLength* bayt içindeki diğer tüm baytlar atılır ya da veri dönüştürmeden sonra tanımlanmamış değerlere sahip olur. Aşağıdaki tüm deyimlerin doğru olması koşuluyla toplu veriler dönüştürülür:

- Veri dönüştürme işlemi gerçekleştirildiğinde iletide toplu veri bulunur.
- MQRMH ' deki *Format* alanında MQFMT_NONE dışında bir değer var.
- Belirtilen biçim adına sahip, kullanıcı tarafından yazılan bir veri dönüştürme çıkışı var.

Ancak, ileti kuyruktayken genellikle toplu verilerin iletide bulunmadığını ve sonuç olarak toplu verilerin MQGMO_CONVERT seçeneğiyle dönüştürüldüğünü unutmayın.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 523. MQRMH için MQRMH ' deki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQRMH_STRUC_ID	'RMH→'
<u>Sürüm</u> (yapı sürümü numarası)	MQRMH_VERSION_1	1
<u>StrucLength</u> (toplu veriler değil, sabit alanların sonundaki dizgiler de içinde olmak üzere toplam MQRMH uzunluğu)	Yok	0
<u>Kodlama</u> (toplu verilerin sayısal kodlaması)	MQENC_NATIVE	Ortama bağlıdır
<u>CodedCharSetId</u> (toplu verilerin karakter kümesi tanıtıcısı)	MQCCSI_AYRILMAMIŞ	0
<u>Biçim</u> (toplu verilerin biçim adı)	MQFMT_NONE	Boşluklar
<u>İşaretler</u> (başvuru iletisi işaretleri)	MQRMHF_NOT_LAST	0
<u>ObjectType</u> (nesne tipi)	Yok	Boşluklar
<u>ObjectInstanceId</u> (nesne eşgörünümü tanıtıcısı)	MQII_NONE	Boş Değerler
<u>SrcEnvLength</u> (kaynak ortam verilerinin uzunluğu)	Yok	0
<u>SrcEnvOffset</u> (kaynak ortam verilerinin görelî konumu)	Yok	0
<u>SrcNameLength</u> (kaynak nesne adının uzunluğu)	Yok	0
<u>SrcNameGörelî Konumu</u> (kaynak nesne adının görelî konumu)	Yok	0
<u>DestEnvLength</u> (hedef ortam verilerinin uzunluğu)	Yok	0
<u>DestEnvGörelî Konum</u> (Hedef Ortam Verilerinin Görelî Konumu)	Yok	0

Çizelge 523. MQRMH için MQRMH 'deki Alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>DestNameLength</u> (hedef nesne adının uzunluğu)	Yok	0
<u>DestNameGörelili Konumu</u> (hedef nesne adının görelili konumu)	Yok	0
<u>DataLogicalLength</u> (Toplu veri uzunluğu)	Yok	0
<u>DataLogicalOffset</u> (Toplu verilerin düşük görelili konumu)	Yok	0
<u>DataLogicalOffset2</u> (toplu verilerin yüksek görelili konumu)	Yok	0

Notlar:

1. - simgesi, tek bir boş karakteri temsil eder.
2. C programlama dilinde, makro değişkeniMQRMH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQRMH MyRMH = {MQRMH_DEFAULT};
```

Dil bildirimleri

MQRMH için C bildirimini

```
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 */
};
```

MQRMH için COBOL bildirimini

```
** 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
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-DATALOGICALENGTH 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.

```

MQRMH için PL/I bildirimi

```

dcl
  1 MQRMH based,
  3 StructId          char(4),          /* Structure identifier */
  3 Version          fixed bin(31), /* Structure version number */
  3 StructLength     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 */

```

MQRMH için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

MQRMH          DSECT
MQRMH_STRUCID DS CL4 Structure identifier

```

MQRMH_VERSION	DS	F	Structure version number
MQRMH_STRUCLength	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_DATALOGICALENGTH	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)	

MQRMH için Visual Basic bildirimi

```

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

```

MQRMH için StrucId (MQCHAR4)

Bu, başvuru iletisi üstbilgi yapısının yapı tanıtcısıdır. Her zaman bir giriş alanıdır. Değeri MQRMH_STRUC_ID'dir.

Değer şöyle olmalıdır:

MQRMH_STRUC_ID

Başvuru iletisi üstbilgi yapısına ilişkin tanıtcı.

C programlama dili için, MQRMH_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQRMH_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQRMH için Sürüm (MQLONG)

Yapı sürümü numarası. Değer şöyle olmalıdır:

MQRMH_VERSION_1

Version-1 başvuru iletisi üstbilgi yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQRMH_CURRENT_VERSION

Başvuru iletisi üstbilgi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri MQRMH_VERSION_1' dir.

MQRMH için StrucLength (MQLONG)

Değişmez alanların sonundaki dizgiler de içinde olmak üzere toplam MQRMH uzunluğu, toplu veriler değil.

Bu alanın ilk değeri sıfırdır.

MQRMH için kodlama (MQLONG)

Bu, toplu verilerin sayısal kodlamasını belirtir; MQRMH yapısındaki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri MQENC_NATIVE.

MQRMH için CodedCharSetId (MQLONG)

Bu, toplu verilerin karakter kümesi tanıtıcısını belirler; MQRMH yapısındaki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değer kullanılabilir:

MQCCSI_INHERIT

Bu yapıyı izleyen verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı MQCCSI_INHERIT değerini döndürmez.

MQMD 'deki PutApp1Type alanının değeri MQAT_BROKER ise, MQCSI_INHERIT kullanmayın.

Bu değer aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

Bu alanın ilk değeri MQCCSI_UNDEFINED 'dir.

MQRMH için biçim (MQCHAR8)

Bu, toplu verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD 'deki *Format* alanıyla aynıdır.

Bu alanın ilk değeri MQFMT_NONE.

MQRMH için İşaretler (MQLONG)

Bunlar başvuru iletisi işaretleridir. Şu işaretler tanımlandı:

MQRMHF_SON

Bu işaret, başvuru iletisinin başvurulmuş nesnenin son kısmını gösterdiğini ya da içerdiğini gösterir.

MQRMHF_NOT_LAST

Başvuru iletisi, nesnenin son bölümünü içermiyor ya da temsil etmiyor. MQRMHF_NOT_LAST yardımcı program belgeleri. Bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu alanın ilk değeri MQRMHF_NOT_LAST 'dir.

MQRMH için ObjectType (MQCHAR8)

Bu, ileti çıkışının desteklediği başvuru iletisi tiplerini tanımak için kullanılabileceği bir addır. Ad, *Format* alanıyla aynı kurallara uymalıdır, bkz. [“MQRMH için biçim \(MQCHAR8\)” sayfa 548.](#)

Bu alanın ilk değeri 8 boşluktur.

ObjectInstanceId (MQBYTE24)-MQRMH için

Bir nesnenin belirli bir örneğini tanımlamak için bu alanı kullanın. Gerekli değilse, şu değere ayarlayın:

MQII_NONE

Nesne eşgörünümü tanıtıcısı belirtilmedi. Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, değişmez MQOII_NONE_ARRAY da tanımlandı; bu değer MQOII_NONE ile aynı, ancak bir dizgi yerine bir karakter dizisi.

Bu alanın uzunluğu MQ_OBJECT_INSTANCE_ID_LENGTH tarafından verilir. Bu alanın ilk değeri MQOII_NONE.

SrcEnvMQRMH için uzunluk (MQLONG)

Kaynak ortam verilerinin uzunluğu. Bu alan sıfırsa, kaynak ortam verileri yoktur ve *SrcEnvOffset* yoksayıdır.

Bu alanın ilk değeri 0 'dir.

SrcEnvGörelî Konumu (MQLONG)

Bu alan, MQRMH yapısının başlangıcından itibaren kaynak ortam verilerinin görelî konumunu belirler. Kaynak ortam verileri, yaratan tarafından biliniyorsa, başvuru iletisini oluşturan tarafından belirtilebilir. Örneğin, Windows üzerinde kaynak ortam verileri, toplu verileri içeren nesnenin dizin yolu olabilir. Ancak, yaratan kaynak ortam verilerini bilmiyorsa, kullanıcı tarafından sağlanan ileti çıkışı gereken ortam bilgilerini belirlemelidir.

Kaynak ortam verilerinin uzunluğu *SrcEnvLength* ; Bu uzunluk sıfır ise, kaynak ortam verileri yoktur ve *SrcEnvOffset* yoksayıdır. Varsa, kaynak ortam verileri, yapının başlangıcından itibaren *StrucLength* bayt içinde tam olarak bulunmalıdır.

Uygulamalar, ortam verilerinin yapıdaki son sabit alandan hemen sonra başladığını ya da *SrcNameOffset*, *DestEnvOffset* ve *DestNameOffset* alanları tarafından adreslenen herhangi bir veriyle bitişik olduğunu varsaymamalıdır.

Bu alanın ilk değeri 0 'dir.

SrcNameMQRMH İçin Uzunluk (MQLONG)

Kaynak nesne adının uzunluğu. Bu alan sıfırsa, kaynak nesne adı yoktur ve *SrcNameOffset* yoksayıdır.

Bu alanın ilk değeri 0 'dir.

MQRMH için SrcNamegörelî konumu (MQLONG)

Bu alan, kaynak nesne adının MQRMH yapısının başlangıcından başlayarak görelî konumunu belirler. Kaynak nesne adı, verileri yaratan tarafından biliniyorsa, başvuru iletisini oluşturan tarafından belirtilebilir. Ancak, yaratan kaynak nesne adını bilmiyorsa, kullanıcı tarafından sağlanan ileti çıkışı, erişilecek nesneyi tanımlamalıdır.

Kaynak nesne adının uzunluğu *SrcNameLength* ; Bu uzunluk sıfır ise, kaynak nesne adı yoktur ve *SrcNameOffset* yoksayılr. Varsa, kaynak nesne adı, yapının başlangıcından itibaren *StrucLength* bayt içinde tam olarak bulunmalıdır.

Uygulamalar, kaynak nesne adının *SrcEnvOffset*, *DestEnvOffset* ve *DestNameOffset* alanları tarafından adreslenen herhangi bir veriyle bitişik olduğunu varsaymamalıdır.

Bu alanın ilk değeri 0 'dır.

DestEnvMQRMH İçin Uzunluk (MQLONG)

Bu, hedef ortam verilerinin uzunluğudur. Bu alan sıfırsa, hedef ortam verileri yoktur ve *DestEnvOffset* yoksayılr.

MQRMH için DestEnvgörelili konumu (MQLONG)

Bu alan, MQRMH yapısının başlangıcından itibaren hedef ortam verilerinin görelili konumunu belirler. Hedef ortam verileri, yaratan tarafından biliniyorsa, başvuru iletisini oluşturan tarafından belirtilebilir. Örneğin, Windows üzerinde hedef ortam verileri, toplu verilerin saklanacağı nesnenin dizin yolu olabilir. Ancak, yaratan hedef ortam verilerini bilmiyorsa, gerekli ortam bilgilerini belirlemek kullanıcı tarafından sağlanan ileti çıkışının sorumluluğundadır.

Hedef ortam verilerinin uzunluğu *DestEnvLength* ; Bu uzunluk sıfır ise, hedef ortam verileri yoktur ve *DestEnvOffset* yoksayılr. Varsa, hedef ortam verileri, yapının başlangıcından itibaren *StrucLength* bayt içinde tam olarak bulunmalıdır.

Uygulamalar, hedef ortam verilerinin *SrcEnvOffset*, *SrcNameOffset* ve *DestNameOffset* alanları tarafından adreslenen herhangi bir veriyle bitişik olduğunu varsaymamalıdır.

Bu alanın ilk değeri 0 'dır.

DestNameMQRMH için Uzunluk (MQLONG)

Hedef nesne adının uzunluğu. Bu alan sıfırsa, hedef nesne adı yoktur ve *DestNameOffset* yoksayılr.

DestNameMQRMH için Görelili Konum (MQLONG)

Bu alan, hedef nesne adının MQRMH yapısının başlangıcından başlayarak görelili konumunu belirler. Hedef nesne adı, bu verileri yaratan kişi biliyorsa, başvuru iletisini oluşturan kişi tarafından belirtilebilir. Ancak, yaratan hedef nesne adını bilmiyorsa, yaratılacak ya da değiştirilecek nesneyi tanımlamak kullanıcı tarafından sağlanan ileti çıkışının sorumluluğundadır.

Hedef nesne adının uzunluğu *DestNameLength* ; Bu uzunluk sıfır ise, hedef nesne adı yoktur ve *DestNameOffset* yoksayılr. Varsa, hedef nesne adı, yapının başlangıcından itibaren *StrucLength* bayt içinde tam olarak bulunmalıdır.

Uygulamalar, hedef nesne adının *SrcEnvOffset*, *SrcNameOffset* ve *DestEnvOffset* alanları tarafından adreslenen herhangi bir veriyle bitişik olduğunu varsaymamalıdır.

Bu alanın ilk değeri 0 'dır.

DataLogicalLength (MQLONG) for MQRMH

DataLogicalLength alanı, MQRMH yapısının gönderme yaptığı toplu verilerin uzunluğunu belirtir.

Toplu veriler iletide gerçekten varsa, veriler MQRMH yapısının başlangıcından itibaren *StrucLength* baytlık bir görelili konumla başlar. Tüm iletinin uzunluğu eksi *StrucLength* , var olan toplu verilerin uzunluğunu verir.

İletide veri varsa, *DataLogicalLength* ilgili verilerin miktarını belirtir. Normal durum, *DataLogicalLength* ' in iletide bulunan verilerin uzunluğuyla aynı değere sahip olması içindir.

MQRMH yapısı, nesnedeki geri kalan verileri (belirlenen mantıksal görelili konumdan başlayarak) gösteriyorsa, toplu veriler iletide gerçekten yoksa, *DataLogicalLength* için sıfır değerini kullanabilirsiniz.

Veri yoksa, MQRMH ' nin sonu iletinin sonuna denk gelir.

Bu alanın ilk değeri 0 'dır.

DataLogicalMQRMH için Offset (MQLONG)

Bu alan, toplu veri formunun parçası olduğu nesnenin başlangıcından itibaren toplu verilerin düşük görelî konumunu belirtir. Toplu verilerin nesnenin başlangıcındaki görelî konumu *mantıksal görelî konum* olarak adlandırılır. Bu, toplu verilerin MQRMH yapısının başlangıcındaki fiziksel görelî konumu değildir; bu görelî konum *StrucLength* tarafından verilir.

Büyük nesnelerin başvuru iletileri kullanılarak gönderilmesini sağlamak için, mantıksal görelî konum iki alana bölünür ve gerçek mantıksal görelî konum şu iki alanın toplamıyla verilir:

- *DataLogicalOffset* , mantıksal görelî konum 1 000 000 000 000 'e bölündüğünde elde edilen kalanı temsil eder. Bu nedenle, 0-999 999 999 999 aralığında bir değerdir.
- *DataLogicalOffset2* , mantıksal görelî konum 1 000 000 000 000 'e bölündüğünde elde edilen sonucu gösterir. Bu nedenle, mantıksal görelî konumda var olan 1 000 000 000 000 'in tam katlarının sayısıdır. Çarpı sayısı, 0-999 999 999 999 aralığındadır.

Bu alanın ilk değeri 0 'dır.

DataLogicalOffset2 (MQLONG)-MQRMH için

Bu alan, toplu veri parçasının oluşturulduğu nesnenin başlangıcından itibaren toplu verilerin yüksek görelî konumunu belirtir. Bu değer, 0-999 999 999 999 aralığında bir değerdir. Ayrıntılar için bkz. *DataLogicalOffset* .

Bu alanın ilk değeri 0 'dır.

MQRR-Yanıt kaydı

Hedef bir dağıtım listesi olduğunda, tek bir hedef kuyruğa ilişkin açma ya da koyma işleminin sonucu olan tamamlanma kodunu ve neden kodunu almak için MQRR yapısını kullanın. MQRR, MQOPEN, MQPUT ve MQPUT1 çağrılarında ilişkin bir çıkış yapısıdır.

Kullanılabilirlik

MQRR yapısı aşağıdaki altyapılarda kullanılabilir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

Karakter kümesi ve kodlama

MQRR içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Kullanım

MQOPEN ve MQPUT çağrılarında ya da MQPUT1 çağrısında bu yapıların bir dizisini sağlayarak, bir dağıtım listesindeki tüm kuyrukların tamamlanma kodlarını ve neden kodlarını, çağrı sonucu karışık olduğunda (yani, çağrı listedeki bazı kuyruklar için başarılı olduğunda ancak diğerleri için başarısız olduğunda)

saptayabilirsiniz. Çağrıda neden kodu MQRC_MULTIPLE_REASON, kuyruk yöneticisi tarafından yanıt kayıtlarının (uygulama tarafından sağlandıysa) ayarlandığını gösteriyor.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 524. MQRR 'deki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>CompCode</u> (kuyruk için tamamlama kodu)	MQCC_OK	0
<u>Neden</u> (kuyruğa ilişkin neden kodu)	MQRC_NONE	0

Notlar:

1. C programlama dilinde, makro değişkeni MQRR_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQRR MyRR = {MQRR_DEFAULT};
```

Dil bildirimleri

MQRR için C bildirimi

```
typedef struct tagMQRR MQRR;
struct tagMQRR {
    MQLONG CompCode; /* Completion code for queue */
    MQLONG Reason; /* Reason code for queue */
};
```

MQRR için COBOL bildirimi

```
** 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.
```

MQRR için PL/I bildirimi

```
dcl
1 MQRR based,
3 CompCode fixed bin(31), /* Completion code for queue */
3 Reason fixed bin(31); /* Reason code for queue */
```

MQRR için Visual Basic bildirimi

```
Type MQRR
CompCode As Long 'Completion code for queue'
Reason As Long 'Reason code for queue'
End Type
```

MQRR için CompCode (MQLONG)

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili ögenin belirttiği adı taşıyan kuyruğa ilişkin açma ya da koyma işleminden kaynaklanan tamamlanma kodudur.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri MQCC_OK.

MQRR için Neden (MQLONG)

Bu neden kodu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki karşılık gelen ögenin belirttiği adı taşıyan kuyruğa ilişkin açma ya da koyma işleminin sonucudur.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri MQRC_NONE.

MQSCO-SSL/TLS yapılandırma seçenekleri

MQCD yapısındaki TLS alanlarıyla birlikte MQSCO yapısı, IBM MQ MQI client olarak çalışan bir uygulamanın, kanal protokolü TCP/IP olduğunda istemci bağlantısı için TLS kullanımını denetleyen yapılandırma seçeneklerini belirtmesine olanak sağlar. Yapı, MQCONNX çağrısındaki bir giriş değiştirgesidir.

Kullanılabilirlik

MQSCO yapısı aşağıdaki istemcilerde kullanılabilir:

-  AIX
-  IBM i
-  Linux
-  Windows

İstemci kanalına ilişkin kanal protokolü TCP/IP değilse, MQSCO yapısı yoksayılr.

Karakter kümesi ve kodlama







MQSCO içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQSCO_STRUC_ID	'SCO-'
<u>Sürüm</u> (yapı sürümü numarası)	MQSCO_CURRENT_VERSION	1
<u>KeyRepository</u> (anahtar havuzunun yeri)	Yok	Boş değerli dizilim ya da boşluklar
<u>CryptoHardware</u> (şifreleme donanımına ilişkin ayrıntılar)	Yok	Boş değerli dizilim ya da boşluklar
<u>AuthInfoRecCount</u> (var olan MQAIR kayıtlarının sayısı)	Yok	0
<u>AuthInfoRecOffset</u> (MQSCO ' nun başlangıcından ilk MQAIR kaydının görelî konumu)	Yok	0
<u>AuthInfoRecPtr</u> (ilk MQAIR kaydının adresi)	Yok	Boş değerli gösterge ya da boş değerli byte
Not: <i>Version</i> , MQSCO_VERSION_2değerinden küçükse, aşağıdaki iki alan yoksayılr.		

Çizelge 525. MQSCO içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>KeyResetCount</u> (TLS gizli anahtar sıfırlama sayısı)	MQSCO_RESET_COUNT_DEFAULT	0
“MQSCO için FipsRequired (MQLONG)” sayfa 559 (IBM MQ’inde FIPS sertifikalı şifreleme algoritmalarını kullanın)	MQSSL_FIPS_NO	0
Not: <i>Version</i> , MQSCO_VERSION_3değerinden küçükse, aşağıdaki iki alan yoksayılır.		
<u>EncryptionPolicySuiteB</u> (yalnızca Suite B şifreleme algoritmalarını kullanın)	MQ_SUITE_B_NONE, MQ_SUITE_B_NOT_AVAILABLE, MQ_SUITE_B_NOT_AVAILABLE, MQ_SUITE_B_NOT_AVAILABLE	1, 0, 0, 0
Not: <i>Version</i> , MQSCO_VERSION_4değerinden küçükse, aşağıdaki iki alan yoksayılır.		
<u>CertificateValİlkesi</u> (sertifika doğrulama ilkesi)	MQ_CERT_VAL_POLICY_DEFAULT	0
Not: <i>Version</i> , MQSCO_VERSION_5değerinden küçükse, aşağıdaki iki alan yoksayılır.		
<u>CertificateLabel</u> (kullanılmakta olan sertifika etiketinin ayrıntıları)	Yok	Boş değerli dizilim ya da boşluklar
Not: <i>Version</i> MQSCO_VERSION_6değerinden küçükse, kalan alanlar yoksayılır.		
  <u>KeyRepoPasswordPtr</u> (TLS anahtar havuzu parolasının adresi)	Yok	Boş değerli gösterge ya da boş değerli byte
  <u>KeyRepoPasswordOffset</u> (TLS anahtar havuzu parolasının görelî konumu)	Yok	0
  <u>KeyRepoPasswordLength</u> (TLS anahtar havuzu parolasının uzunluğu)	Yok	0

Notlar:

1. – simgesi, tek bir boş karakteri temsil eder.
2. C programlama dilinde, makro değişkeniMQSCO_DEFAULT , çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQSCO MySCO = {MQSCO_DEFAULT};
```

Dil bildirimleri

MQSCO için C bildirimi

```
typedef struct tagMQSCO MQSCO;
struct tagMQSCO {
    MQCHAR4    StrucId;                /* Structure identifier */
    MQLONG     Version;                /* Structure version number */
    MQCHAR256  KeyRepository;         /* Location of TLS key */
                                        /* repository */
    MQCHAR256  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 */
                                        /* algorithms */
    /* Ver:2 */
    MQLONG     EncryptionPolicySuiteB[4]; /* Use only Suite B */
    /* Ver:3 */
    MQLONG     CertificateValPolicy;  /* cryptographic algorithms */
                                        /* Certificate validation */
                                        /* policy */
    /* 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 */
};
```

MQSCO için COBOL bildirimi

```
** 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 **

```

MQSCO için PL/I bildirimi

```

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 */

```

MQSCO için Visual Basic bildirimi

```

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'
  'string'
  AuthInfoRecCount As Long 'Number of MQAIR records present'
  AuthInfoRecOffset As Long 'Offset of first MQAIR record from'
  'start of MQSCO structure'
  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

```

İlgili başvurular

“MQCNO-Bağlantı seçenekleri” sayfa 316

MQCNO yapısı, uygulamanın kuyruk yöneticisine yönelik bağlantıyla ilgili seçenekleri belirtmesine olanak sağlar. Yapı, MQCONNX çağrısındaki bir giriş/çıkış değıştirgesidir.

MQSCO için StrucId (MQCHAR4)

Bu, SSL/TLS yapılandırma seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQSCO_STRUC_ID.

Değer şöyle olmalıdır:

MQSCO_STRUC_ID

SSL/TLS yapılandırma seçenekleri yapısına ilişkin tanıttıcı.

C programlama dili için, MQSCO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQSCO_STRUC_ID ile aynı değere sahip, ancak bir dizgi yerine bir karakter dizisi.

MQSCO için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQSCO_VERSION_1

Version-1 TLS yapılandırma seçenekleri yapısı.

MQSCO_VERSION_2

Version-2 TLS yapılandırma seçenekleri yapısı.

MQSCO_VERSION_3

Version-3 TLS yapılandırma seçenekleri yapısı.

MQSCO_VERSION_4

Version-4 TLS yapılandırma seçenekleri yapısı.

MQSCO_VERSION_5

Version-5 TLS yapılandırma seçenekleri yapısı.

MQSCO_VERSION_6

Version-6 TLS yapılandırma seçenekleri yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQSCO_CURRENT_VERSION

TLS yapılandırma seçenekleri yapısının geçerli sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQSCO_VERSION_1' dir.

Multi MQSCO için KeyRepository (MQCHAR256)

Bu alan yalnızca IBM i, AIX, Linux, and Windows sistemlerinde çalışan IBM MQ MQI clients için geçerlidir. Anahtarların ve sertifikaların saklandığı anahtar veritabanı dosyasının konumunu belirtir. Dosya soneki belirtilmezse, otomatik olarak .kdb soneki eklenir.

Her anahtar veritabanı dosyasının ilişkili bir *parola saklama kütüğü* olabilir. Parola saklama dosyası, anahtar veritabanına programlı erişim sağlamak için kullanılan kodlanmış parolaları içerir. Parola saklama kütüğü aynı dizinde bulunmalı ve anahtar veritabanıyla aynı kütük sistemine sahip olmalı ve .sth sonekiyle bitmelidir.

Örneğin, anahtar veritabanı dosyası /xxx/yyy/key.kdb ise, parola saklama dosyası /xxx/yyy/key.sth olmalıdır; burada xxx ve yyy dizin adlarını gösterir.

Anahtar veritabanı parolası, MQSCO yapısının *KeyRepoPasswordPtr* ya da *KeyRepoPasswordOffset* alanı kullanılarak da belirtilebilir.

Değer alanın uzunluğundan kısaysa, değeri boş bir karakterle sonlandırın ya da alanın uzunluğuna kadar boşluklarla doldurun. Değer denetlenmez; anahtar havuzuna erişilirken hata oluşursa, çağrı başarısız olur; neden kodu MQRC_KEY_REPOSITORY_ERROR.

Bir IBM MQ MQI clientinden TLS bağlantısı çalıştırmak için, *KeyRepository* değerini geçerli bir anahtar veritabanı dosyası adına ayarlayın.

Bu bir giriş alanıdır. Bu alanın uzunluğu MQ_SSL_KEY_REPOSITORY_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki boş karakterdir.

MQSCO için CryptoHardware (MQCHAR256)

Bu alanda, istemci sistemine bağlı şifreleme donanımına ilişkin yapılandırma ayrıntıları verilir.

Alanı aşağıdaki biçimde bir dizgiye ayarlayın ya da boş bırakın:

```
GSK_PKCS11=the PKCS #11 driver path and file name;the PKCS #11  
token label;the PKCS #11 token password;symmetric cipher setting;
```

PKCS #11 arabirimine uyan şifreleme donanımını kullanmak için, örneğin, IBM 4960 ya da IBM 4764, PKCS #11 sürücü yolu, PKCS #11 simge etiketi ve PKCS #11 simge parolası dizgileri belirtilmelidir; bunların her biri noktalı virgülle sonlandırılmalıdır.

PKCS #11 sürücü yolu, PKCS #11 kartı için destek sağlayan paylaşılan kitaplığın mutlak yoludur. PKCS #11 sürücü dosyası adı, paylaşılan kitaplığın adıdır. PKCS #11 yolu ve dosya adı için gereken değerlerin bir örneği:

```
/usr/lib/pkcs11/PKCS11_API.so
```

PKCS #11 simge etiketi, donanımınızı yapılandırdığınız etiketle eşleşmelidir.

Şifreleme donanımı yapılandırması gerekmiyorsa, alanı boş ya da boş olarak ayarlayın.

Değer alanın uzunluğundan kısaysa, değeri boş bir karakterle sonlandırın ya da alanın uzunluğuna kadar boşluklarla doldurun. Değer geçerli değilse ya da şifreleme donanımının konfigürasyonunu tanımlamak için kullanıldığında bir hataya neden olursa, çağrı MQRC_CRYPTO_HARDWARE_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın uzunluğu MQ_SSL_CRYPTO_HARDWARE_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki boş karakterdir.

AuthInfoRecCount (MQSCO için MQLONG)

Bu, *AuthInfoRecPtr* ya da *AuthInfoRecOffset* alanları tarafından adreslenen kimlik doğrulama bilgileri (MQAIR) kayıtlarının sayısıdır. Daha fazla bilgi için, bkz. "[MQAIR-Kimlik doğrulama bilgileri kaydı](#)" sayfa 271. Değer sıfır ya da daha büyük olmalıdır. Değer geçerli değilse, çağrı MQRC_AUTH_INFO_REC_COUNT_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir.

MQSCO için AuthInfoRecOffset (MQLONG)

Bu, MQSCO yapısının başlangıcından itibaren ilk kimlik doğrulama bilgileri kaydının bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir. *AuthInfoRecCount* sıfırsa alan yoksayılır.

MQAIR kayıtlarını belirtmek için *AuthInfoRecOffset* ya da *AuthInfoRecPtr* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız; ayrıntılar için *AuthInfoRecPtr* alanının tanımına bakın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir.

AuthInfoRecPtr (PMQAIR)

Bu, ilk kimlik doğrulama bilgileri kaydının adresidir. *AuthInfoRecCount* sıfırsa alan yoksayılır.

MQAIR kayıtları dizisini aşağıdaki iki yoldan biriyle sağlayabilirsiniz:

- *AuthInfoRecPtr* gösterge alanını kullanarak

Bu durumda, uygulama MQSCO yapısından ayrı bir MQAIR kayıtları dizisi bildirebilir ve *AuthInfoRecPtr* dizisini dizinin adresine ayarlayabilir.

İşaretçi veri tipini destekleyen programlama dilleri için farklı ortamlarda (örneğin, C programlama dili) taşınabilir bir şekilde *AuthInfoRecPtr* ' i kullanmayı düşünün.

- *AuthInfoRecOffset* görelî konum alanını kullanarak

Bu durumda, uygulama MQSCO içeren bir bileşik yapıyı ve ardından MQAIR kayıtları dizisini bildirmeli ve *AuthInfoRecOffset* ögesini MQSCO yapısının başlangıcından itibaren dizideki ilk kaydın görelî

konumuna ayarlamalıdır. Bu değerin doğru olduğundan ve MQLONG içinde tutulabilecek bir değere sahip olduğundan emin olun (en kısıtlayıcı programlama dili COBOL 'dur; geçerli aralık -999 999 999 ile 999 999 999 999 arasındadır).

İşaretçi veri tipini desteklemeyen ya da gösterge veri tipini farklı ortamlara (örneğin, COBOL programlama dili) taşınmayacak şekilde uygulayan programlama dilleri için *AuthInfoRecOffset* komutunu kullanmayı düşünün.

Hangi tekniği seçerseniz seçin, *AuthInfoRecPtr* ve *AuthInfoRecOffset* yöntemlerinden yalnızca biri kullanılabilir; her ikisi de sıfır değilse, çağrı MQRCAUTH_INFO_REC_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizisidir.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizisi olarak bildirilir.

MQSCO için KeyResetCount (MQLONG)

Bu, gizli anahtar yeniden anlaşılmadan önce bir TLS görüşmesi içinde gönderilen ve alınan toplam şifrenmemiş bayt sayısını temsil eder.

Bayt sayısı, MCA tarafından gönderilen denetim bilgilerini içerir.

1-32 KB aralığında bir TLS gizli anahtar sıfırlama sayısı belirtirseniz, TLS kanalları 32 KB gizli anahtar sıfırlama sayısını kullanır. Bu, küçük TLS gizli anahtar sıfırlama değerleri için ortaya çıkabilecek aşırı anahtar sıfırlamalarını işleme maliyetini önlemek içindir.

Bu bir giriş alanıdır. Değer, 0 ile 999 999 999 999 arasında bir sayıdır ve varsayılan değer 0 'dır. Gizli anahtarların hiçbir zaman yeniden anlaşılmadığını belirtmek için 0 değerini kullanın.

MQSCO için FipsRequired (MQLONG)

IBM MQ , kullanılan şifreleme modüllerinin donanım ürünü tarafından sağlanacak şekilde şifreleme donanımıyla yapılandırılabilir; bunlar, kullanılmakta olan şifreleme donanım ürününe bağlı olarak belirli bir düzeyde FIPS onaylı olabilir. Şifreleme IBM MQ tarafından sağlanan yazılımda sağlanırsa yalnızca FIPS onaylı algoritmaların kullanılacağını belirtmek için bu alanı kullanın.

Not: AIX, Linux, and Windows işletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu [İşlem listesindeki NIST CMVP modüllerinde](#) aranarak görüntülenebilir.

IBM MQ kurulduğunda, FIPS onaylı bazı modüller sağlayan bir TLS şifreleme uygulaması da kurulur.

Değerler şunlar olabilir:

MQSSL_FIPS_NO

Bu varsayılan değerdir. Bu değere ayarlandığında:

- Belirli bir platformda desteklenen herhangi bir CipherSpec kullanılabilir.
- Şifreleme donanımı kullanılmadan çalıştırılırsa, CipherSpecs , IBM MQ platformlarında FIPS 140-2 sertifikalı şifreleme kullanılarak çalıştırılır.

FIPS onaylı CipherSpecs çizelgesinin bir listesi için [CipherSpecs' in etkinleştirilmes](#) başlıklı konuda açıklanan çizegeye bakın.

MQSSL_FIPS_YES

Bu değere ayarlandığında, şifrelemeyi gerçekleştirmek için şifreleme donanımı kullanmıyorsan,

- Bu istemci bağlantısına uygulanan CipherSpec içinde yalnızca FIPS onaylı şifreleme algoritmaları kullanılabilir.
- Belirli Şifre Belirtileri kullanılıyorsa, gelen ve giden TLS kanal bağlantıları yalnızca başarılı olur.

Ek bilgi için [CipherSpecs](#) özelliğinin etkinleştirilmesi başlıklı konuya bakın.

Not: Olanaklıysa, yalnızca FIPS CipherSpecs yapılandırılırsa, MQI istemcisi MQRC_SSL_INITIALIZATION_ERROR ile FIPS olmayan CipherSpec belirten bağlantıları reddeder. IBM MQ, bu tür bağlantıların tümünü reddetmeyi garanti etmez ve IBM MQ yapılandırmanızın FIPS uyumlu olup olmadığını belirlemek sizin sorumluluğunuzdadır.

EncryptionPolicySuiteB (MQSCO için MQLONG)

Bu alan, Suite B uyumlu şifrelemenin kullanılıp kullanılmadığını ve hangi güç düzeyinin kullanıldığını belirtir. Değer aşağıdakilerden biri ya da daha fazlası olabilir:

- MQ_SUITE_B_NONE
Suite B uyumlu şifreleme kullanılmaz.
- MQ_SUITE_B_128_BIT
Suite B 128-bit güvenlik kullanılır.
- MQ_SUITE_B_192_BIT
Suite B 192-bit güvenlik kullanılır.

Not: MQ_SUITE_B_NONE değerinin bu alanda başka bir değerle kullanılması geçersizdir.

MQSCO için CertificateValİlkesi (MQLONG)

Bu alan, hangi sertifika doğrulama ilkesinin kullanıldığını belirtir.

Alan aşağıdaki değerlerden birine ayarlanabilir:

MQ_CERT_VAL_POLICY_ANY

Güvenli yuva kitaplığı tarafından desteklenen sertifika geçerlilik denetimi ilkelerinin her birini uygulayın. İlkelere herhangi biri sertifika zincirini geçerli sayarsa sertifika zincirini kabul edin.

MQ_CERT_VAL_POLICY_RFC5280

Yalnızca RFC5280 uyumlu sertifika doğrulama ilkesini uygulayın. Bu ayar, HERHANGİ BİRİ ayarından daha sıkı doğrulama sağlar, ancak bazı eski dijital sertifikaları reddeder.

Bu alanın ilk değeri: MQ_CERT_VAL_POLICY_ANY

MQSCO için CertificateLabel (MQCHAR64)

Bu alan, kullanılmakta olan sertifika etiketinin ayrıntılarını verir.

IBM MQ, *CertificateLabel* alanı için varsayılan değer boşluk olarak başlatır.

Bu, yürütme sırasında varsayılan değer olarak yorumlanır ve geriye doğru uyumludur.

Örneğin, 5.0' dan küçük bir MQSCO sürümünü belirtmek ya da *CertificateLabel* alanı için varsayılan değer olan boşlukları kullanmak, önceden var olan *ibmwebspheremquser_id* varsayılan değerini kullanır.

KeyRepoPasswordPtr (MQSCO için MQPTR)

Bu, TLS anahtar havuzu parolası bayt cinsinden adresidir.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizisidir. *Version*, MQSCO_VERSION_6' dan küçükse bu alan yoksayılr.

Bu alan yalnızca IBM i, AIX, Linux, and Windows sistemlerinde çalışan IBM MQ MQI clients için geçerlidir.

Anahtar havuzu parolası, **runmqicred** yardımcı programı kullanılarak şifrelenmiş bir parola ya da düz metin dizisi olarak belirtilebilir.

Şifrelenmiş bir parola belirtirseniz, aynı istemci uygulaması tarafından sağlanan MQCSP yapısında parolayı şifrelemek için kullanılan ilk anahtarı belirtin.

Bu alan kullanılarak belirtilen anahtar havuzu parolası, *MQKEYRPWD* ortam değişkeni ya da istemci yapılandırma dosyasının SSL kısmına ilişkin *SSLKeyRepositoryPassword* özelliği kullanılarak belirtilen anahtar havuzu parolasını geçersiz kılar.

Anahtar havuzu parolasını belirtmek için *KeyRepoPasswordOffset* ya da *KeyRepoPasswordPtr* kullanabilirsiniz, ancak ikisini birden kullanamazsınız.

İlgili görevler

AIX, Linux ve Windows üzerinde IBM MQ MQI istemcisi için başlangıç anahtarı sağlanması

[IBM MQ bileşeni yapılandırma dosyalarındaki parolaları koruma](#)

İlgili başvurular

[mqicred \(IBM MQ istemci parolalarını koru\)](#)

[“MQCSP için InitialKeyPtr \(MQPTR\)” sayfa 341](#)

Parola koruma sistemine ilişkin ilk anahtarın adresi.

V 9.3.0 V 9.3.0 Multi KeyRepoPasswordOffset (MQSCO için MQLONG)

Bu, MQSCO yapısının başlangıcından itibaren TLS anahtar havuzu geçiş tümcesinin bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir.

Anahtar havuzu parolasını belirtmek için *KeyRepoPasswordOffset* ya da *KeyRepoPasswordPtr* kullanabilirsiniz, ancak ikisini birden kullanamazsınız. Daha fazla bilgi için *KeyRepoPasswordPtr* alanının açıklamasına bakın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir. *Version* , MQSCO_VERSION_6' dan küçükse bu alan yoksayılr.

V 9.3.0 V 9.3.0 Multi KeyRepoPasswordLength (MQSCO için MQLONG)

Bu, TLS anahtar havuzu parolası uzunluğudur.

IBM i' da anahtar havuzu parolası uzunluğu üst sınırı 128 karakterdir. Anahtar havuzu parolası izin verilen uzunluk üst sınırından fazlaysa, MQRC_KEY_REPOSITORY_ERROR ile bağlantı başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir. *Version* , MQSCO_VERSION_6' dan küçükse bu alan yoksayılr.

MQSD-Abonelik tanımlayıcısı

MQSD yapısı, yapılmakta olan abonelikle ilgili ayrıntıları belirtmek için kullanılır. Yapı, MQSUB çağrısında bir giriş/çıkış değıştirgesidir. Daha fazla bilgi için bkz. [MQSUB kullanım notları](#).

Kullanılabilirlik

MQSD yapısı aşağıdaki altyapılarda bulunur:

- **AIX** AIX
- **IBM i** IBM i
- **Linux** Linux
- **Windows** Windows
- **z/OS** z/OS

ve bu sistemlere bağılı IBM MQ MQI clients için.

Sürüm

MQSD ' nin yürürlükteki sürümü: MQSD_VERSION_1.

Karakter kümesi ve kodlama

MQSD ' deki veriler, **CodedCharSetId** kuyruk yöneticisi özneliğı tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında yer almalıdır.

Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Yönetilen abonelikler

Bir uygulamanın belirli bir kuyruğu aboneliğiyle eşleşen yayınlar için hedef olarak kullanması gerekmiyorsa, yönetilen abonelik özelliğini kullanabilir. Bir uygulama yönetilen abonelik kullanmayı seçerse, kuyruk yöneticisi, MQSUB çağrısından çıkış olarak bir nesne tanıtıcısı sağlayarak aboneyi yayınlanan iletilerin gönderildiği hedef hakkında bilgilendirir. Daha fazla bilgi için bkz. [Hobj \(MQHOBJ\)-giriş/çıkış](#).

Abonelik kaldırıldığında, kuyruk yöneticisi aşağıdaki durumlarda yönetilen hedeften alınmayan iletileri temizlemeyi de taahhüt eder:

- Abonelik kaldırıldığında-MQCO_REMOVE_SUB ile MQCLOSE kullanımı ile yönetilen Hobj kapatılır.
- Kalıcı olmayan bir abonelik (MQSO_NON_DURABLE) kullanılarak bir uygulamayla bağlantı kesildiğinde örtük olarak (MQSO_NON_DURABLE)
- Süresi dolduğu ve yönetilen Hobj kapatıldığı için bir abonelik kaldırıldığında süre bitimine göre.

Sürekli olmayan aboneliklerle yönetilen abonelikleri kullanmanız gerekir; böylece bu temizleme işlemi gerçekleştirilebilir ve böylece, kapalı kalıcı olmayan aboneliklere ilişkin iletiler kuyruk yöneticinizde yer kaplamaz. Sürekli abonelikler, yönetilen hedefleri de kullanabilir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQSD_STRUC_ID	'SD--'
Sürüm (yapı sürümü numarası)	MQSD_VERSION_1	1
Seçenekler (seçenekler)	MQSO_NON_DURABLE	0
ObjectName (nesne adı)	Yok	Boş değerli dizilim ya da boşluklar
AlternateUserKimliği (diğer kullanıcı kimliği)	Yok	Boş değerli dizilim ya da boşluklar
AlternateSecurityTanıtıcısı (diğer güvenlik tanıtıcısı)	MQSID_YOK	Boş Değerler
SubExpiry (abonelik süre bitimi)	MQEI_UNLIMITED	-1
ObjectString (nesne dizgisi)	Yok	MQCHARV için tanımlanan adlar ve değerler
SubName (abonelik adı)	Yok	MQCHARV için tanımlanan adlar ve değerler
SubUserVerileri (abonelik kullanıcı verileri)	Yok	MQCHARV için tanımlanan adlar ve değerler
SubCorrelTanıtıcısı (abonelik ilinti tanıtıcısı)	MQCI_NONE	Boş Değerler
PubPriority (yayın önceliği)	MQPRI_PRIORITY_AS_Q_DEF	-3

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>PubAccountingBelirteci</u> (yayın muhasebesi simgesi)	MQACT_NONE	Boş Değerler
<u>PubAppIdentityData</u> (yayın uygulaması kimlik verileri)	Yok	Boş değerli dizilim ya da boşluklar
<u>SelectionString</u> (seçim ölçütlerini sağlayan dizgi)	Yok	MQCHARV için tanımlanan adlar ve değerler
<u>SubLevel</u> (abonelik düzeyi)	Yok	1
<u>ResObjectDizesi</u> (uzun nesne adı)	Yok	MQCHARV için tanımlanan adlar ve değerler

Notlar:

1. – simgesi, tek bir boş karakteri temsil eder.
2. Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.
3. C programlama dilinde, makro değişkeniMQSD_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQSD MySD = {MQSD_DEFAULT};
```

Dil bildirimleri

MQSD için C bildirim

```
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 */
};
```

MQSD için COBOL bildirim

```
** 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.

```

MQSD için PL/I bildirimi

```

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 */

```

MQSD için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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 OF 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_VSLLENGTH 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 OF 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_VSLLENGTH 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 OF 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_VSLLENGTH 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_VSLLENGTH 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_VSLLENGTH 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)

```

MQSD için StrucId (MQCHAR4)

Bu, abonelik tanımlayıcı yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQSD_STRUC_ID.

Değer şöyle olmalıdır:

MQSD_STRUC_ID

Abonelik tanımlayıcı yapısına ilişkin tanıtıcı.

C programlama dili için, MQSD_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQSD_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQSD için sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQSD_VERSION_1

Version-1 Abonelik Tanımlayıcı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQSD_CURRENT_VERSION

Abonelik Tanımlayıcı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQSD_VERSION_1' dir.

MQSD için seçenekler (MQLONG)

Bu, MQSUB çağrısının işlemi denetleme seçenekleri sağlar.

Aşağıdaki seçeneklerden en az birini belirtmelisiniz:

- MQSO_ALTER
- MQSO_RESUME
- MQSO_OLUŞTURMA

birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemi kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyse).

Geçerli olmayan birleşimler bu konuda belirtilmiştir; diğer birleşimler geçerlidir.

Erişim ya da yaratma seçenekleri: Erişim ve yaratma seçenekleri, bir aboneliğin oluşturulup oluşturulmayacağını ya da var olan bir aboneliğin döndürülüp döndürülmeyeceğini ya da değiştirilip değiştirilmeyeceğini denetler. Bu seçeneklerden en az birini belirtmelisiniz.

<i>Çizelge 526. Geçerli erişim ve yaratma seçenekleri birleşimleri</i>	
Seçenek birleşimi	Notlar
MQSO_OLUŞTURMA	Yoksa bir abonelik oluşturur. Abonelik varsa bu birleşim başarısız olur.
MQSO_RESUME	Var olan bir aboneliği sürdürür. Abonelik yoksa bu birleşim başarısız olur.
MQSO_CREATE + MQSO_RESUME	Yoksa bir abonelik oluşturur ve varsa, eşleşen bir aboneliği sürdürür. Bu birleşim, birçok kez çalıştırılan bir uygulamada kullanıldığında yararlıdır.

Çizelge 526. Geçerli erişim ve yaratma seçenekleri birleşimleri (devamı var)	
Seçenek birleşimi	Notlar
MQSO_ALTER (nota bakın)	Var olan bir aboneliği sürdürür ve MQSD ' de belirtilen alanlarla eşleşecek şekilde değiştirir. Abonelik yoksa bu birleşim başarısız olur.
MQSO_CREATE + MQSO_ALTER (nota bakın)	Yoksa abonelik yaratır ve varsa, eşleşen bir aboneliği sürdürür; varsa, MQSD ' de belirtilen alanlarla eşleşecek şekilde değiştirilir. Bu birleşim, devam etmeden önce aboneliğinin belirli bir durumda olduğundan emin olmak isteyen bir uygulamada kullanıldığında kullanışlıdır.

Not:

MQSO_ALTER ile belirtilen seçenekler de MQSO_RESUME belirtilebilir, ancak bu birleşim yalnızca MQSO_ALTER ile ilgili ek bir etkiye sahip değildir. MQSO_ALTER, bir aboneliği değiştirmek için MQSUB ' un çağrılması aboneliğin de sürdürüleceğini belirttiği için, MQSO_RESUME ögesini belirtir. Bunun tersi doğru değildir, ancak: bir aboneliğin sürdürülmesi değiştirileceği anlamına gelmez.

MQSO_OLUŞTURMA

Belirtilen konu için yeni bir abonelik oluşturun. Aynı *SubName* ögesini kullanan bir abonelik varsa, çağrı MQRC_SUB_ALREADY_EXISTS ile başarısız olur. MQSO_CREATE seçeneği MQSO_RESUME ile birleştirilerek bu başarısızlık önlenir. *SubName* her zaman gerekli değildir. Daha fazla ayrıntı için bu alanın açıklamasına bakın.

MQSO_RESUME ile MQSO_CREATE birleştirildiğinde, belirtilen *SubName* için önceden var olan bir aboneliğe bir tanıtıcı döndürülür; abonelik yoksa, MQSD ' de sağlanan tüm alanlar kullanılarak yeni bir tane yaratılır.

MQSO_CREATE, MQSO_ALTER ile de benzer bir etki için birleştirilebilir.

MQSO_RESUME

SubName ile belirtilen önceden var olan bir aboneliğe tanıtıcı döndürür. Eşleşen abonelik özneliklerinde değişiklik yapılmadı ve bunlar MQSD yapısındaki çıkışta döndürülür. Yalnızca şu MQSD alanları kullanılır: StrucId, Version, Options, AlternateUserId ve AlternateSecurityId ve SubName.

Tam abonelik adıyla eşleşen bir abonelik yoksa, çağrı MQRC_NO_SUBSCRIPTION neden koduyla başarısız olur. MQSO_CREATE seçeneği MQSO_RESUME ile birleştirilerek bu başarısızlık önlenir.

Aboneliğin kullanıcı kimliği, aboneliği oluşturan kullanıcı kimliğidir ya da daha sonra farklı bir kullanıcı kimliği tarafından değiştirildiyse, en son başarılı değişikliğin kullanıcı kimliğidir. Bir AlternateUser kimliği kullanılırsa ve o kullanıcı için diğer kullanıcı kimliklerinin kullanılmasına izin verilirse, diğer kullanıcı kimliği, aboneliğin yapıldığı kullanıcı kimliği yerine, aboneliği yaratan kullanıcı kimliği olarak kaydedilir.

MQSO_ANY_USERID seçeneği olmadan yaratılmış eşleşen bir abonelik varsa ve aboneliğin kullanıcı kimliği, abonelik için tanıtıcı isteyen uygulamadan farklıysa, çağrı başarısız olur. Neden kodu: MQRC_IDENTITY_MISMATCH.

Eşleşen bir abonelik varsa ve şu anda kullanıyorsa, çağrı MQRC_SUBSCRIPTION_IN_USE ile başarısız olur.

SubName içinde adı belirtilen abonelik, bir uygulamadan sürdürmek ya da değiştirmek için geçerli bir abonelik değilse, çağrı MQRC_INVALID_SUBSCRIPTION ile başarısız olur.

MQSO_RESUME, MQSO_ALTER tarafından örtük olarak belirtilmiştir; bu nedenle, bu seçeneği o seçenekle birleştirmeniz gerekmez. Ancak, iki seçeneğin birleştirilmesi hataya neden olmaz.

MQSO ALTER

SubName içinde adla belirtilen tam abonelik adıyla eşleşen, önceden var olan bir aboneliğe bir tanıtıcı döndürür. Aboneliğin MQSD ' de belirtilenden farklı olan öznitelikleri, o öznitelik için değişikliğe izin verilmediği sürece abonelikte değiştirilir. Ayrıntılar her özneliğin tanımında belirtilir ve aşağıdaki tabloda özetlenir. Değiştirilemeyen bir özneliği değiştirmeye ya da MQSO_IMMUTABLE seçeneğini ayarlı bir aboneliği değiştirmeye çalışırsanız, çağrı aşağıdaki çizelgede gösterilen neden koduyla başarısız olur.

Tam abonelik adıyla eşleşen bir abonelik yoksa, çağrı MQRC_NO_SUBSCRIPTION neden koduyla başarısız olur. MQSO_CREATE seçeneğini MQSO ALTER ile birleştirerek bu başarısızlığı önleyebilirsiniz.

MQSO_CREATE ile MQSO ALTER birleşimi, varsa, belirtilen *SubName* için önceden var olan bir aboneliğe bir tanıtıcı döndürür; var olan bir abonelik yoksa, MQSD ' de sağlanan tüm alanlar kullanılarak yeni bir tane yaratılır.

Aboneliğin kullanıcı kimliği, aboneliği oluşturan kullanıcı kimliğidir ya da daha sonra farklı bir kullanıcı kimliği tarafından değiştirilirse, en son ve başarılı değişikliğin kullanıcı kimliğidir. Bir AlternateUser kimliği kullanılırsa ve o kullanıcı için diğer kullanıcı kimliklerinin kullanılmasına izin verilirse, diğer kullanıcı kimliği, aboneliğin yapıldığı kullanıcı kimliği yerine, aboneliği yaratan kullanıcı kimliği olarak kaydedilir.

MQSO_ANY_USERID seçeneği olmadan yaratılmış eşleşen bir abonelik varsa ve aboneliğin kullanıcı kimliği, abonelik için tanıtıcı isteyen uygulamadan farklıysa, çağrı başarısız olur. Neden kodu MQRC_IDENTITY_MISMATCH.

Eşleşen bir abonelik varsa ve şu anda kullanıyorsa, çağrı MQRC_SUBSCRIPTION_IN_USE ile başarısız olur.

SubName içinde adı belirtilen abonelik, bir uygulamadan sürdürmek ya da değiştirmek için geçerli bir abonelik değilse, çağrı MQRC_INVALID_SUBSCRIPTION ile başarısız olur.

Aşağıdaki çizelge, MQSO ALTER 'in MQSD ve MQSUB' daki öznitelik değerlerini değiştirme yeteneğini göstermektedir.

Veri tipi tanımlayıcısı ya da işlev çağırısı	Alan adı	Bu öznitelik MQSO ALTER kullanılarak değiştirilebilir	Neden Kodu
MQSD	Dayanıklılık seçenekleri	Hayır	MQRC_DURABILITY_NOT_ALTERABLE
MQSD	Hedef Seçenekleri	Evet	Yok
MQSD	Kayıt seçenekleri	Evet (bkz. not "1" sayfa 569)	MQSO_GROUP_SUB ögesini değiştirmeye çalışırsanız, MQRC_GROUPING_NOT_ALTERABLE
MQSD	Yayınlama seçenekleri	Evet (bkz. not "2" sayfa 569)	Yok
MQSD	Genel arama karakteri seçenekleri	Hayır	MQRC_TOPIC_NOT_DEĞİŞTİRİLEBİLİR
MQSD	Diğer seçenekler	Hayır (bkz. not "3" sayfa 569)	Yok
MQSD	ObjectName	Hayır	MQRC_TOPIC_NOT_DEĞİŞTİRİLEBİLİR
MQSD	AlternateUserKimliği	Hayır (bkz. not "4" sayfa 569)	Yok
MQSD	AlternateSecurityTanıtıcısı	Hayır (bkz. not "4" sayfa 569)	Yok
MQSD	SubExpiry	Evet	Yok
MQSD	ObjectString	Hayır	MQRC_TOPIC_NOT_DEĞİŞTİRİLEBİLİR
MQSD	SubName	Hayır (bkz. not "5" sayfa 569)	Yok
MQSD	SubUserVerileri	Evet	Yok

Çizelge 527. MQSD ve MQSUB içindeki değiştirilebilecek öznitelikler (devamı var)

Veri tipi tanımlayıcısı ya da işlev çağırısı	Alan adı	Bu öznitelik MQSO ALTER kullanılarak değiştirilebilir	Neden Kodu
MQSD	SubCorrelTanıtıcısı	Evet (bkz. not "6" sayfa 569)	Gruplanmış abonelikteyken MQRC_GROUPING_NOT_ALTERABLE
MQSD	PubPriority	Evet	Yok
MQSD	PubAccountingSimgesi	Evet	Yok
MQSD	PubApplIdentityData	Evet	Yok
MQSD	SubLevel	Hayır	MQRC_SUBLEVEL_NOT_ALTERABLE
MQSUB	Hobj.	Evet (bkz. not "6" sayfa 569)	Gruplanmış abonelikteyken MQRC_GROUPING_NOT_ALTERABLE

Notlar:

1. MQSO_GROUP_SUB değiştirilemiyor.
2. MQSO_NEW_PUBLICATIONS_ONLY, aboneliğin bir parçası olmadığı için değiştirilemiyor
3. Bu seçenekler aboneliğin bir parçası değildir
4. Bu öznitelik, aboneliğin bir parçası değil
5. Bu öznitelik, değiştirilmekte olan aboneliğin tanıtıcısıdır
6. Gruplanmış bir alt (MQSO_GROUP_SUB) ögesinin bir parçası dışında değiştirilebilen

Dayanıklılık seçenekleri: Aşağıdaki seçenekler, aboneliğin ne kadar dayanıklı olduğunu denetler. Bu seçeneklerden yalnızca birini belirleyebilirsiniz. MQSO ALTER seçeneğini kullanarak var olan bir aboneliği değiştiriyorsanız, aboneliğin dayanıklılığını değiştiremezsiniz. MQSO_RESUME kullanılarak bir MQSUB çağırısından dönüş yapılıyorsa, uygun dayanıklılık seçeneği ayarlanır.

MQSO DURABLE

Bu konuya ilişkin aboneliğin, MQCO_REMOVE_SUB seçeneğiyle MQCLOSE kullanılarak belirtik olarak kaldırılıncaya kadar kalmasını isteyin. Bu abonelik belirtik olarak kaldırılmazsa, kuyruk yöneticisine yönelik bu uygulama bağlantısı kapatıldıktan sonra da kalır.

Sürekli aboneliğe izin verilmemesi olarak tanımlanan bir konuya sürekli abonelik istenirse, çağrı MQRC_DURABILITY_NOT_ALLOWED ile başarısız olur.

MQSO_NON_DURABLE

Önceden belirtik olarak kaldırılmadıysa, kuyruk yöneticisine yönelik uygulamalar bağlantısı kapatıldığında bu konuya ilişkin aboneliğin kaldırılmasını isteyin. MQSO_NON_DURABLE, MQSO_DURABLE seçeneğinin karşıtıdır ve program belgelerine yardımcı olmak için tanımlanır. İkisi de belirtilmezse, varsayılan değer budur.

Hedef seçenekleri: Aşağıdaki seçenek, abone olunan bir konuya ilişkin yayınların gönderileceği hedefi denetler. MQSO ALTER seçeneğini kullanarak var olan bir abonelik değiştiriliyorsa, aboneliğe ilişkin yayınlar için kullanılan hedef değiştirilebilir. MQSO_RESUME kullanılarak bir MQSUB çağırısından dönüş yapılırsa, uygunsuz bu seçenek ayarlanır.

MQSO_YÖNETİLDİR

Yayınların gönderildiği hedefin kuyruk yöneticisi tarafından yönetilmesini isteyin.

Hobj içinde döndürülen nesne tanıtıcısı, kuyruk yöneticisi tarafından yönetilen bir kuyruğu temsil eder ve sonraki MQGET, MQCB, MQINQ ya da MQCLOSE çağrılarıyla birlikte kullanılır.

MQSO_MANAGED belirtilmediğinde, **Hobj** değiştirilmesinde önceki bir MQSUB çağırısından döndürülen bir nesne tanıtıcısı belirtilemez.

MQSO_NO_MULTICAST

Yayınların gönderildiği hedefin çok hedefli bir grup adresi olmadığını belirtin. Bu seçenek yalnızca MQSO_MANAGED seçeneğiyle birlikte kullanıldığında geçerlidir. **Hobj** parametresinde bir kuyruk tanıtıcısı sağlandığında, bu abonelik için çoklu yayın kullanılamaz ve seçenek geçerli değildir.

Konu, MCAST (ONLY) ayarı kullanılarak yalnızca çoklu yayın aboneliklerine izin verecek şekilde tanımlanırsa, çağrı MQR_C_MULTICAST_REQUIRED neden koduyla başarısız olur.

Kapsam Seçeneği: Aşağıdaki seçenek, yapılmakta olan aboneliğin kapsamını denetler. MQSO ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, bu abonelik kapsamı seçeneği değiştirilemez. MQSO-RESUME kullanılarak bir MQSUB çağrısından dönüş yapılıyorsa, uygun kapsam seçeneği ayarlanır.

MQSO_SCOPE_QMGR

Bu abonelik yalnızca yerel kuyruk yöneticisinde yapılır. Ağdaki diğer kuyruk yöneticilerine bir yetkili sunucu aboneliği dağıtılmadı. Yalnızca bu kuyruk yöneticisinde yayınlanan yayınlar bu aboneye gönderilir. Bu, SUBSCOPE konu özneliği kullanılarak ayarlanan tüm davranışları geçersiz kılar.

Not: Ayarlanmazsa, abonelik kapsamı SUBSCOPE konu özneliği tarafından belirlenir.

Kayıt seçenekleri: Aşağıdaki seçenekler, bu abonelik için kuyruk yöneticisine yapılan kaydın ayrıntılarını denetler. MQSO ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, bu kayıt seçenekleri değiştirilebilir. MQSO-RESUME kullanılarak bir MQSUB çağrısından dönüş olduğunda, uygun kayıt seçenekleri ayarlanır.

MQSO_GROUP_SUB

Bu abonelik, aynı kuyruk kullanılarak aynı SubLevel ' in diğer abonelikleriyle gruplanacak ve aynı ilinti tanıtıcısını belirterek, birden çok yayın iletilsinin abonelik grubuna sağlanmasına neden olacak yayınların, üst üste gelen bir dizi konu dizisi kullanıldığından, yalnızca bir iletinin kuyruğa gönderilmesine neden olacak şekilde gruplanacaktır. Bu seçenek kullanılmazsa, eşleşen her benzersiz abonelik (SubName ile tanıtılır), yayının bir kopyasıyla birlikte sağlanır; bu, yayının birden çok kopyasının birden çok abonelik tarafından paylaşılan kuyruğa konabileceği anlamına gelebilir.

Yalnızca gruptaki en önemli abonelik, yayının bir kopyasıyla birlikte sağlanır. En önemli abonelik, genel arama karakterinin bulunduğu noktaya kadar Tam konu adını temel alır. Grup içinde genel arama karakteri şemalarının bir karışımı kullanılırsa, yalnızca genel arama karakterinin konumu önemlidir. Aynı kuyruğu paylaşan abonelikler grubunda farklı genel arama karakteri şemalarını birleştirmemeniz önerilir.

Yeni bir gruplanmış abonelik yaratılırken benzersiz bir SubName olmalıdır; ancak, grupta var olan bir aboneliğin tam konu adıyla eşleşiyorsa, çağrı MQR_DUPLICATE_GROUP_SUB ile başarısız olur.

Gruptaki en önemli abonelik MQSO_NOT_OWN_PUBS ' yi de belirtiyorsa ve bu aynı uygulamadan bir yayınsa, kuyruğa yayın teslim edilmez.

Bu seçenekle yapılan bir abonelik değiştirilirken, gruplamayı belirten alanlar, MQSUB çağrısında Hobj (kuyruk ve kuyruk yöneticisi adını gösterir) ve SubCorreltanıtıcısı değiştirilemez. Bunları değiştirme girişimi, çağrıyı MQR_GROUPING_NOT_ALTERABLE ile başarısız olmasına neden olur.

Bu seçenek, MQCI_NONE olarak ayarlanmamış bir SubCorreltanıtıcısıyla MQSO_SET_CORREL_ID ile birleştirilmeli ve MQSO_MANAGED ile birleştirilmemelidir.

MQSO_ANY_USERID

MQSO_ANY_USERID belirtildiğinde, abonenin kimliği tek bir kullanıcı kimliğiyle sınırlı değildir. Bu, herhangi bir kullanıcının uygun yetkiye sahip olduğunda aboneliği değiştirmesini ya da sürdürmesini sağlar. Aynı anda yalnızca tek bir kullanıcı aboneliğe sahip olabilir. Başka bir uygulama tarafından kullanılmakta olan bir aboneliği kullanmaya devam etme girişimi, çağrı MQR_SUBSCRIPTION_IN_USE ile başarısız olmasına neden olur.

Bu seçeneği var olan bir aboneliğe eklemek için MQSUB çağrısı (MQSO ALTER kullanılarak), özgün aboneliğin kendisiyle aynı kullanıcı kimliğinden gelmelidir.

Bir MQSUB çağrısı MQSO_ANY_USERID ile var olan bir aboneliğe gönderme yapıyorsa ve kullanıcı kimliği özgün abonelikten farklıysa, çağrı ancak yeni kullanıcı kimliğinin konuya abone olma yetkisi varsa başarılı olur. İşlem hatasız olarak tamamlandığında, bu abonenin gelecekteki yayınları, yayın iletilsinde yeni kullanıcı kimliği ayarlanmış olarak aboneler kuyruğuna yerleştirilir.

Hem MQSO_ANY_USERID, hem de MQSO_FIXED_USERID belirtmeyin. İkisi de belirtilmezse, varsayılan değer MQSO_FIXED_USERID 'dir.

MQSO_FIXED_USERID

MQSO_FIXED_USERID belirtildiğinde, abonelik yalnızca aboneliği değiştirmek için son kullanıcı kimliği tarafından değiştirilebilir ya da sürdürülebilir. Abonelik değiştirilmediyse, aboneliği yaratan kullanıcı kimliği olur.

Bir MQSUB komutu MQSO_ANY_USERID ile var olan bir aboneliğe gönderme yaparsa ve MQSO_ALTER seçeneğini kullanmak için aboneliği MQSO_FIXED_USERID kullanarak değiştirirse, aboneliğin kullanıcı kimliği artık bu yeni kullanıcı kimliğinde düzeltilmiştir. Arama, yalnızca yeni kullanıcı kimliğinin konuya abone olma yetkisi varsa başarılı olur.

Aboneliğin sahibi olarak kaydedilenden başka bir kullanıcı kimliği bir MQSO_FIXED_USERID aboneliğini sürdürmeyi ya da değiştirmeyi denerse, çağrı MQRC_IDENTITY_MISMATCH ile başarısız olur. Bir aboneliğin sahip olan kullanıcı kimliği, DISPLAY SBSTATUS komutu kullanılarak görüntülenebilir.

Hem MQSO_ANY_USERID, hem de MQSO_FIXED_USERID belirtmeyin. İkisi de belirtilmezse, varsayılan değer MQSO_FIXED_USERID 'dir.

Yayın seçenekleri: Aşağıdaki seçenekler, yayınların bu aboneye gönderilme şeklini denetler.

MQSO_ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, bu yayınlama seçenekleri değiştirilebilir.

MQSO_NOT_OWN_PUBS

Aracıya, uygulamanın kendi yayınlarından hiçbirini görmek istemediğini bildirir. Bağlantı tanıtıcıları aynıysa, yayınların aynı uygulamadan kaynaklandığı kabul edilir. MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılırsa, uygunsuz bu seçenek ayarlanır.

MQSO_NEW_PUBLICATIONS_ONLY

Bu abonelik oluşturulduğunda, şu anda alıkonan hiçbir yayın gönderilmez, yalnızca yeni yayınlar gönderilir. Bu seçenek yalnızca MQSO_CREATE belirtildiğinde geçerlidir. Abonelikte yapılan sonraki değişiklikler, yayınların akışını değiştirmez ve bu nedenle bir konuda tutulan yayınlar, aboneye yeni yayınlar olarak gönderilir.

Bu seçenek MQSO_CREATE olmadan belirtilirse, çağrı MQRC_OPTIONS_ERROR ile başarısız olur. MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılırsa, abonelik bu seçenek kullanılarak yaratılmış olsa bile bu seçenek ayarlanmaz.

Bu seçenek kullanılmazsa, önceden alıkonan iletiler sağlanan hedef kuyruğa gönderilir.

Bu işlem bir hata nedeniyle başarısız olursa, MQRC_RETAINED_MSG_Q_ERROR ya da MQRC_RETAINED_NOT_TESLIM edilirse, abonelik yaratılması başarısız olur.

MQSO_PUBLICATIONS_ON_REQUEST

Bu seçeneğin ayarlanması, abonenin özellikle gerektiğinde bilgi isteyeceğini belirtir. Kuyruk yöneticisi aboneye istenmeyen iletiler göndermez. Önceki bir MQSUB çağrısından Hsub tanıtıcısı kullanılarak bir MQSUBRQ çağrısı her yapıldığında, alıkonan yayın (ya da bir joker karakter belirtilirse birden çok yayın) aboneye gönderilir. Bu seçenek kullanılarak MQSUB çağrısının sonucu olarak hiçbir yayın gönderilmez. MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılırsa, uygunsuz bu seçenek ayarlanır.

Bu seçenek, 1 'den büyük bir SubLevel ile birlikte geçerli değildir.

Önden okuma seçenekleri: Aşağıdaki seçenekler, kalıcı olmayan iletilerin uygulama öncesinde bir uygulamaya gönderilip gönderilmeyeceğini denetler.

MQSO_READ_AHEAD_AS_Q_DEF

MQSUB çağrısı yönetilen bir tanıtıcı kullanıyorsa, abone olunan konuyla ilişkilendirilmiş model kuyruğunun varsayılan önden okuma özneliği, iletilerin uygulama tarafından istenmeden önce uygulamaya gönderilip gönderilmeyeceğini belirler.

Bu varsayılan değerdir.

MQSO_NO_READ_ÖNDEM

MQSUB çağrısı yönetilen bir tanıtıcı kullanıyorsa, uygulama bunları istemeden önce uygulamaya ileti gönderilmez.

MQSO_READ_ÖNDEN

MQSUB çağırısı yönetilen bir tanıtıcı kullanıyorsa, uygulama bunları istemeden önce uygulamaya iletiler gönderilebilir.

Not:

Aşağıdaki notlar, önden okuma seçenekleri için geçerlidir:

1. Bu seçeneklerden yalnızca biri belirtilebilir. Hem MQOO_READ_ÖNDEN, hem de MQOO_NO_READ_ÖNDEN belirtilirse, neden kodu MQRC_OPTIONS_ERROR döndürülür. Bu seçenekler yalnızca MQSO_MANAGED belirtilirse geçerlidir.
2. Bunlar, daha önce açılmış bir kuyruk geçirildiğinde MQSUB için geçerli değildir. İstendiğinde önden okuma etkinleştirilmeyebilir. İlk MQGET çağırısında kullanılan MQGET seçenekleri, önden okuma özelliğinin etkinleştirilmesini engelleyebilir. Ayrıca, istemci önden okuma özelliğinin desteklenmediği bir kuyruk yöneticisine bağlanırken önden okuma geçersiz kılınır. Uygulama IBM MQ istemcisi olarak çalışmıyorsa, bu seçenekler yoksayılr.

Genel arama karakteri seçenekleri: Aşağıdaki seçenekler, MQSD ' nin ObjectString alanında sağlanan dizgide joker karakterlerin nasıl yorumlanacağını denetler. Bu seçeneklerden yalnızca birini belirleyebilirsiniz. MQSO_ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, bu genel arama karakteri seçenekleri değiştirilemez. MQSO_RESUME kullanılarak bir MQSUB çağırısından dönüş olduğunda, uygun genel arama karakteri seçeneği ayarlanır.

MQSO_WILDCARD_CHAR

Genel arama karakterleri yalnızca konu dizesi içindeki karakterlerde çalışır.

MQSO_WILDCARD_CHAR tarafından tanımlanan davranış aşağıdaki çizelgede gösterilir.

Çizelge 528. Joker karakterlerin yorumlanması	
Özel Karakter	Davranış
Eğik Çizgi (/)	Anlamı yok, sadece başka bir karakter.
Yıldız işareti (*)	Joker karakter, sıfır ya da daha fazla karakter
Soru işareti (?)	Joker karakter, 1 karakter
Yüzde işareti (%)	(*), (?) ya da (%) karakterlerinin bir dizgide kullanılmasına ve özel karakter olarak yorumlanmamasına izin vermek için çıkış karakteri; örneğin, (% *), (%?) ya da (%%).

Örneğin, şu konuda yayınlama:

```
/level0/level1/level2/level3/level4
```

aboneleri aşağıdaki konuları kullanarak eşleştirir:

```
*  
/*  
/ level0/level1/level2/level3/*  
/ level0/level1/*/level3/level4  
/ level0/level1/le?el2/level3/level4
```

Not: Bu joker karakter kullanımı, yayınlama/abone olma için MQRFH1 biçimli iletileri kullanırken IBM MQ V6 ve WebSphere MB V6 ' da sağlanan anlamı tam olarak sağlar. Bunun yeni yazılan uygulamalar için kullanılmaması ve yalnızca daha önce o sürüm için çalışan uygulamalar için kullanılması ve MQSO_WILDCARD_TOPIC içinde açıklandığı gibi varsayılan genel arama karakteri davranışını kullanacak şekilde değiştirilmemesi önerilir.

MQSO_WILDCARD_TOPIC

Joker karakterler yalnızca konu dizesi içindeki konu öğelerinde çalışır. Hiçbiri seçilmezse, bu varsayılan davranıştır.

MQSO_WILDCARD_TOPIC için gereken davranış aşağıdaki tabloda gösterilir:

Özel Karakter	Davranış
(/)	Konu düzeyi ayırıcısı
Sayı işareti (#)	Genel arama karakteri: birden çok konu düzeyi
Artı işareti (+)	Genel arama karakteri: tek konu düzeyi

Notlar:

(+) ve (#), bir konu düzeyinde diğer karakterlerle (kendileri dahil) karıştırılırsa genel arama karakteri olarak kabul edilmez. Aşağıdaki dizgide, (#) ve (+) karakterleri olağan karakterler olarak işlenir.

```
level0/level1/#+/level3/level#
```

Örneğin, şu konuda yayınlama:

```
/level0/level1/level2/level3/level4
```

aboneleri aşağıdaki konuları kullanarak eşleştirir:

```
#  
/#  
/ level0/level1/level2/level3/#  
/ level0/level1+/level3/level4
```

Diğer seçenekler: Aşağıdaki seçenekler, abonelik yerine API çağrısını nasıl yayınlanacağına denetler. MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılıyorsa, bu seçenekler değişmez. Daha fazla ayrıntı için bkz. [“AlternateUserId \(MQCHAR12\) for MQSD” sayfa 575](#).

MQSO_ALTERNATE_USER_AUTHORITY

AlternateUser(Alternatif Kullanıcı) alanı, bu MQSUB çağrısının geçerliliğini denetlemek için kullanılacak bir kullanıcı kimliği içerir. Bu AlternateUserkimliği, uygulamanın çalıştırıldığı kullanıcı kimliğinin bunu yapma yetkisi olup olmadığına bakılmaksızın, belirtilen erişim seçenekleriyle nesneyi açma yetkisine sahip ise, çağrı başarılı olabilir.

MQSO_SET_CORREL_ID

Abonelik, *SubCorrelId* alanında sağlanan ilinti tanıtıcısını kullanmaktır. Bu seçenek belirtilmezse, abonelik sırasında kuyruk yöneticisi tarafından otomatik olarak bir ilinti tanıtıcısı yaratılır ve *SubCorrelId* alanında uygulamaya döndürülür. Daha fazla bilgi için bkz. [“MQSD için SubCorrelTanıtıcısı \(MQBYTE24\)” sayfa 577](#).

Bu seçenek MQSO_MANAGED ile birleştirilemez.

MQSO_SET_IDENTITY_CONTEXT

Abonelik, *PubAccountingToken* ve *PubApplIdentityData* alanlarında sağlanan muhasebe simgesini ve uygulama kimlik verilerini kullanmaktır.

Bu seçenek belirtilirse, MQSO_MANAGED seçeneğinin de kullanıldığı durumlar dışında, hedef kuyruğa MQOO_SET_IDENTITY_CONTEXT ile MQOPEN çağrısı kullanılarak erişildiği gibi aynı yetki denetimi gerçekleştirilir; bu durumda hedef kuyruқта yetki denetimi yoktur.

Bu seçenek belirtilmezse, bu aboneye gönderilen yayınlarla ilişkili varsayılan bağlam bilgileri aşağıdaki gibi olur:

<i>Çizelge 530. Bu aboneye gönderilen yayınlar için varsayılan bağlam bilgileri</i>	
MQMD 'deki Alan	Kullanılan değer
<i>UserIdentifier</i>	Abonelik yapıldığında abonelik ile ilişkilendirilen kullanıcı kimliği.
<i>AccountingToken</i>	Olanaklıysa ortamdan saptanır; yoksa MQACT_NONE olarak ayarlayın.
<i>ApplIdentityData</i>	Boşluklara ayarla

Bu seçenek yalnızca MQSO_CREATE ve MQSO ALTER ile geçerlidir. MQSO_RESUME ile kullanılırsa, *PubAccountingToken* ve *PubApplIdentityData* alanları yoksayılır, bu nedenle bu seçeneğin bir etkisi yoktur.

Daha önce abonelik tarafından sağlanan kimlik bağlamı bilgilerinin bulunduğu bu seçenek kullanılmadan bir abonelik değiştirilirse, değiştirilen abonelik için varsayılan bağlam bilgileri oluşturulur.

Farklı kullanıcı kimliklerinin MQSO_ANY_USERID seçeneğiyle kullanılmasına izin veren bir abonelik farklı bir kullanıcı kimliğiyle sürdürüldüğünde, aboneliğe sahip yeni kullanıcı kimliği için varsayılan kimlik bağlamı oluşturulur ve yeni kimlik bağlamını içeren sonraki yayınlar teslim edilir.

MQSO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturma durumundaysa MQSUB çağrısı başarısız olur. z/OS sistemlerinde, bir CICS ya da IMS uygulaması için bu seçenek, bağlantı susturma durumundaysa MQSUB çağrısına da başarısız olmasına neden olur.

MQSD için ObjectName (MQCHAR48)

Bu, yerel kuyruk yöneticisinde tanımlandığı şekilde, konu nesnesinin adıdır.

Ad aşağıdaki karakterleri içerebilir:

- Büyük harf alfabetik karakterler (A-Z)
- Küçük harf alfabetik karakterler (a-z)
- Sayısal basamaklar (0-9)
- Nokta (.), eğik çizgi (/), altçizgi (_), yüzde (%)

Ad başta ya da gömülü boşluklar içermemelidir, ancak sonda boşluklar içerebilir. Addaki önemli verilerin sonunu göstermek için boş bir karakter kullanın; bu karakteri izleyen karakterler ve boş değer boşluk olarak değerlendirilir. Belirtilen ortamlarda aşağıdaki kısıtlamalar geçerlidir:

- EBCDIC Katakana kullanan sistemlerde küçük harfli karakterler kullanılamaz.
- z/OS'ta:
 - Alt çizgiyle başlayan ya da biten adlardan kaçınınız; bunlar işlemler ve denetim panoları tarafından işlenemez.
 - Yüzde karakterinin RACF için özel bir anlamı vardır. Dış güvenlik yöneticisi olarak RACF kullanılırsa, adlar yüzdeyi içermemelidir. Varsa, RACF sosyal tanımları kullanıldığında bu adlar güvenlik denetimlerine dahil edilmez.
- IBM üzerinde, komutlarda belirtildiğinde küçük harf, eğik çizgi ya da yüzde içeren adlar tırnak içine alınmalıdır. Bu tırnak işaretleri, yapılarda alan olarak ya da çağrılarda parametre olarak oluşan adlar için belirlenmemelidir.

ObjectName , tam konu adını oluşturmak için kullanılır.

Tam konu adı iki farklı alandan oluşturulabilir: *ObjectName* ve *ObjectString*. Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.

ObjectName alanıyla tanıtilan nesne bulunamazsa, *ObjectString* içinde belirtilmiş bir dizgi olsa bile, çağrı MQRC_UNKOWN_OBJECT_NAME neden koduyla başarısız olur.

MQSO_RESUME seçeneği kullanılarak bir MQSUB çağrısından dönüş yapılıyorsa, bu alan değişmez.

Bu alanın uzunluğu MQ_TOPIC_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQSO_ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, abone olunan konu nesnesinin adı değiştirilemez. Bu alan ve *ObjectString* alanı atlanabilir. Sağlanırsa, aynı tam konu adına çözümleri gerekir. Çağrılmazsa, çağrı MQRC_TOPIC_NOT_ALTERABLE ile başarısız olur.

AlternateUserId (MQCHAR12) for MQSD

MQSO_ALTERNATE_USER_AUTHORITY belirtilirse, bu alan, uygulamanın çalışmakta olduğu kullanıcı kimliği yerine, abonelik ve hedef kuyruğa ilişkin çıkışı (MQSUB çağrısının **Hobj** değiştirilmesinde belirtilir) denetlemek için kullanılan alternatif bir kullanıcı kimliği içerir.

Başarılı olursa, bu alanda belirtilen kullanıcı kimliği, uygulamanın çalışmakta olduğu kullanıcı kimliği yerine, abonelik sahibi kullanıcı kimliği olarak kaydedilir.

MQSO_ALTERNATE_USER_AUTHORITY belirtilirse ve bu alan ilk boş karakter ya da alanın sonuna kadar tamamen boşsa, abonelik ancak belirtilen seçeneklerle ya da çıkış için hedef kuyruk ile bu konuya abone olmak için kullanıcı yetkilendirmesi gerekmediğinde başarılı olabilir.

MQSO_ALTERNATE_USER_AUTHORITY belirtilmezse, bu alan yoksayıdır.

Belirtilen ortamlarda aşağıdaki farklılıklar vardır:

- z/OS işletim sistemlerinde, aboneliğe ilişkin yetkilendirmeyi denetlemek için yalnızca AlternateUserkimliğinin ilk 8 karakteri kullanılır. Ancak, yürürlükteki kullanıcı kimliğinin bu diğer kullanıcı kimliğini belirtme yetkisi olmalıdır; bu denetim için diğer kullanıcı kimliğinin 12 karakterinin tümü de kullanılır. Kullanıcı kimliği yalnızca dış güvenlik yöneticisi tarafından izin verilen karakterleri içermelidir.

MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılıyorsa, bu alan değişmez.

Bu bir giriş alanıdır. Bu alanın uzunluğu MQ_USER_ID_LENGTH tarafından verilir. Bu alanın ilk değeri C ' deki boş dizgi ve diğer programlama dillerindeki 12 boş karakterdir.

MQSD için AlternateSecurityTanıtıcısı (MQBYTE40)

Bu, uygun yetki denetimlerinin gerçekleştirilmesine izin vermek için yetkilendirme hizmetine AlternateUserkimliği ile iletilen bir güvenlik tanıtıcısıdır.

AlternateSecurityTanıtıcısı yalnızca MQSO_ALTERNATE_USER_AUTHORITY belirtildiyse ve AlternateUserId alanı ilk boş karaktere ya da alanın sonuna kadar tamamen boş değilse kullanılır.

MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılıyorsa, bu alan değişmez.

Ek bilgi için MQOD veri tipindeki [“MQOD için AlternateSecurityTanıtıcısı \(MQBYTE40\)” sayfa 487](#) tanımına bakın.

MQSD için SubExpiry (MQLONG)

Bu, abonelik süresinin sona ermesinden sonra saniyenin onda biri olarak ifade edilen süredir. Bu aralık geçtikten sonra başka hiçbir yayın bu abonelikte eşleşmeyecek. Abonelik süresi biter bitmez, yayınlar artık kuyruğa gönderilmez. Ancak, zaten var olan yayınlar hiçbir şekilde etkilenmez. *SubExpiry* ' i yayın süre bitimi üzerinde bir etkisi yoktur.

Şu özel değer tanınır:

MQEI_UNLIMITED

Aboneliğin sınırsız bir süre sonu vardır.

MQSO ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, aboneliğin süre bitimi değiştirilebilir.

MQSO RESUME seçeneği kullanılarak bir MQSUB çağrısından dönüşte bu alan, aboneliğin özgün süre bitimine ayarlanır, kalan süre bitimine ayarlanmaz.

MQSD için ObjectString (MQCHARV)

Bu, kullanılacak uzun nesne adıdır.

ObjectString , Tam konu adını oluşturmak için kullanılır.

Tam konu adı iki farklı alandan oluşturulabilir: *ObjectName* ve *ObjectString*. Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.

ObjectString uzunluk üst sınırı 10240 'tır.

ObjectString doğru belirtilmezse, MQCHARV yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı MQRC_OBJECT_STRING_ERROR neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu yapıdaki alanların başlangıç değerleri, MQCHARV yapısındaki değerlerle aynıdır.

ObjectString içinde genel arama karakterleri varsa, bu genel arama karakterlerinin yorumlanması, MQSD ' nin Seçenekler alanında belirtilen genel arama karakteri seçenekleri kullanılarak denetlenebilir.

MQSO RESUME seçeneği kullanılarak bir MQSUB çağrısından dönüş yapılıyorsa, bu alan değişmez. Bir arabellek sağlanırsa, *ResObjectString* alanında kullanılan tam konu adı döndürülür.

MQSO ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, abone olunan konu nesnesinin uzun adı değiştirilemez. Bu alan ve *ObjectName* alanı atlanabilir. Sağlanırsa, bunların aynı tam konu adına çözülmesi gerekir ya da çağrı MQRC_TOPIC_NOT_ALTERABLE ile başarısız olur.

MQSD için SubName (MQCHARV)

Bu, abonelik adını belirtir. Bu alan yalnızca *Options* MQSO DURABLE seçeneğini belirtiyorsa, ancak sağlanırsa, kuyruk yöneticisi tarafından MQSO_NON_DURABLE için de kullanılır.

Belirtilirse, *SubName* kuyruk yöneticisi içinde benzersiz olmalıdır; bu, aboneliği tanımlamak için kullanılan yöntemdir.

SubName uzunluk üst sınırı 10240 'tır.

Bu alan iki amaca hizmet eder. Bir MQSO DURABLE aboneliği için, abonelik yaratıldıktan sonra (MQCO_KEEP_SUB seçeneğini kullanarak) aboneliği kapattıysanız ya da kuyruk yöneticisiyle bağlantınız kesildiyse, aboneliği devam ettirebilmeniz için bu alanı kullanırsınız. Bu, MQSO RESUME seçeneğiyle MQSUB çağrısı kullanılarak yapılır. Ayrıca, DISPLAY SBSTATUS içindeki SUBID alanındaki aboneliklerin yönetim görünümünde de görüntülenir.

SubName yanlış belirtilirse, MQCHARV yapısının nasıl kullanılacağına ilişkin açıklamaya göre, gerekli olduğunda (yani *SubName*) dışarıda bırakılır. *VSLength* sıfır) ya da uzunluk üst sınırını aşarsa, çağrı başarısız olur; neden kodu MQRC_SUB_NAME_ERROR.

Bu bir giriş alanıdır. Bu yapıdaki alanların başlangıç değerleri, MQCHARV yapısındaki değerlerle aynıdır.

MQSO ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, abonelik adı, gönderme yapılan aboneliği bulmak için kullanılan tanıttıcı alan olduğundan değiştirilemez. MQSO RESUME seçeneğiyle bir MQSUB çağrısının çıkışında değiştirilmez.

MQSD için SubUserVerileri (MQCHARV)

Bu, abonelik kullanıcı verilerini belirtir. Bu alandaki abonelikte sağlanan veriler, bu aboneliğe gönderilen her yayının MQSubUserveri iletisi özelliği olarak içerilir.

SubUserData uzunluk üst sınırı 10240 'tır.

SubUserData yanlış belirtilirse, MQCHARV yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı başarısız olur; neden kodu MQRC_SUB_USER_DATA_ERROR.

Bu bir giriş alanıdır. Bu yapıdaki alanların başlangıç değerleri, MQCHARV yapısındaki değerlerle aynıdır.

MQSO ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, abonelik kullanıcı verileri değiştirilebilir.

Bir arabellek sağlandıysa ve *VSBuflen* içinde pozitif bir arabellek uzunluğu varsa, MQSO_RESUME seçeneği kullanılarak bir MQSUB çağrısının çıkışında bu değişken uzunluklu alan döndürülür. Çağrıda arabellek sağlanmazsa, MQCHARV ' nin *VSLength* alanında yalnızca abonelik kullanıcı tarihinin uzunluğu döndürülür. Sağlanan arabellek, alanı döndürmek için gereken alandan küçükse, sağlanan arabellekte yalnızca *VSBuflen* byte döndürülür.

MQSD için SubCorrelTanıtıcısı (MQBYTE24)

Bu alan, bu abonelikte eşleşen tüm yayınlar için ortak bir ilinti tanıtıcısı içerir.



Uyarı: bir ilinti tanıtıcısı yalnızca bir yayınlama/abone olma kümesindeki kuyruk yöneticileri arasında iletilebilir, bir sıradüzen arasında değil.

Bu abonelikte eşleşmesi için gönderilen tüm yayınlar, ileti tanımlayıcısında bu ilinti tanıtıcısını içerir. Birden çok abonelik yayınlarını aynı kuyruktan alıyorsa, ilinti tanıtıcısı temelinde MQGET kullanılması yalnızca belirli bir aboneliğe ilişkin yayınların elde edilmesine izin verir. Bu ilinti tanıtıcısı kuyruk yöneticisi ya da kullanıcı tarafından oluşturulabilir.

MQSO_SET_CORREL_ID seçeneği belirtilmezse, ilinti tanıtıcısı kuyruk yöneticisi tarafından üretilir ve bu alan, bu abonelik için yayınlanan her iletide ayarlanacak ilinti tanıtıcısını içeren bir çıkış alanıdır. Oluşturulan ilinti tanıtıcısı, 4 baytlık bir ürün tanıtıcısından (ASCII ya da EBCDIC ' de AMQX ya da CSQM) ve ardından benzersiz bir dizginin ürüne özgü somutlamasından oluşur.

MQSO_SET_CORREL_ID seçeneği belirtilirse, ilinti tanıtıcısı kullanıcı tarafından üretilir ve bu alan, bu abonelik için her yayında ayarlanacak ilinti tanıtıcısını içeren bir giriş alanıdır. Bu durumda, alan MQCI_NONE içeriyorsa, bu abonelik için yayınlanan her iletide ayarlanan ilinti tanıtıcısı, iletinin özgün konmasıyla yaratılan ilinti tanıtıcısıdır.

MQSO_GROUP_SUB seçeneği belirtilirse ve belirtilen ilinti tanıtıcısı, aynı kuyruk ve çakışan bir konu dizgisi kullanılarak var olan bir gruplanmış abonelikte aynıysa, yayının bir kopyasıyla yalnızca gruptaki en önemli abonelik sağlanır.

Bu alanın uzunluğu MQ_CORREL_ID_LENGTH tarafından verilir. Bu alanın ilk değeri MQCI_NONE.

MQSO ALTER seçeneğini kullanarak var olan bir aboneliği değiştiriyorsanız ve bu alan bir giriş alanıysa, abonelik gruplanmış bir abonelik olmadığı sürece abonelik ilinti tanıtıcısı değiştirilebilir; yani, abonelik ilinti tanıtıcısı MQSO_GROUP_SUB seçeneği kullanılarak yaratılmıştır; bu durumda, abonelik ilinti tanıtıcısı değiştirilemez.

MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılarak, bu alan aboneliğin yürürlükteki ilinti tanıtıcısına ayarlanır.

MQSD için PubPriority (MQLONG)

Bu değer, bu abonelikte eşleşen tüm yayın iletilerinin İleti Tanımlayıcısı 'nın (MQMD) *Priority* alanında olacak değerdir. MQMD ' deki *Priority* alanıyla ilgili daha fazla bilgi için bkz. "[MQMD için Öncelik \(MQLONG\)](#)" sayfa 449.

Değer sıfırdan büyük ya da sıfıra eşit olmalıdır; sıfır en düşük önceliklidir. Aşağıdaki özel değerler de kullanılabilir:

MQPRI_PRIORITY_AS_Q_DEF

MQSUB çağrısının *Hobj* alanında bir abonelik kuyruğu sağlandığında ve yönetilen bir tanıtıcı olmadığında, iletinin önceliği bu kuyruğun **DefPriority** özniteliğinden alınır. Kuyruk bir küme kuyruğuysa ya da kuyruk adı çözüm yolunda birden çok tanımlama varsa, yayın iletisi "MQMD için Öncelik (MQLONG)" sayfa 449 için açıklandığı gibi kuyruğa konduğunda öncelik belirlenir.

MQSUB çağrısı yönetilen bir tanıtıcı kullanıyorsa, iletinin önceliği, abone olunan konuyla ilişkili model kuyruğunun **DefPriority** özniteliğinden alınır.

MQPRI_PRIORITY_AS_PUBLISHED

İletinin önceliği, özgün yayının önceliğidir. Bu, alanın ilk değeridir.

MQSO_ALTER seçeneğini kullanarak var olan bir abonelik değiştiriliyorsa, ilerideki yayın iletilerinin *Priority* değeri değiştirilebilir.

MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılırsa, bu alan abonelik için kullanılmakta olan yürürlükteki önceliğe ayarlanır.

PubAccountingMQSD için simge (MQBYTE32)

Bu değer, bu abonelikte eşleşen tüm yayın iletilerinin İleti Tanımlayıcısı'nın (MQMD) *AccountingToken* alanında olacak değerdir. *AccountingToken*, iletinin kimlik bağlamının bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. İleti bağlamı. MQMD 'deki *AccountingToken* alanıyla ilgili daha fazla bilgi için bkz. "MQMD için AccountingToken (MQBYTE32)" sayfa 456

PubAccountingToken alanı için aşağıdaki özel değeri kullanabilirsiniz:

MQACT_NONE

Hesap simgesi belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQACT_NONE_ARRAY değişmezi de tanımlanır; bu değer MQACT_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

MQSO_SET_IDENTITY_CONTEXT seçeneği belirtilmezse, hesap simgesi kuyruk yöneticisi tarafından varsayılan bağlam bilgileri olarak oluşturulur ve bu alan, bu abonelik için yayınlanan her iletide ayarlanacak *AccountingToken* değerini içeren bir çıkış alanıdır.

MQSO_SET_IDENTITY_CONTEXT seçeneği belirtilirse, hesap simgesi kullanıcı tarafından üretilir ve bu alan, bu abonelik için her yayında ayarlanacak *AccountingToken* değerini içeren bir giriş alanıdır.

Bu alanın uzunluğu MQ_ACCOUNTING_TOKEN_LENGTH tarafından verilir. Bu alanın ilk değeri MQACT_NONE.

MQSO_ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, ilerideki yayın iletilerinde *AccountingToken* değeri değiştirilebilir.

MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılırsa, bu alan abonelik için kullanılmakta olan *AccountingToken* olarak ayarlanır.

MQSD için PubApplIdentityData (MQCHAR32)

Bu değer, bu abonelikte eşleşen tüm yayın iletilerinin iletisi tanımlayıcısının (MQMD) *ApplIdentityData* alanında bulunan değerdir. *ApplIdentityData*, iletinin kimlik bağlamının bir parçasıdır. İletisi bağlamıyla ilgili daha fazla bilgi için bkz. İletisi bağlamı. MQMD 'deki *ApplIdentityData* alanıyla ilgili daha fazla bilgi için bkz. "MQMD için ApplIdentityVerileri (MQCHAR32)" sayfa 458

MQSO_SET_IDENTITY_CONTEXT seçeneği belirtilmezse, varsayılan bağlam bilgileri olarak, bu abonelik için yayınlanan her iletide ayarlanan *ApplIdentityData* boş olur.

MQSO_SET_IDENTITY_CONTEXT seçeneği belirtilirse, *PubApplIdentityData* kullanıcı tarafından üretilir ve bu alan, bu abonelik için her yayında ayarlanacak *ApplIdentityData* değerini içeren bir giriş alanıdır.

Bu alanın uzunluğu MQ_APPL_IDENTITY_DATA_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerinde 32 boş karakterdir.

MQSO_ALTER seçeneğini kullanarak var olan bir abonelik değiştiriliyorsa, ilerideki yayın iletilerinin *ApplIdentityData* değeri değiştirilebilir.

MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılırsa, bu alan abonelik için kullanılmakta olan *ApplIdentityData* olarak ayarlanır.

MQSD için SelectionString (MQCHARV)

Bu, bir konudan gelen iletilere abone olurken kullanılan seçim ölçütlerini sağlamak için kullanılan dizidir.

Bu değişken uzunluklu alan, bir arabellek sağlandıysa, MQSO_RESUME seçeneği kullanılarak bir MQSUB çağrısının çıkışında döndürülür ve VSBufSize içinde artı bir arabellek uzunluğu da vardır. Çağrıda arabellek sağlanmazsa, MQCHARV 'nin VSL kalınlık alanında yalnızca seçim dizgisinin uzunluğu döndürülür. Sağlanan arabellek alanı döndürmek için gereken alandan küçükse, sağlanan arabellekte yalnızca VSBufSize byte döndürülür.

SelectionString yanlış belirtilirse, “MQCHARV-Değişken Uzunluklu Dizgi” sayfa 294 yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı MQRC_SELECTION_STRING_ERROR neden koduyla başarısız olur.

SelectionString kullanımı, [Selectors](#) içinde açıklanmaktadır.

MQSD için SubLevel (MQLONG)

Bu, abonelik ile ilişkili düzeydir. Yayınlar, yalnızca yayın sırasında kullanılan PubLevel değerinden küçük ya da bu değere eşit olan en yüksek SubLevel değerine sahip abonelikler kümesinde yer alıyorsa bu aboneliğe teslim edilir. Ancak, bir yayın alıkonduysa, PubLevel 1 'de yeniden yayınlandığından, artık daha yüksek düzeylerde aboneler tarafından kullanılamaz.

Değer, sıfır ile 9 aralığında olmalıdır. Sıfır en düşük düzeydir.

Bu alanın ilk değeri 1 'dir.

Daha fazla bilgi için bkz. [Intercepting publications](#).

MQSO_ALTER seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, SubLevel değiştirilemez.

MQSO_PUBLICATIONS_ON_REQUEST seçeneğiyle 1 'den büyük bir değerle SubLevel birleştirilmesine izin verilmez.

MQSO_RESUME kullanılarak bir MQSUB çağrısından dönüş yapılırsa, bu alan abonelik için kullanılan yürürlükteki düzeye ayarlanır.

MQSD için ResObjectDizesi (MQCHARV)

Bu, kuyruk yöneticisi *ObjectName* içinde sağlanan adı çözdükten sonra uzun nesne adıdır.

Uzun nesne adı *ObjectString* içinde sağlanıyorsa ve *ObjectName* içinde hiçbir şey sağlanmıyorsa, bu alanda döndürülen değer *ObjectString* içinde sağlananla aynıdır.

Bu alan atlanırsa (*ResObjectString.VSBufSize* sıfır olur), *ResObjectString* döndürülmez, ancak uzunluk *ResObjectString.VSLength* içinde döndürülür. Uzunluk tam *ResObjectString* dizgisinden kısaysa, kesilir ve sağlanan uzunluğa sığabilecek en sağdaki karakterlerin sayısını döndürür.

ResObjectString yanlış belirtilirse, [MQCHARV](#) yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı MQRC_RES_OBJECT_STRING_ERROR neden koduyla başarısız olur.

MQSMPO-İleti özelliği seçeneklerini ayarla

MQSMPO yapısı, uygulamaların, ileti özelliklerinin nasıl ayarlanacağını denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, **MQSETMP** çağrısındaki bir giriş parametresidir.

Kullanılabilirlik

Tüm IBM MQ sistemleri ve IBM MQ istemcileri.

Karakter kümesi ve kodlama

MQSMPO içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında olmalıdır (**MQENC_NATIVE**).

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 531. MQSMPO içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQSMPO_STRUC_ID	'SMPO'
Sürüm (yapı sürümü numarası)	MQSMPO_VERSION_1	1
Seçenekler (seçenekler)	MQSMPO_NONE	0
ValueEncoding (özellik değeri kodlaması)	MQENC_NATIVE	Ortama bağlıdır
ValueCCSID (özellik değeri karakter takımı)	MQCCSI_APPL	-3

Notlar:

- Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.
- C programlama dilinde, makro değişkeniMQSMPO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQSMPO MySMPO = {MQSMPO_DEFAULT};
```

Dil bildirimleri

MQSMPO için C bildirimini

```
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 */
};
```

MQSMPO için COBOL bildirimini

```
** 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.

```

MQSMPO için PL/I bildirimi

```

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 */

```

MQSMPO için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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)

```

MQSMPO için StrucId (MQCHAR4)

Bu, ileti özellik seçeneklerini ayarla yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQSMPO_STRUC_ID.

Değer şöyle olmalıdır:

MQSMPO_STRUC_ID

İleti özellik seçeneklerini ayarla yapısına ilişkin tanıttıcı.

C programlama dili için, **MQSMPO_STRUC_ID_ARRAY** sabiti de tanımlanır. Bu, **MQSMPO_STRUC_ID** ile aynı değere sahiptir, ancak bir dizgi yerine bir karakter dizisidir.

MQSMPO için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQSMPO_VERSION_1

Version-1 ileti özelliği seçenekleri yapısını ayarlayın.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQSMPO_CURRENT_VERSION

İleti ayarlama özelliği seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **MQSMPO_VERSION_1**.

MQSMPO için Seçenekler (MQLONG)

Konum seçenekleri

Aşağıdaki seçenekler, özellik imleciyle karşılaştırıldığında özelliğin görel konumuyla ilgilidir:

MQSMPO_SET_FIRST

Belirtilen adla eşleşen ilk özelliğin değerini ayarlar ya da yoksa, eşleşen bir sıradüzene sahip diğer tüm özelliklerden sonra yeni bir özellik ekler.

MQSMPO_SET_PROP_UNDER_CURSOR

Özellik imlecinin gösterdiği özelliğin değerini belirler. Özellik imlecinin gösterdiği özellik, MQIMPO_INQ_FIRST ya da MQIMPO_INQ_NEXT seçeneği kullanılarak en son sorulmuş özelliktir.

İleti tanıtıcısı bir MQGET çağrısında yeniden kullanıldığında ya da ileti tanıtıcısı bir MQPUT çağrısındaki MQGMO ya da MQPMO yapısının *MsgHandle* alanında belirtildiğinde özellik geçici çizelgesi ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında ya da özellik imleci tarafından özellik imleci silindiğinde bu seçenek kullanılırsa, çağrı tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_PROPERTY_NOT_ALLOWED ile başarısız olur.

MQSMPO_SET_PROP_BEFORE_CURSOR

Özellik imlecinin gösterdiği özellikten önce yeni bir özellik belirler. Özellik imlecinin gösterdiği özellik, MQIMPO_INQ_FIRST ya da MQIMPO_INQ_NEXT seçeneği kullanılarak en son sorulmuş özelliktir.

İleti tanıtıcısı bir MQGET çağrısında yeniden kullanıldığında ya da ileti tanıtıcısı bir MQPUT çağrısındaki MQGMO ya da MQPMO yapısının *MsgHandle* alanında belirtildiğinde özellik geçici çizelgesi ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında ya da özellik imleci tarafından özellik imleci silindiğinde bu seçenek kullanılırsa, çağrı tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_PROPERTY_NOT_ALLOWED ile başarısız olur.

MQSMPO_SET_PROP_AFTER_CURSOR

Özellik imlecinin gösterdiği özellikten sonra yeni bir özellik belirler. Özellik imlecinin gösterdiği özellik, MQIMPO_INQ_FIRST ya da MQIMPO_INQ_NEXT seçeneği kullanılarak en son sorulmuş özelliktir.

İleti tanıtıcısı bir MQGET çağrısında yeniden kullanıldığında ya da ileti tanıtıcısı bir MQPUT çağrısındaki MQGMO ya da MQPMO yapısının *MsgHandle* alanında belirtildiğinde özellik geçici çizelgesi ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında ya da özellik imleci tarafından özellik imleci silindiğinde bu seçenek kullanılırsa, çağrı tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_PROPERTY_NOT_ALLOWED ile başarısız olur.

MQSMPO_APPEND_ÖZELLİĞİ

Eşleşen bir sıradüzene sahip diğer tüm özelliklerden sonra yeni bir özelliğin eklenmesine neden olur. Belirtilen adla eşleşen en az bir özellik varsa, bu özellik listesinin sonundan sonra yeni bir özellik eklenir.

Bu seçenek, aynı ada sahip özelliklerin bir listesinin yaratılmasına olanak sağlar.

Açıklanan seçeneklerin hiçbirine gerek duymazsanız, aşağıdaki seçeneği kullanın:

MQSMPO_NONE

Seçenek belirtilmedi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQSMPO_SET_FIRST.

MQSMPO için ValueEncoding (MQLONG)

Değer sayıysa ayarlanacak özellik değerinin kodlaması.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **MQENC_NATIVE**.

MQSMPO için ValueCCSID (MQLONG)

Değer bir karakter dizisiyse ayarlanacak özellik değerinin karakter kümesi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **MQCCSI_APPL**.

MQSRO-Abonelik isteği seçenekleri

MQSRO yapısı, uygulamanın abonelik isteğinin nasıl yapılacağını denetleyen seçenekleri belirtmesini sağlar. Yapı, MQSUBRQ çağrısında bir giriş/çıkış değiştirgesidir.

Kullanılabilirlik

MQSRO yapısı aşağıdaki altyapılarda bulunur:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

ve bu sistemlere bağlı IBM MQ MQI clients için.

Sürüm

MQSRO ' nun yürürlükteki sürümü: MQSRO_VERSION_1.

Karakter kümesi ve kodlama

MQSRO ' daki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında yer almalıdır. Ancak, uygulama bir MQ MQI istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQSRO_STRUC_ID	' SRO- '
<u>Sürüm</u> (yapı sürümü numarası)	MQSRO_VERSION_1	1
<u>Seçenekler</u> (seçenekler)	MQSRO_NONE	0
<u>NumPubs</u> (yayın sayısı)	Yok	0

Notlar:

1. - simgesi, tek bir boş karakteri temsil eder.
2. C programlama dilinde, makro değişkeniMQSRO_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir:

```
MQSRO MySRO = {MQSRO_DEFAULT};
```

Dil bildirimleri

MQSRO için C bildirimi

```
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 */
};
```

MQSRO için COBOL bildirimi

```
** 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.
```

MQSRO için PL/I bildirimi

```
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 */
```

MQSRO için High Level Assembler bildirimi

```
MQSRO          DSECT
MQSRO_STRUCID  DS    CL4    Structure identifier
MQSRO_VERSION  DS    F      Structure version number
MQSRO_OPTIONS  DS    F      Options that control the action of MQSUBRQ
MQSRO_NUMPUBS  DS    F      Number of publications sent
*
MQSRO_LENGTH   EQU    *-MQSRO
               ORG    MQSRO
MQSRO_AREA     DS    CL(MQSRO_LENGTH)
```

MQSRO için StrucId (MQCHAR4)

Bu, abonelik isteği seçenekleri yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQSRO_STRUC_ID.

Değer şöyle olmalıdır:

MQSRO_STRUC_ID

Abonelik isteği seçenekleri yapısına ilişkin tanıtıcı.

C programlama dili için, MQSRO_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQSRO_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQSRO için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

MQSRO_VERSION_1

Version-1 Abonelik İsteği Seçenekleri yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQSRO_CURRENT_VERSION

Abonelik İsteği Seçenekleri yapısının geçerli sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MQSRO_VERSION_1' dir.

MQSRO için Seçenekler (MQLONG)

Aşağıdaki seçeneklerden biri belirtilmelidir. Tek bir seçenek belirlenebilir.

MQSRO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturma durumundaysa MQSUBRQ çağrısı başarısız olur. z/OS sistemlerinde, bir CICS ya da IMS uygulaması için bu seçenek, bağlantı susturma durumundaysa MQSUBRQ çağrısına da başarısız olmasına neden olur.

Varsayılan seçenek: Önceden açıklanan seçenek gerekli değilse, aşağıdaki seçenek kullanılmalıdır:

MQSRO_NONE

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar.

MQSRO_NONE, program belgelerine yardımcı olur. Değeri sıfır olduğu için bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmasa da, bu kullanım algılanamaz.

MQSRO için NumPubs (MQLONG)

Bu, bu çağrı sonucunda abonelik kuyruğuna gönderilen yayınların sayısını belirtmek için uygulamaya döndürülen bir çıkış alanıdır. Bu çağrı sonucunda bu sayıda yayın gönderilmesine rağmen, özellikle kalıcı olmayan iletiler olduğunda, uygulamanın bu kadar çok iletiyi alabileceğine dair bir garanti yoktur.

Konu bir genel arama karakteri içeriyorsa, birden çok yayın olabilir. *Hsub* tarafından temsil edilen abonelik oluşturulduğunda konu dizisinde genel arama karakteri yoksa, bu çağrı sonucunda en çok bir yayın gönderilir.

MQSTS-Durum raporlama yapısı

MQSTS yapısı, MQSTAT komutunun bir çıkış değiştirgesidir. Durum bilgilerini almak için MQSTAT komutu kullanılır. Bu bilgiler bir MQSTS yapısında döndürülür.

Karakter kümesi ve kodlama

MQSTS içindeki karakter verileri yerel kuyruk yöneticisinin karakter kümesinde; bu, *CodedCharSetId* kuyruk yöneticisi özneteliği tarafından verilir. MQSTS ' deki sayısal veriler yerli makine kodlamasında; bu, *Kodlamatarafından* verilir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

<i>Çizelge 532. MQSTS ' deki Alanlar</i>		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQSTS_STRUC_ID	' STAT↵ '
<u>Sürüm</u> (yapı sürümü numarası)	MQSTS_VERSION_1	1
<u>CompCode</u> (ilk hatanın tamamlanma kodu)	MQCC_OK	0
<u>Neden</u> (ilk hatanın neden kodu)	MQRC_NONE	0

Çizelge 532. MQSTS 'deki Alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>PutSuccessCount</u> (başarılı zamanuyumsuz koyma çağrısı sayısı)	Yok	0
<u>PutWarningCount</u> (uyarıları olan zamanuyumsuz koyma çağrılarının sayısı)	Yok	0
<u>PutFailureCount</u> (başarısız zamanuyumsuz koyma çağrısı sayısı)	Yok	0
<u>ObjectType</u> (arızalı nesnenin tipi)	MQOT_Q	1
<u>ObjectName</u> (başarısız olan nesnenin adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>ObjectQMgrAd</u> (arızalı nesnenin sahibi olan kuyruk yöneticisinin adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>ResolvedObjectName</u> (hedef kuyruğun çözülmüş adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>ResolvedQMgrAdı</u> (hedef kuyruk yöneticisinin çözümlenen adı)	Yok	Boş değerli dizilim ya da boşluklar
Not: Sürüm MQSTS_VERSION_2' den küçükse kalan alanlar yoksayılr.		
<u>ObjectString</u> (başarısız olan nesnenin uzun nesne adı)	MQCHARV_DEFAULT	{NULL,0,0,0,-3}
<u>SubName</u> (başarısız olan aboneliğin abonelik adı)	MQCHARV_DEFAULT	{NULL,0,0,0,-3}
<u>OpenOptions</u> (hatayla ilişkili açma seçenekleri)	Yok	0
<u>SubOptions</u> (hatayla ilişkili abonelik seçenekleri)	Yok	0
Notlar:		
<ol style="list-style-type: none"> 1. ↪ simgesi, tek bir boş karakteri temsil eder. 2. Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir. 3. C programlama dilinde, MQSTS_DEFAULT makro değişkeni çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere aşağıdaki şekilde kullanılabilir: 		
<pre>MQSTS MySTS = {MQSTS_DEFAULT};</pre>		

Dil bildirimleri

MQSTS için C bildirimini

```
typedef struct tagMQSTS MQSTS;
struct tagMQSTS {
    MQCHAR4   StrucId;           /* 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 */
};

```

MQSTS için COBOL bildirimi

```

** 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 **

```

MQSTS için PL/I bildirimi

```

dcl
 1 MQSTS based,
 3 StructId          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 */

```

MQSTS için High Level Assembler (Üst Düzey Çevirici) bildirimi

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)

İlgili başvurular

“MQSTAT-Durum bilgilerini al” sayfa 776

Durum bilgilerini almak için MQSTAT çağrısına bakın. Döndürülen durum bilgisi tipi, çağrıda belirtilen Tip değerine göre belirlenir.

MQSTS için StrucId (MQCHAR4)

Bu, durum raporlama yapısının yapı tanıtcısıdır. Her zaman bir giriş alanıdır. Değeri MQSTS_STRUC_ID.

Değer şöyle olmalıdır:

MQSTS_STRUC_ID

Durum raporlama yapısına ilişkin tanıttıcı.

C programlama dili için, MQSTS_STRUC_ID_ARRAY sabiti de tanımlanır. Bu, MQSTS_STRUC_ID ile aynı değere sahiptir, ancak bir dizgi yerine bir karakter dizisidir.

MQSTS için Sürüm (MQLONG)

Yapı sürümü numarası.

Değer aşağıdakilerden biri olmalıdır:

MQSTS_VERSION_1

Sürüm 1 durum raporlama yapısı.

MQSTS_VERSION_2

Sürüm 2 durum raporlama yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQSTS_CURRENT_VERSION

Durum raporlama yapısının geçerli sürümü. Geçerli sürüm: MQSTS_VERSION_2.

Version her zaman bir giriş alanıdır. İlk değeri MQSTS_VERSION_1.

MQSTS için CompCode (MQLONG)

Bildirilmekte olan işlemin tamamlanma kodu.

CompCode yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

Bu, ObjectName içinde belirtilen nesne üzerinde önceki bir zamanuyumsuz koyma işleminin sonucundaki tamamlanma kodudur.

MQSTAT_TYPE_RECONNECTION

Bağlantı yeniden bağlanıyorsa ya da yeniden bağlanamıyorsa, bağlantının yeniden bağlanmaya başlamasına neden olan tamamlanma kodu budur.

Bağlantı şu anda bağlıysa, değer MQCC_OK olur.

MQSTAT_TYPE_RECONNECTION_HATA

Bağlantı yeniden bağlanamazsa bu, yeniden bağlantının başarısız olmasına neden olan tamamlanma kodudur.

Bağlantı şu anda bağlıysa ya da yeniden bağlanıyorsa, değer MQCC_OK olur.

CompCode her zaman bir çıkış alanıdır. İlk değeri MQCC_OK.

MQSTS için neden (MQLONG)

Bildirilmekte olan işlemin neden kodu.

Reason yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

Bu, ObjectName içinde belirtilen nesne üzerinde önceki bir zamanuyumsuz koyma işleminden kaynaklanan neden kodudur.

MQSTAT_TYPE_RECONNECTION

Bağlantı yeniden bağlanıyorsa ya da yeniden bağlanamıyorsa, yeniden bağlantının yeniden bağlanmaya başlamasına neden olan neden kodu budur.

Bağlantı şu anda bağlıysa, değer MQRC_NONE olur.

MQSTAT_TYPE_RECONNECTION_HATA

Bağlantı yeniden bağlanamadıysa, bu, yeniden bağlantının başarısız olmasına neden olan neden kodudur.

Bağlantı şu anda bağlıysa ya da yeniden bağlanıyorsa, değer MQRC_NONE olur.

Reason bir çıkış alanıdır. İlk değeri MQRC_NONE.

MQSTS için PutSuccessSayısı (MQLONG)

Başarılı olan zamanuyumsuz koyma işlemlerinin sayısı.

PutSuccessCount değeri, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

MQCC_OK ile tamamlanan MQSTS yapısında adı belirtilen nesneye yönelik zamanuyumsuz koyma işlemlerinin sayısı.

MQSTAT_TYPE_RECONNECTION

Sıfır.

MQSTAT_TYPE_RECONNECTION_HATA

Sıfır.

PutSuccessCount bir çıkış alanıdır. İlk değeri sıfırdır.

MQSTS için PutWarningCount (MQLONG)

Bir uyarıyla sona eren zamanuyumsuz koyma işlemlerinin sayısı.

PutWarningCount değeri, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

MQCC_WARNING ile tamamlanan MQSTS yapısında adı belirtilen nesneye yönelik zamanuyumsuz koyma işlemlerinin sayısı.

MQSTAT_TYPE_RECONNECTION

Sıfır.

MQSTAT_TYPE_RECONNECTION_HATA

Sıfır.

PutWarningCount bir çıkış alanıdır. İlk değeri sıfırdır.

PutFailureMQSTS için sayı (MQLONG)

Başarısız olan zamanuyumsuz koyma işlemlerinin sayısı.

PutFailureCount değeri, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

MQCC_FAILED ile tamamlanan MQSTS yapısında adı belirtilen nesneye yönelik zamanuyumsuz koyma işlemlerinin sayısı.

MQSTAT_TYPE_RECONNECTION

Sıfır.

MQSTAT_TYPE_RECONNECTION_HATA

Sıfır.

PutFailureCount bir çıkış alanıdır. İlk değeri sıfırdır.

MQSTS için ObjectType (MQLONG)

Raporlanmakta olan *ObjectName* içinde adı belirtilen nesnenin tipi.

Olası ObjectType değerleri “MQOT_* (Nesne Tipleri ve Genişletilmiş Nesne Tipleri)” sayfa 163’ünde listelenmiştir.

ObjectType bir çıkış alanıdır. İlk değeri MQOT_Q.

MQSTS için ObjectName (MQCHAR48)

Raporlanmakta olan nesnenin adı.

ObjectName yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

Bu, MQSTS yapısındaki *CompCode* ve *Reason* alanlarında başarısızlığı bildirilen, koyma işleminde kullanılan kuyruğun ya da konunun adıdır.

MQSTAT_TYPE_RECONNECTION

Bağlantı yeniden bağlanıyorsa, bu, bağlantıyla ilişkilendirilmiş kuyruk yöneticisinin adıdır.

MQSTAT_TYPE_RECONNECTION_HATA

Bağlantı yeniden bağlanamazsa, bu, yeniden bağlantının başarısız olmasına neden olan nesnenin adıdır. Hatanın nedeni, MQSTS yapısındaki *CompCode* ve *Reason* alanlarında raporlanır.

ObjectName bir çıkış alanıdır. İlk değeri C ' deki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQSTS için ObjectQMgrAdı (MQCHAR48)

Bildirilmekte olan kuyruk yöneticisinin adı.

ObjectQMgrName yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

Bu, *ObjectName* nesnesinin tanımlandığı kuyruk yöneticisinin adıdır. İlk boş değere kadar ya da alanın sonuna kadar tümüyle boş bırakılan bir ad, uygulamanın bağlı olduğu kuyruk yöneticisini (yerel kuyruk yöneticisi) gösterir.

MQSTAT_TYPE_RECONNECTION

Multi

ObjectQMgrName alanı, yeniden bağlantı istenmekte olan bir kuyruk yöneticisinin adını içerir ya da kuyruk yöneticisi belirtilmediyse boş bırakılır. Olanaklıysa, istemci bu adı taşıyan bir kuyruk yöneticisine yeniden bağlanmayı dener.

z/OS

Boş.

MQSTAT_TYPE_RECONNECTION_HATA

Bağlantı yeniden bağlanamazsa, bu, yeniden bağlantının başarısız olmasına neden olan nesnenin adıdır. Hatanın nedeni, MQSTS yapısındaki *CompCode* ve *Reason* alanlarında raporlanır.

ObjectQMgrName bir çıkış alanıdır. Değeri, C dilinde boş dizgi ve diğer programlama dillerinde 48 boş karakterdir.

MQSTS için ResolvedObjectName (MQCHAR48)

Yerel kuyruk yöneticisi adı çözdükten sonra *ObjectName* içinde adı belirtilen nesnenin adı.

ResolvedObjectName yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

ResolvedObjectName , yerel kuyruk yöneticisi adı çözdükten sonra *ObjectName* içinde adı belirtilen nesnenin adıdır. Döndürülen ad, *ResolvedQMgrName* ile tanımlanan kuyruk yöneticisinde var olan bir nesnenin adıdır.

MQSTAT_TYPE_RECONNECTION

Boş.

MQSTAT_TYPE_RECONNECTION_HATA

Boş.

ResolvedObjectName bir çıkış alanıdır. İlk değeri C ' deki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

ResolvedQMgrMQSTS için Ad (MQCHAR48)

Yerel kuyruk yöneticisi adı çözdükten sonra hedef kuyruk yöneticisinin adı.

ResolvedQMgrName yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

ResolvedQMgrName , yerel kuyruk yöneticisi adı çözdükten sonra hedef kuyruk yöneticisinin adıdır. Döndürülen ad, *ResolvedObjectName* ile tanımlanan nesnenin iyesi olan kuyruk yöneticisinin adıdır. *ResolvedQMgrName* yerel kuyruk yöneticisinin adı olabilir.

MQSTAT_TYPE_RECONNECTION

Boş.

MQSTAT_TYPE_RECONNECTION_HATA

Boş.

ResolvedQMgrName her zaman bir çıkış alanıdır. İlk değeri C ' deki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQSTS için ObjectString (MQCHARV)

Raporlanan başarısız olan nesnenin uzun nesne adı. Yalnızca MQSTS ya da üstü sürüm 2 'de bulunur.

ObjectString yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

Bu, MQPUT işleminde kullanılan kuyruğun ya da konunun başarısız olan uzun nesne adıdır.

MQSTAT_TYPE_RECONNECTION

Sıfır uzunluklu dizgi

MQSTAT_TYPE_RECONNECTION_HATA

Bu, yeniden bağlantının başarısız olmasına neden olan nesnenin uzun nesne adıdır.

ObjectString bir çıkış alanıdır. İlk değeri sıfır uzunluklu bir dizedir.

MQSTS için SubName (MQCHARV)

Başarısız olan aboneliğin adı. Yalnızca MQSTS ya da üstü sürüm 2 'de bulunur.

SubName yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

Sıfır uzunluklu dizgi.

MQSTAT_TYPE_RECONNECTION

Sıfır uzunluklu dizgi.

MQSTAT_TYPE_RECONNECTION_HATA

Yeniden bağlantının başarısız olmasına neden olan aboneliğin adı. Abonelik adı yoksa ya da hata bir abonelik ile ilgili değilse, bu, sıfır uzunluklu bir dizedir.

SubName bir çıkış alanıdır. İlk değeri sıfır uzunluklu bir dizedir.

MQSTS için OpenOptions (MQLONG)

Raporlanan nesneyi açmak için kullanılan `OpenOptions` . Yalnızca MQSTS ya da üstü sürüm 2 'de bulunur.

`OpenOptions` değeri, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

Sıfır.

MQSTAT_TYPE_RECONNECTION

Sıfır.

MQSTAT_TYPE_RECONNECTION_HATA

Hata oluştuğunda kullanılan `OpenOptions` . Hatanın nedeni, MQSTS yapısındaki *CompCode* ve *Reason* alanlarında raporlanır.

`OpenOptions` bir çıkış alanıdır. İlk değeri sıfırdır.

MQSTS için SubOptions (MQLONG)

Hatalı aboneliği açmak için kullanılan `SubOptions` . Yalnızca MQSTS ya da üstü sürüm 2 'de bulunur.

`SubOptions` yorumu, MQSTAT **Type** parametresinin değerine bağlıdır.

MQSTAT_TYPE_ASYNC_HATASI

Sıfır.

MQSTAT_TYPE_RECONNECTION

Sıfır.

MQSTAT_TYPE_RECONNECTION_HATA

Hata oluştuğunda kullanılan `SubOptions` . Başarısızlık bir konuya abone olma ile ilgili değilse, döndürülen değer sıfırdır.

`SubOptions` bir çıkış alanıdır. İlk değeri sıfırdır.

MQTM-Tetikleyici iletisi

MQTM yapısı, kuyruk yöneticisi tarafından bir kuyruk için bir tetikleyici olayı oluştuğunda tetikleyici uygulamasına gönderilen tetikleyici iletideki verileri tanımlar. Bu yapı, IBM MQ çerçeve arabirimlerinden biri olan IBM MQ Trigger Monitor Interface (TMI) olanağının bir parçasıdır.

Biçim Adı

MQFMT_TRIGGER.

Karakter kümesi ve kodlama

MQTM 'deki karakter verileri, MQTM' yi oluşturan kuyruk yöneticisinin karakter kümesinde yer aldı. MQTM 'deki sayısal veriler, MQTM' yi oluşturan kuyruk yöneticisinin makine kodlamasında yer aldı.

MQTM ' nin karakter kümesi ve kodlaması *CodedCharSetId* ve *Encoding* alanları tarafından aşağıdaki alanlarda verilir:

- MQMD (MQTM yapısı ileti verilerinin başındadır) ya da
- MQTM yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Kullanım

Tetikleyici izleme uygulamasının, tetikleyici iletisindeki bilgilerin bir kısmını ya da tümünü tetikleyici izleme uygulamasının başlattığı uygulamaya aktarması gerekebilir. Başlatılan uygulamanın gereksinim duyabileceği bilgiler arasında *QName*, *TriggerDatave UserDatayer* alır. Tetikleyici izleyici uygulaması MQTM yapısını doğrudan başlatılan uygulamaya geçirebilir ya da ortamın izin verdiğine ve başlatılan

uygulama için uygun olduğuna bağlı olarak bir MQTMC2 yapısı geçirebilir. MQTMC2 hakkında bilgi için bkz. “MQTMC2 -Tetikleyici ileti 2 (karakter biçimi)” sayfa 600.

- **z/OS** z/OS üzerinde, CKTI hareketi kullanılarak başlatılan bir MQAT_CICS uygulaması için, tetikleyici ileti yapısının tamamı MQTM başlatılmış hareket tarafından kullanılabilir; bilgiler EXEC CICS RETRIEVE komutu kullanılarak alınabilir.
- **IBM i** IBM sistemlerinde, IBM MQ ile sağlanan tetikleyici izleyicisi uygulaması, başlatılan uygulamaya bir MQTMC2 yapısı geçirir.

Tetikleyicilerin kullanılmasıyla ilgili bilgi için [Tetikleyici kullanarak IBM MQ uygulamalarının başlatılması](#) başlıklı konuya bakın.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 533. MQTM for MQTM içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
StrucId (yapı tanıtıcısı)	MQTM_STRUC_ID	'TM-'
Sürüm (yapı sürümü numarası)	MQTM_VERSION_1	1
QName (tetiklenen kuyruğun adı)	Yok	Boş değerli dizilim ya da boşluklar
ProcessName (süreç nesnesinin adı)	Yok	Boş değerli dizilim ya da boşluklar
TriggerData (tetikleyici verileri)	Yok	Boş değerli dizilim ya da boşluklar
ApplType (uygulama tipi)	Yok	0
AppId (uygulama tanıtıcısı)	Yok	Boş değerli dizilim ya da boşluklar
EnvData (ortam verileri)	Yok	Boş değerli dizilim ya da boşluklar
UserData (kullanıcı verileri)	Yok	Boş değerli dizilim ya da boşluklar

Notlar:

- simgesi, tek bir boş karakteri temsil eder.
- Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.
- C programlama dilinde, makro değişkeni MQTM_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQTM MyTM = {MQTM_DEFAULT};
```

Dil bildirimleri

MQTM için C bildirimi

```
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 */
};

```

MQTM için COBOL bildirimi

```

**  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).

```

MQTM için PL/I bildirimi

```

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 */

```

MQTM için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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)

```

MQTM için Visual Basic bildirimi

```

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

```

Tetikleyici iletisine ilişkin MQMD

Çizelge 534. Kuyruk yöneticisi tarafından oluşturulan bir tetikleyici iletisine ilişkin MQMD 'deki alanlara ilişkin ayarlar

MQMD 'deki Alan	Kullanılan değer
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_1
<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>	Kuyruk yöneticisinin CodedCharSetId özneliği
<i>Format</i>	MQFMT_TRIGGER
<i>Priority</i>	Başlatma kuyruğunun DefPriority özneliği
<i>Persistence</i>	MQPER_NOT_KALICI
<i>MsgId</i>	Benzersiz bir değer
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	Boşluklar
<i>ReplyToQMgr</i>	Kuyruk yöneticisinin adı
<i>UserIdentifier</i>	Boşluklar
<i>AccountingToken</i>	MQACT_NONE
<i>ApplIdentityData</i>	Boşluklar
<i>PutApplType</i>	MQAT_QMGR ya da ileti kanalı aracısı için uygun olduğu şekilde
<i>PutApplName</i>	Kuyruk yöneticisi adının ilk 28 baytı
<i>PutDate</i>	Tetikleyici iletisinin gönderildiği tarih
<i>PutTime</i>	Tetikleyici iletisinin gönderildiği zaman
<i>ApplOriginData</i>	Boşluklar

Aşağıdakiler dışında benzer değerleri ayarlamak için bir tetikleyici iletisi oluşturan bir uygulama önerilir:

- *Priority* alanı MQPRI_PRIORITY_AS_Q_DEF olarak ayarlanabilir (kuyruk yöneticisi, ileti bulunduğu başlatma kuyruğu için varsayılan önceliğe çevirir).
- *ReplyToQMgr* alanı boş olarak ayarlanabilir (kuyruk yöneticisi, yerleştirdiği iletide bu değeri yerel kuyruk yöneticisinin adına çevirir).
- Bağlam alanlarını uygulama için uygun şekilde ayarlayın.

MQTM için StrucId (MQCHAR4)

Bu, tetikleyici ileti yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQTM_STRUC_ID.

Değer şöyle olmalıdır:

MQTM_STRUC_ID

Tetikleyici ileti yapısına ilişkin tanıttıcı.

C programlama dili için, MQTM_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQTM_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQTM için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MQTM_VERSION_1

Tetikleyici ileti yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQTM_CURRENT_VERSION

Tetikleyici ileti yapısının yürürlükteki sürümü.

Bu alanın ilk değeri MQTM_VERSION_1' dir.

MQTM için QName (MQCHAR48)

Bu, tetikleyici olayının oluştuğu kuyruğun adıdır ve tetikleyici izleme uygulaması tarafından başlatılan uygulama tarafından kullanılır. Kuyruk yöneticisi, tetiklenen kuyruğun **QName** özniteliğinin değeriyle bu alanı kullanıma hazırlar; bu özniteliğin ayrıntıları için bkz. "[Kuyruklara ilişkin öznitelikler](#)" sayfa 826.

Alanın tanımlı uzunluğundan daha kısa olan adlar sağa boşluklarla doldurulur; bunlar boş bir karakterle zamanından önce sona erdirilmez.

Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQTM için ProcessName (MQCHAR48)

Bu, tetiklenen kuyruk için belirtilen kuyruk yöneticisi işlem nesnesinin adıdır ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılabilir. Kuyruk yöneticisi bu alanı, **QName** alanıyla tanımlanan kuyruğun **ProcessName** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. "[Kuyruklara ilişkin öznitelikler](#)" sayfa 826.

Alanın tanımlı uzunluğundan daha kısa olan adlar her zaman sağa boşluklarla doldurulur; boş bir karakterle zamanından önce sona erdirilmezler.

Bu alanın uzunluğu MQ_PROCESS_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQTM için TriggerData (MQCHAR64)

Bu, tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılmak üzere serbest biçimli verilerdir. Kuyruk yöneticisi bu alanı, **QName** alanıyla tanımlanan kuyruğun **TriggerData** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. "[Kuyruklara ilişkin öznitelikler](#)" sayfa 826 . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

z/OS üzerinde, CKTI işlemini kullanmaya başlayan bir CICS uygulaması için bu bilgiler kullanılmaz.

Bu alanın uzunluğu MQ_TRIGGER_DATA_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerinde 64 boş karakterdir.

ApplType (MQTM için MQLONG)

Bu, başlatılacak programın türünü tanımlar ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılır. Kuyruk yöneticisi bu alanı, **ProcessName** alanıyla tanımlanan süreç nesnesinin

AppType özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. “Süreç tanımlamalarına ilişkin öznitelikler” sayfa 860 . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

AppType aşağıdaki standart değerlerden birine sahip olabilir. Kullanıcı tanımlı tipler de kullanılabilir, ancak MQAT_USER_LAST ile MQAT_USER_FIRST aralığındaki değerlerle kısıtlanmalıdır:

MQAT_AIX

AIX uygulaması (MQAT_UNIX ile aynı değer).

MQAT_BATCH

Toplu iş uygulaması

MQAT_BROKER

Aracı uygulaması

MQAT_CICS

CICS hareketi.

MQAT_CICS_BRIDGE

CICS bridge uygulaması.

MQAT_CICS_VSE

CICS/VSE hareketi.

MQAT_DOS

PC DOS üzerinde IBM MQ MQI client uygulaması.

MQAT_IMS

IMS uygulaması.

MQAT_IMS_BRIDGE

IMS köprü uygulaması.

MQAT_JAVA

Java uygulaması.

MQAT_MVS

MVS ya da TSO uygulaması (MQAT_ZOS ile aynı değer).

MQAT_NOTES_AGENT

Lotus Notes Agent uygulaması.

MQAT_OS390

OS/390 uygulaması (MQAT_ZOS ile aynı değer).

MQAT_OS400

IBM i uygulaması.

MQAT_RRS_BATCH

RRS toplu iş uygulaması.

MQAT_UNIX

UNIX uygulaması.

MQAT_BILINMIYOR

Bilinmeyen tipte uygulama.

MQAT_KULLANICISI

Kullanıcı tanımlı uygulama tipi.

MQAT_VOS

Stratus VOS uygulaması.

MQAT_WINDOWS

16 bit Windows uygulaması.

MQAT_WINDOWS_NT

32 bit Windows uygulaması.

MQAT_WLM

z/OS iş yükü yöneticisi uygulaması.

MQAT_XCF

-XCF.

MQAT_ZOS

z/OS uygulaması.

MQAT_USER_FIRST

Kullanıcı tanımlı uygulama tipi için en düşük değer.

MQAT_USER_LAST

Kullanıcı tanımlı uygulama tipi için en yüksek değer.

Bu alanın ilk değeri şudur: 0.

MQTM için ApplId (MQCHAR256)

Bu, başlatılacak uygulamayı tanıtan ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılan bir karakter dizilimidir. Kuyruk yöneticisi bu alanı, *ProcessName* alanıyla tanıtilan süreç nesnesinin **AppId** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. “Süreç tanımlamalarına ilişkin öznitelikler” sayfa 860 . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

AppId ' in anlamı, tetikleyici-izleyici uygulaması tarafından belirlenir. IBM MQ tarafından sağlanan tetikleyici, *AppId* ' in yürütülebilir bir programın adı olmasını gerektirir. Aşağıdaki notlar, belirtilen ortamlar için geçerlidir:

- z/OS' da *AppId* :
 - CICS tetikleyici izleme işlemi CKTI kullanılarak başlatılan uygulamalar için bir CICS işlem tanıtıcısı
 - IMS tetikleyici izleme programı CSQQTRMN kullanılarak başlatılan uygulamalara ilişkin IMS hareket tanıtıcısı
- Windows sistemlerinde, program adının başına bir sürücü ve izin yolu öneki konabilir.
- IBM işletim sistemi üzerinde, program adının başına bir kitaplık adı ve/karakteri konabilir.
- AIX and Linux üzerinde, program adının başına bir izin yolu öneki konabilir.

Bu alanın uzunluğu MQ_PROCESS_APPL_ID_LENGTH tarafından verilir. Bu alanın ilk değeri C ' deki boş dizgi ve diğer programlama dillerinde 256 boş karakterdir.

MQTM için EnvData (MQCHAR128)

Bu, başlatılacak uygulamayla ilgili ortamla ilgili bilgileri içeren ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılan bir karakter dizilimidir. Kuyruk yöneticisi bu alanı, *ProcessName* alanıyla tanıtilan süreç nesnesinin **EnvData** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. “Süreç tanımlamalarına ilişkin öznitelikler” sayfa 860 . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

z/OS sistemlerinde, CKTI hareketi kullanılarak başlatılan bir CICS uygulaması ya da CSQQTRMN hareketi kullanılarak başlatılacak bir IMS uygulaması için bu bilgi kullanılmaz.

Bu alanın uzunluğu MQ_PROCESS_ENV_DATA_LENGTH tarafından verilir. Bu alanın ilk değeri, C ' deki boş dizgi ve diğer programlama dillerindeki 128 boş karakterdir.

MQTM için UserData (MQCHAR128)

Bu, başlatılacak uygulamayla ilgili kullanıcı bilgilerini içeren ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılan bir karakter dizilimidir. Kuyruk yöneticisi bu alanı, *ProcessName* alanıyla tanıtilan süreç nesnesinin **UserData** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. “Süreç tanımlamalarına ilişkin öznitelikler” sayfa 860 . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

Microsoft Windows için, süreç tanımlaması **runmqtrm** ' e geçirilecekse, karakter dizgisi çift tırnak işareti içermemelidir.

Bu alanın uzunluğu MQ_PROCESS_USER_DATA_LENGTH tarafından verilir. Bu alanın ilk değeri, C ' deki boş dizgi ve diğer programlama dillerindeki 128 boş karakterdir.

MQTM2 -Tetikleyici ileti 2 (karakter biçimi)

Tetikleyici uygulaması başlatma kuyruğundan bir tetikleyici iletisi (MQTM) aldığımda, tetikleyici izleme programının tetikleyici iletisindeki bilgilerin bir kısmını ya da tümünü tetikleyici izleyicinin başlatdığı uygulamaya geçirmesi gerekebilir.

Başlatılan uygulamanın gereksinim duyabileceği bilgiler arasında *QName*, *TriggerData*ve *UserData*yer alır. Tetikleyici izleyici uygulaması MQTM yapısını doğrudan başlatılan uygulamaya geçirebilir ya da ortamın izin verdiğine ve başlatılan uygulama için uygun olduğuna bağlı olarak bir MQTM2 yapısı geçirebilir.

Bu yapı, IBM MQ çerçeve arabirimlerinden biri olan IBM MQ Trigger Monitor Interface (TMI) olanağının bir parçasıdır.

Karakter kümesi ve kodlama

MQTM2 içindeki karakter verileri yerel kuyruk yöneticisinin karakter kümesinde; bu, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından verilir.

Kullanım

MQTM2 yapısı, MQTM yapısının biçimine çok benzer. Fark, MQTM 'deki karakter olmayan alanların MQTM2 ' de aynı uzunlukta karakter alanlarına değiştirilmesi ve kuyruk yöneticisi adının yapının sonuna eklenmeleridir.

- **z/OS** z/OSüzerinde, CSQQTRMN uygulaması kullanılarak başlatılan bir MQAT_IMS uygulaması için, başlatılan uygulama için bir MQTM2 yapısı kullanılır kılmıştır.
- **IBM i** IBM sistemlerinde, IBM MQ ile sağlanan tetikleyici izleyicisi uygulaması, başlatılan uygulamaya bir MQTM2 yapısı geçirir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 535. MQTM2 içindeki alanlar		
Alan adı ve açıklaması	Değişmez adı	Değişmez ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQTM2_STRUC_ID	' TMC2 '
<u>Sürüm</u> (yapı sürümü numarası)	MQTM2_VERSION_2	' 2222 '
<u>QName</u> (tetiklenen kuyruğun adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>ProcessName</u> (süreç nesnesinin adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>TriggerData</u> (tetikleyici verileri)	Yok	Boş değerli dizilim ya da boşluklar
<u>ApplType</u> (uygulama tipi)	Yok	Boşluklar
<u>AppId</u> (uygulama tanıtıcısı)	Yok	Boş değerli dizilim ya da boşluklar
<u>EnvData</u> (ortam verileri)	Yok	Boş değerli dizilim ya da boşluklar
<u>UserData</u> (kullanıcı verileri)	Yok	Boş değerli dizilim ya da boşluklar

Çizelge 535. MQTMC2 içindeki alanlar (devamı var)

Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
QMgrName (kuyruk yöneticisi adı)	Yok	Boş değerli dizilim ya da boşluklar

Notlar:

1. ↪ simgesi, tek bir boş karakteri temsil eder.
2. Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.
3. C programlama dilinde, makro değişkeniMQTMC2_DEFAULT , yukarıda listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQTMC2 MyTMC = {MQTMC2_DEFAULT};
```

Dil bildirimleri

MQTMC2 için C bildirimi

```
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 */
};
```

MQTMC2 için COBOL bildirimi

```
** 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).
```

MQTMC2 için PL/I bildirimi

```
dcl
 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 */

```

MQTMC2 için High Level Assembler bildirimi

```

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)

```

MQTMC2 için Visual Basic bildirimi

```

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) for MQTMC2

Bu, tetikleyici iletisi 2 (karakter biçimi) yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri: MQTMC2_STRUC_ID.

Değer şöyle olmalıdır:

MQTMC2_STRUC_ID

Tetikleyici ileti (karakter biçimi) yapısının tanıtıcısı.

C programlama dili için MQTMC2_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQTMC2_STRUC_ID ile aynı değere sahip, ancak dizgi yerine bir karakter dizisi.

MQTMC2 için Sürüm (MQCHAR4)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

MQTMC_VERSION_2

Sürüm 2 tetikleyici iletisi (karakter biçimi) yapısı.

C programlama dili için, MQTMC_VERSION_2_ARRAY değişmezi de tanımlanır; bu değer MQTMC_VERSION_2 ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQTMC_CURRENT_VERSION

Tetikleyici ileti (karakter biçimi) yapısının yürürlükteki sürümü.

MQTM2 için QName (MQCHAR48)

Tetiklenen kuyruğun adı.

MQTM yapısındaki *QName* alanına bakın.

ProcessName (MQCHAR48)- MQTM2

Süreç nesnesinin adı.

MQTM yapısındaki *ProcessName* alanına bakın.

MQTM2 için TriggerData (MQCHAR64)

Tetikleyici veri.

MQTM yapısındaki *TriggerData* alanına bakın.

AppType (MQCHAR4)- MQTM2

Uygulama tipi.

Bu alan, özgün tetikleyici iletisinin MQTM yapısındaki *AppType* alanındaki değer ne olursa olsun, her zaman boşluk içerir.

AppId (MQCHAR256)- MQTM2

Uygulama tanıtıcısı.

MQTM yapısındaki *AppId* alanına bakın.

EnvData (MQCHAR128)- MQTM2

Ortam verileri.

MQTM yapısındaki *EnvData* alanına bakın.

UserData (MQCHAR128)- MQTM2

Kullanıcı verileri.

MQTM yapısındaki *UserData* alanına bakın.

QMgrName (MQCHAR48)- MQTM2

Kuyruk yöneticisi adı.

Bu, tetikleme olayının oluşturduğu kuyruk yöneticisinin adıdır.

MQWIH-İş bilgileri üstbilgisi

Bir ileti z/OS iş yükü yöneticisi (WLM) tarafından işlenecek olursa, iletinin bir MQWIH yapısıyla başlaması gerekir. Bu yapı, WLM tarafından işlenecek bir iletinin başında bulunması gereken bilgileri açıklar.

Kullanılabilirlik

Tüm IBM MQ sistemleri ve bu sistemlere bağlı IBM MQ istemcileri.

Biçim Adı

MQFMT_WORK_INFO_HEADER.

Karakter kümesi ve kodlama

MQWIH yapısındaki alanlar, MQWIH ' den önce gelen üstbilgi yapısındaki *CodedCharSetId* ve *Encoding* alanları tarafından belirtilen karakter kümesinde ve kodlamasında ya da MQWIH uygulama iletilisi verilerinin başlangıcında ise MQMD yapısındaki bu alanlar tarafından bulunur.

Karakter takımı, kuyruk adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır.

Kullanım

Desteklenen herhangi bir IBM MQ altyapısı için MQWIH yapısını içeren bir ileti yaratabilir ve iletebilir, ancak yalnızca bir IBM MQ for z/OS kuyruk yöneticisi WLM ile etkileşimde bulunabilir. Bu nedenle, iletinin z/OS dışı bir kuyruk yöneticisinden WLM ' ye ulaşabilmesi için kuyruk yöneticisi ağızının, iletinin yönetilebileceği en az bir z/OS kuyruk yöneticisi içermesi gerekir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 536. MQWIH ' deki Alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQWIH_STRUC_ID	'WIH↵'
<u>Sürüm</u> (yapı sürümü numarası)	MQWIH_VERSION_1	1
<u>StrucLength</u> (MQWIH yapısının uzunluğu)	MQWIH_LENGTH_1	120
<u>Kodlama</u> (MQWIH ' yi izleyen verilerin sayısal kodlaması)	Yok	0
<u>CodedCharSetId</u> (MQWIH ' yi izleyen verilerin karakter kümesi tanıtıcısı)	MQCCSI_AYRILMAMIŞ	0
<u>Biçim</u> (MQWIH ' yi izleyen verilerin biçim adı)	MQFMT_NONE	Boşluklar
<u>İşaretler</u> (işaretler)	MQWIH_NONE	0
<u>ServiceName</u> (hizmet adı)	Yok	Boşluklar
<u>ServiceStep</u> (hizmet adımı adı)	Yok	Boşluklar
<u>MsgToken</u> (ileti simgesi)	MQMTOK_NONE	Boş Değerler
<u>Ayrılmış</u> (ayrılmış)	Yok	Boşluklar
Notlar: 1. ↵ simgesi, tek bir boş karakteri temsil eder. 2. C programlama dilinde, makro değişkeni MQWIH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın: <pre>MQWIH MyWIH = {MQWIH_DEFAULT};</pre>		

Dil bildirimleri

MQWIH için C bildirimi

```
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 */
};

```

MQWIH için COBOL bildirimi

```

** 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).

```

MQWIH için PL/I bildirimi

```

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 */

```

MQWIH için High Level Assembler bildirimi

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)	

MQWIH için Visual Basic bildirimi

```

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'
  follows MQWIH'
  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

```

MQWIH için StrucId (MQCHAR4)

Bu, iş bilgileri üstbilgi yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQWIH_STRUC_ID.

Değer şöyle olmalıdır:

MQWIH_STRUC_ID

İş bilgileri üstbilgi yapısına ilişkin tanıttıcı.

C programlama dili için, MQWIH_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQWIH_STRUC_ID ile aynı değere sahip, ancak bir dizgi yerine bir karakter dizisi.

MQWIH için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MQWIH_VERSION_1

Version-1 iş bilgileri üstbilgi yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQWIH_CURRENT_VERSION

İş bilgileri üstbilgi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri MQWIH_VERSION_1' dir.

MQWIH için StrucLength (MQLONG)

Bu, MQWIH yapısının uzunluğudur. Değer şöyle olmalıdır:

MQWIH_LENGTH_1

version-1 iş bilgileri üstbilgi yapısının uzunluğu.

Aşağıdaki değişmez, yürürlükteki sürümün uzunluğunu belirtir:

MQWIH_CURRENT_LENGTH

İş bilgileri üstbilgi yapısının yürürlükteki sürümünün uzunluğu.

Bu alanın ilk değeri: MQWIH_LENGTH_1.

MQWIH için kodlama (MQLONG)

MQWIH yapısını izleyen verilerin sayısal kodlamasını belirtir; MQWIH yapısının kendisindeki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk deęeri 0 'dir.

MQWIIH için MQLONG

MQWIIH yapısını izleyen verilerin karakter kümesi tanıtıcısını belirtir; MQWIIH yapısının kendisindeki karakter verileri için geçerli deęildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun deęere ayarlaması gerekir. Aşağıdaki özel deęeri kullanabilirsiniz:

MQCCSI_INHERIT

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu deęeri, yapının gerçek karakter kümesi tanıtıcısına deęiştirir. Hata oluşmaması durumunda, MQGET çağrısı MQCCSI_INHERIT deęerini döndürmez.

MQMD ' deki *PutAppType* alanının deęeri MQAT_BROKER ise, MQCSI_INHERIT kullanılamaz.

Bu alanın ilk deęeri MQCCSI_UNDEFINED 'dir.

MQWIIH için biçim (MQCHAR8)

MQWIIH yapısını izleyen verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun deęere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *Format* alanıyla aynıdır.

Bu alanın uzunluğu MQ_FORMAT_LENGTH tarafından verilir. Bu alanın ilk deęeri MQFMT_NONE.

MQWIIH için İşaretler (MQLONG)

Deęer şöyle olmalıdır:

MQWIIH_NONE

Bayrak yok.

Bu alanın ilk deęeri MQWIIH_NONE.

MQWIIH için ServiceName (MQCHAR32)

Bu, iletiyi işleyecek hizmetin adıdır.

Bu alanın uzunluğu MQ_SERVICE_NAME_LENGTH tarafından verilir. Bu alanın ilk deęeri 32 boş karakterdir.

MQWIIH için ServiceStep (MQCHAR8)

Bu, iletinin ilgili olduęu *ServiceName* adımının adıdır.

Bu alanın uzunluğu MQ_SERVICE_STEP_LENGTH tarafından verilir. Bu alanın ilk deęeri 8 boş karakterdir.

MQWIIH için MsgToken (MQBYTE16)

Bu, iletiyi benzersiz olarak tanıtan bir ileti simgesidir.

MQPUT ve MQPUT1 çağrıları için bu alan yoksayılr. Bu alanın uzunluğu MQ_MSG_TOKEN_LENGTH tarafından verilir. Bu alanın ilk deęeri MQMTOK_NONE.

MQWIIH için ayrılmış (MQCHAR32)

Bu ayrılmış bir alandır; boş olmalıdır.

MQXP-Çıkış deęiştirgesi öbeęi

MQXP yapısı, API-geçiş çıkışına giriş/çıkış parametresi olarak kullanılır. Bu çıkışla ilgili daha fazla bilgi için bkz. [API-geçiş çıkışı](#).

Karakter kümesi ve kodlama

MQXP 'deki karakter verileri yerel kuyruk yöneticisinin karakter kümesinde; bu, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından verilir. MQXP 'deki sayısal veriler yerli makine kodlamasında; bu, MQENC_NATIVE tarafından verilir.

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 537. MQXP 'deki Alanlar	
Alan adı ve açıklaması	Değişmezin adı
StrucId (yapı tanıtıcısı)	MQXP_STRUC_ID
Sürüm (yapı sürümü numarası)	MQXP_VERSION_1
ExitId (çıkış tanıtıcısı)	MQXT_API_CROSSING_EXIT
ExitReason (çıkışın çağrılmasının nedeni)	Yok
ExitResponse (çıkıştan yanıt)	Yok
ExitCommand (API çağrı kodu)	Yok
ExitParmSayı (parametre sayısı)	Yok
Ayrılmış (ayrılmış)	Yok
ExitUserAlanı (kullanıcı alanı)	Yok

Dil bildirimleri

MQXP için C bildirimini

```
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 */
};
```

MQXP için COBOL bildirimini

```
** 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) .

```

MQXP için PL/I bildirimi

```

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 */

```

MQXP için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

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)

```

MQXP için StrucId (MQCHAR4)

Bu, çıkış parametresi yapısının yapı tanıtıcısıdır. Her zaman bir giriş alanıdır. Değeri MQXP_STRUC_ID 'dir.

Değer şöyle olmalıdır:

MQXP_STRUC_ID

Çıkış parametresi yapısına ilişkin tanıtıcı.

C programlama dili için, değişmez MQXP_STRUC_ID_ARRAY da tanımlanır. Bu değer MQXP_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

MQXP için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MQXP_VERSION_1

Çıkış değiştirgesi öbeği yapısına ilişkin sürüm numarası.

Not: Bu yapının yeni bir sürümü tanıtıldığında, var olan parçanın düzeni değişmez. Bu nedenle çıkış, sürüm numarasının, çıkışın kullanması gereken alanları içeren en düşük sürüme eşit ya da bu sürümden büyük olup olmadığını denetlemesi gerekir.

Bu, çıkışa bir giriş alanıdır.

MQXP için ExitId (MQLONG)

Bu, çıkış yordamının girişinde ayarlanır ve çıkış tipini gösterir:

MQXT_API_CROSSING_EXIT

CICS için API geçiş çıkışı.

Bu, çıkışa bir giriş alanıdır.

MQXP için ExitReason (MQLONG)

Bu, çıkış yordamına ilişkin girişte ayarlanır. API geçişi çıkışı için, yordamın API çağrısından önce mi, yoksa sonra mı çağrıldığını gösterir:

MQXR_BEFORE

API yürütmeden önce.

MQXR_AFTER

API yürütüldükten sonra.

Bu, çıkışa bir giriş alanıdır.

MQXP için ExitResponse (MQLONG)

Değer, çağırana ile iletişim kurmak için çıkış tarafından ayarlanır. Aşağıdaki değerler tanımlanır:

MQXCC_OK

Çıkış başarıyla tamamlandı.

MQXCC_SUPPRESS_FUNCTION

İşlevi engelle.

Bu değer, API çağrısından *önce* adlı bir API geçiş çıkışı tarafından ayarlandığında API çağrısı gerçekleştirilmez. Aramaya ilişkin *CompCode* , *MQCC_FAILED* olarak ayarlanır, *Reason* , *MQRC_SUPPRESSED_BY_EXIT* olarak ayarlanır ve diğer tüm parametreler çıkış onları bıraktığı gibi kalır.

Bu değer, API çağrısından *sonra* adlı bir API geçiş çıkışı tarafından ayarlandığında, kuyruk yöneticisi tarafından yoksayılır.

MQXCC_SKIP_FUNCTION

İşlevi atla.

Bu değer, API çağrısından *önce* adlı bir API geçiş çıkışı tarafından ayarlandığında, API çağrısı gerçekleştirilmez; *CompCode* ve *Reason* ve diğer tüm parametreler çıkış onlardan ayrıldığında kalır.

Bu değer, API çağrısından *sonra* adlı bir API geçiş çıkışı tarafından ayarlandığında, kuyruk yöneticisi tarafından yoksayılır.

Bu, çıkıştan bir çıkış alanıdır.

MQXP için ExitCommand (MQLONG)

Bu alan, çıkış yordamına ilişkin girişte ayarlanır. Çıkışın çağrılmasına neden olan API çağrısını tanımlar:

MQXC_CALLBACK

CALLBACK çağrısı.

MQXC_MQBACK

MQBACK çağrısı.

MQXC_MQCB

MQCB çağrısı.

MQXC_MQCLOSE

MQCLOSE çağrısı.

MQXC_MQCMIT

MQCMIT çağrısı.

MQXC_MQCTL

MQCTL çağrısı.

MQXC_MQGET

MQGET çağrısı.

MQXC_MQINQ

MQINQ çağrısı.

MQXC_MQOPEN

MQOPEN çağrısı.

MQXC_MQPUT

MQPUT çağrısı.

MQXC_MQPUT1

MQPUT1 çağrısı.

MQXC_MQSET

MQSET çağrısı.

MQXC_MQSTAT

MQSTAT çağrısı.

MQXC_MQSUB

MQSUB çağrısı.

MQXC_MQSUBRQ

MQSUBRQ çağrısı.

Bu, çıkışa bir giriş alanıdır.

ExitParmMQXP için sayı (MQLONG)

Bu alan, çıkış yordamına ilişkin girişte ayarlanır. MQ çağrısının aldığı parametre sayısını içerir.

Çizelge 538. Her MQ çağrısı için parametre sayısı

Arama adı	Parametre sayısı
MQBACK	3
MQCLOSE	5
MQCMIT	3
MQGet	9
MQINQ	10
MQOPEN	6
MQPUT	8
MQPUT1	8
MQSET	10

Bu, çıkışa bir giriş alanıdır.

MQXP için ayrılmış (MQLONG)

Bu ayrılmış bir alandır. Değeri çıkış için önemli değil.

MQXP için ExitUserAlanı (MQBYTE16)

Bu, çıkışın kullanabileceği bir alandır. Görev için çıkışın ilk çağrılmasından önce alanın uzunluğu için ikili sıfıra başlatılır ve çıkışla bu alanda yapılan değişiklikler çıkışın çağrılmasıyla korunur. Aşağıdaki değer tanımlandı:

MQXUA_NONE

Kullanıcı bilgisi yok.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQXUA_NONE_ARRAY değişmezi de tanımlanır; bu değer MQXUA_NONE ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu alanın uzunluğu MQ_EXIT_USER_AREA_LENGTH tarafından verilir. Bu, çıkışa bir giriş/çıkış alanıdır.

MQXQH-İletim kuyruğu üstbilgisi

MQXQH yapısı, iletim kuyruklarındayken iletilerin uygulama ileti verilerine önek olarak eklenen bilgileri tanımlar.İletim kuyruğu, uzak kuyruklara (yerel kuyruk yöneticisine ait olmayan kuyruklara) yönelik iletileri geçici olarak tutan özel bir yerel kuyruk tipidir. Bir iletim kuyruğu, MQUS_ILETIM değerine sahip **Usage** kuyruk özneliğiyle gösterilir.

Biçim Adı

MQFMT_XMIT_Q_HEADER

Karakter kümesi ve kodlama

MQXQH içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından belirtilen karakter kümesinde ve MQENC_NATIVE tarafından verilen yerel kuyruk yöneticisinin kodlamasında olmalıdır.

MQXQH ' nin karakter kümesini ve kodlamasını aşağıdaki alanlardaki *CodedCharSetId* ve *Encoding* alanlarına ayarlayın:

- Ayrı MQMD (MQXQH yapısı ileti verilerinin başındadır) ya da
- MQXQH yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Alanlar

Not: Aşağıdaki tabloda, alanlar alfabetik olarak değil, kullanıma göre gruplanır. Alt konular aynı sırayı izler.

Çizelge 539. MQXQH için MQXQH içindeki alanlar		
Alan adı ve açıklaması	Değişmezin adı	Değişmezin ilk değeri (varsa)
<u>StrucId</u> (yapı tanıtıcısı)	MQXQH_STRUC_ID	'XQH-'
<u>Sürüm</u> (yapı sürümü numarası)	MQXQH_VERSION_1	1
<u>RemoteQName</u> (hedef kuyruğun adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>RemoteQMgrAd</u> (hedef kuyruk yöneticisinin adı)	Yok	Boş değerli dizilim ya da boşluklar
<u>MsgDesc</u> (özgün ileti tanımlayıcısı)	MQMD ile aynı adlar ve değerler; bkz. Çizelge 500 sayfa 421	-

Notlar:

1. - simgesi, tek bir boş karakteri temsil eder.
2. Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.
3. C programlama dilinde, makro değişkeniMQXQH_DEFAULT, çizelgede listelenen değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQXQH MyXQH = {MQXQH_DEFAULT};
```

Dil bildirimleri

MQXQH için C bildirimi

```
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 */
};

```

MQXQH için COBOL bildirimi

```

** 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).

```

MQXQH için PL/I bildirimi

```

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,         /* 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 ReplyToMgr       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 */

```

MQXQH için High Level Assembler (Üst Düzey Çevirici) bildirimi

```

MQXQH              DSECT
MQXQH_STRUCID      DS CL4  Structure identifier
MQXQH_VERSION      DS F    Structure version number
MQXQH_REMOTENAME   DS CL48 Name of destination queue
MQXQH_REMOTEMGRNAME 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

```

```
MQXQH_AREA          ORG  MQXQH
                    DS   CL(MQXQH_LENGTH)
```

MQXQH için Visual Basic bildirimi

```
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
```

Ayrı ileti tanımlayıcısındaki alanlar

İletim kuyruğundaki bir iletide *iki* ileti tanımlayıcısı vardır:

- Bir ileti tanımlayıcısı, ileti verilerinden ayrı olarak saklanır; bu, *ayrı ileti tanımlayıcısı* olarak adlandırılır ve ileti iletim kuyruğuna yerleştirildiğinde kuyruk yöneticisi tarafından oluşturulur. Ayrı ileti tanımlayıcısındaki bazı alanlar, MQPUT ya da MQPUT1 çağrısındaki uygulama tarafından sağlanan ileti tanımlayıcısından kopyalanır.

Ayrı ileti tanımlayıcısı, ileti iletim kuyruğundan kaldırıldığında MQGET çağrısının **MsgDesc** değiştirilmesinde uygulamaya döndürülen iletidir.

- İleti verilerinin bir parçası olarak MQXQH yapısında ikinci bir ileti tanımlayıcısı saklanır; bu, *yerleşik ileti tanımlayıcısı* olarak adlandırılır ve MQPUT ya da MQPUT1 çağrısında (ikincil çeşitlemelerle) uygulama tarafından sağlanan ileti tanımlayıcısının bir kopyasıdır.

Yerleşik ileti tanımlayıcısı her zaman bir version-1 MQMD 'dir. Uygulama tarafından konan iletide MQMD 'deki version-2 alanlarından biri ya da daha fazlası için varsayılan olmayan değerler varsa, MQMDE yapısı MQXQH' yi izler ve ardından uygulama iletisi verileri (varsa) gelir. MQMDE aşağıdakilerden biri olabilir:

- Kuyruk yöneticisi tarafından üretilir (uygulama iletiyi koymak için version-2 MQMD kullanıyorsa) ya da
- Uygulama ileti verilerinin başlangıcında zaten var (uygulama iletiyi koymak için version-1 MQMD kullanıyorsa).

Yerleşik ileti tanımlayıcısı, ileti son hedef kuyruktan kaldırıldığında MQGET çağrısının **MsgDesc** değiştirilmesinde uygulamaya döndürülen iletidir.

Ayrı ileti tanımlayıcısındaki alanlar, kuyruk yöneticisi tarafından gösterildiği gibi ayarlanır. Kuyruk yöneticisi version-2 MQMD ' yi desteklemiyorsa, işlev kaybı olmadan version-1 MQMD kullanılır.

Çizelge 540. Ayrı MQMD ' deki alanlar için kullanılan değerler

Ayrı MQMD ' deki alan	Kullanılan değer
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_2
<i>Report</i>	Yerleşik ileti tanımlayıcısından kopyalandı, ancak MQRO_ACCEPT_UNSUP_IF_XMIT_MASK tarafından tanımlanan bitler sıfır olarak ayarlandı. (Bu, bir ileti bir iletim kuyruğuna yerleştirildiğinde ya da iletiden kaldırıldığında bir COA ya da COD rapor iletisinin oluşturulmasını önler.)
<i>MsgType</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>Expiry</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>Feedback</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>Encoding</i>	MQENC_NATIVE (nota bakın)
<i>CodedCharSetId</i>	Kuyruk yöneticisinin CodedCharSetId özneliği.

Çizelge 540. Ayrı MQMD 'deki alanlar için kullanılan değerler (devamı var)

Ayrı MQMD 'deki alan	Kullanılan değer
<i>Format</i>	MQFMT_XMIT_Q_HEADER
<i>Priority</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>Persistence</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>MsgId</i>	Kuyruk yöneticisi tarafından yeni bir değer oluşturulur. Bu ileti tanıtıcısı, kuyruk yöneticisinin önceden açıklanan yerleşik ileti tanımlayıcısı için oluşturduğu <i>MsgId</i> ileteninden farklıdır.
<i>CorrelId</i>	Yerleşik ileti tanımlayıcısından <i>MsgId</i> . SYSTEM.CLUSTER.TRANSMIT.QUEUE, <i>CorrelId</i> iç kullanım için ayrılmıştır.
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>ReplyToQMgr</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>UserIdentifier</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>AccountingToken</i>	Yerleşik ileti tanımlayıcısından kopyalandı. SYSTEM.CLUSTER.TRANSMIT.QUEUE, <i>AccountingToken</i> iç kullanım için ayrılmıştır.
<i>AppIdentityData</i>	Yerleşik ileti tanımlayıcısından kopyalandı.
<i>PutAppType</i>	MQAT_QMGR
<i>PutAppName</i>	Kuyruk yöneticisi adının ilk 28 baytı.
<i>PutDate</i>	İletinin iletim kuyruğuna konma tarihi.
<i>PutTime</i>	İletinin iletim kuyruğuna konma süresi.
<i>AppOriginData</i>	Boşluklar
<i>GroupId</i>	MQGI_NONE
<i>MsgSeqNumber</i>	1
<i>Offset</i>	0
<i>MsgFlags</i>	MQMF_NONE
<i>OriginalLength</i>	MQOL_UNDEFINED

- Windowsüzerinde, Micro Focus COBOL için MQENC_NATIVE değeri C değerinden farklıdır. Ayrı ileti tanımlayıcısındaki *Encoding* alanındaki değer, bu ortamlardaki C için her zaman değerdir; bu değer ondalık olarak 546 'dır. MQXQH yapısındaki tamsayı alanları da bu değere (yerel Intel kodlaması) karşılık gelen kodlamadır.

Katıştırılmış ileti tanımlayıcısındaki alanlar

Gömülü ileti tanımlayıcısındaki alanlar, aşağıdakiler dışında, MQPUT ya da MQPUT1 çağrısının **MsgDesc** değıştirgesiyle aynı değerlere sahiptir:

- *Version* alanında her zaman MQMD_VERSION_1değeri bulunur.
- *Priority* alanı MQPRI_PRIORITY_AS_Q_DEF değerine sahipse, kuyruğun **DefPriority** özniteliğinin değeriyle değıştirilir.
- *Persistence* alanı MQPER_PERSISTENCE_AS_Q_DEF değerine sahipse, kuyruğun **DefPersistence** özniteliğinin değeriyle değıştirilir.

- *MsgId* alanında MQMI_NONE değeri varsa ya da MQPMO_NEW_MSG_ID seçeneği belirlendiye ya da ileti bir dağıtım listesi iletisiyse, *MsgId* yerine kuyruk yöneticisi tarafından oluşturulan yeni bir ileti tanıtıcısı konur.

Bir dağıtım listesi iletisi, farklı iletim kuyruklarına yerleştirilen daha küçük dağıtım listesi iletilerine bölündüğünde, yeni yerleşik ileti tanımlayıcılarının her birindeki *MsgId* alanı, özgün dağıtım listesi iletisindekiyle aynıdır.

- MQPMO_NEW_CORREL_ID seçeneği belirtildiyse, *CorrelId* kuyruk yöneticisi tarafından oluşturulan yeni bir ilinti tanıtıcısıyla değiştirilir.
- Bağlam alanları, **PutMsgOpts** değiştirgesinde belirtilen MQPMO_*_CONTEXT seçenekleriyle belirtildiği şekilde ayarlanır; bağlam alanları şunlardır:
 - *AccountingToken*
 - *ApplIdentityData*
 - *ApplOriginData*
 - *PutApplName*
 - *PutApplType*
 - *PutDate*
 - *PutTime*
 - *UserIdentifier*
- version-2 alanları (varsa) MQMD ' den kaldırılır ve version-2 alanlarından birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa MQMDE yapısına taşınır.

Uzak kuyruklara ileti konulması

Bir uygulama uzak kuyruğa bir ileti koyduğunda (uzak kuyruğun adını doğrudan belirterek ya da uzak kuyruğun yerel tanımlamasını kullanarak), yerel kuyruk yöneticisi:

- Yerleşik ileti tanımlayıcısını içeren bir MQXQH yapısı yaratır
- Gerekliyse ve önceden yoksa bir MQMDE ekler
- Uygulama ileti verilerini ekler
- İletiyi uygun bir iletim kuyruğuna yerleştirir

İletilerin doğrudan iletim kuyruklarına yerleştirilmesi

Bir uygulama, iletiyi doğrudan bir iletim kuyruğuna da yerleştirebilir. Bu durumda uygulama, uygulama iletisi verilerine bir MQXQH yapısıyla önek eklemeli ve alanları uygun değerlerle kullanıma hazırlamalıdır. Buna ek olarak, MQPUT ya da MQPUT1 çağrısının **MsgDesc** değiştirgesindeki *Format* alanı MQFMT_XMIT_Q_HEADER değerine sahip olmalıdır.

Uygulama tarafından yaratılan MQXQH yapısındaki karakter verileri, yerel kuyruk yöneticisinin (**CodedCharSetId** kuyruk yöneticisi özneliği tarafından tanımlanan) karakter kümesinde olmalı ve tamsayı verileri yerel makine kodlamasında bulunmalıdır. Buna ek olarak, MQXQH yapısındaki karakter verileri, alanın tanımlı uzunluğuna kadar boşluklarla doldurulacaktır; kuyruk yöneticisi, boş ve sonraki karakterleri MQXQH yapısında boşluklara dönüştürmediği için, veriler boş bir karakter kullanılarak zamanından önce sona erdirilmemelidir.

Ancak, kuyruk yöneticisi bir MQXQH yapısının var olduğunu ya da alanlar için geçerli değerlerin belirtildiğini denetlemez.

Uygulamalar iletilerini doğrudan SYSTEM.CLUSTER.TRANSMIT.QUEUE.

İletim kuyruklarından ileti alma

Bir iletim kuyruğundan ileti alan uygulamaların, MQXQH yapısındaki bilgileri uygun şekilde işlemesi gerekir. Uygulama iletisi verilerinin başında MQXQH yapısının varlığı, MQGET çağrısının **MsgDesc**

değiştirgesindeki *Format* alanında döndürülen MQFMT_XMIT_Q_HEADER değeriyle gösterilir. **MsgDesc** değiştirgesindeki *CodedCharSetId* ve *Encoding* alanlarında döndürülen değerler, MQXQH yapısındaki karakter kümesi ve tamsayı verilerinin kodlamasını gösterir. Uygulama iletisi verilerinin karakter kümesi ve kodlaması, yerleşik ileti tanımlayıcısındaki *CodedCharSetId* ve *Encoding* alanları tarafından tanımlanır.

MQXQH için StrucId (MQCHAR4)

Bu, iletim kuyruğu üstbilgi yapısının yapı tanıttıcısıdır. Her zaman bir giriş alanıdır. Değeri MQXQH_STRUC_ID.

Değer şöyle olmalıdır:

MQXQH_STRUC_ID

İletim kuyruğu üstbilgi yapısına ilişkin tanıttıcı.

C programlama dili için, MQXQH_STRUC_ID_ARRAY değişmezi de tanımlanır. Bu değer MQXQH_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

MQXQH için Sürüm (MQLONG)

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MQXQH_VERSION_1

İletim kuyruğu üstbilgi yapısına ilişkin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQXQH_CURRENT_VERSION

İletim kuyruğu üstbilgi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri MQXQH_VERSION_1' dir.

MQXQH için RemoteQName (MQCHAR48)

Bu, iletinin görünen nihai hedefi olan ileti kuyruğunun adıdır (örneğin, bu kuyruk *RemoteQMgrName* adresinde başka bir uzak kuyruğun yerel tanımlaması olarak tanımlandıysa, bu, nihai hedef olmadığını kanıtlayabilir).

İleti bir dağıtım listesi iletisiyse (yani, yerleşik ileti tanımlayıcısındaki *Format* alanı MQFMT_DIST_HEADER ise), *RemoteQName* boş olur.

Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

RemoteQMgrMQXQH için Ad (MQCHAR48)

Bu, iletinin görünen nihai hedefi olan kuyruğun iyisi olan kuyruk yöneticisinin ya da kuyruk paylaşım grubunun adıdır.

İleti bir dağıtım listesi iletisiyse, *RemoteQMgrName* boş olur.

Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir.

MQXQH için MsgDesc (MQMD1)

Bu, yerleşik ileti tanımlayıcısıdır ve ileti uzak kuyruğa ilk yerleştirildiğinde MQPUT ya da MQPUT1 çağrısında **MsgDesc** değiştirgesi olarak belirtilen ileti tanımlayıcı MQMD ' nin yakın bir kopyasıdır.


Not: Bu bir version-1 MQMD 'dir.

Bu yapıdaki alanların başlangıç değerleri, MQMD yapısındaki değerlerle aynıdır.

İşlev çağrıları

Bu kısım, olası tüm MQI çağrılarına ilişkin bilgi verir. Farklı çağrılarının her biri için, olası her dil için açıklamalar, sözdizimi, parametre bilgileri, kullanım notları ve dil çağrıları verilir.

İlgili başvurular

 MQI çağrılarında CEDF çıkışı örnekleri

Arama açıklamaları

Bu kısım MQI çağrılarını açıklar.

- [“MQBACK-Geri dönüş değişiklikleri” sayfa 621](#)
- [“MQBEGIN-İş birimini başlat” sayfa 625](#)
- [“MQBUFMH-Arabelleği ileti tanıtıcısına dönüştür” sayfa 628](#)
- [“MQCB-Geri çağırma yönet” sayfa 632](#)
- [“MQCB_FUNCTION-Geri çağırma işlevi” sayfa 641](#)
- [“MQCLOSE-Nesneyi kapat” sayfa 642](#)
- [“MQCMIT-Değişiklikleri kesinleştir” sayfa 650](#)
- [“MQCONN-Bağlantı kuyruğu yöneticisi” sayfa 654](#)
- [“MQCONNX-Connect kuyruk yöneticisi \(genişletilmiş\)” sayfa 661](#)
- [“MQCRTMH-İleti tanıtıcısı yarat” sayfa 666](#)
- [“MQCTL-Denetim geri çağrılar” sayfa 670](#)
- [“MQDISC-Kuyruk yöneticisinin bağlantısını kes” sayfa 676](#)
- [“MQDLTMH-İleti silme tanıtıcısı” sayfa 680](#)
- [“MQDLTMP-İleti özelliğini sil” sayfa 682](#)
- [“MQGET-İletiyi al” sayfa 684](#)
- [“MQINQ-Nesne özniteliklerini sorgu” sayfa 696](#)
- [“MQINQMP-İleti sorgusuna ilişkin özellik” sayfa 715](#)
- [“MQMHBUF-İleti tanıtıcısını arabelleğe dönüştür” sayfa 721](#)
- [“MQOPEN-Nesneyi aç” sayfa 725](#)
- [“MQPUT-Put iletisi” sayfa 742](#)
- [“MQPUT1 -Bir ileti koyun” sayfa 755](#)
- [“MQSET-Nesne özniteliklerini ayarla” sayfa 765](#)
- [“MQSETMP-İleti özelliğini ayarla” sayfa 771](#)
- [“MQSTAT-Durum bilgilerini al” sayfa 776](#)
- [“MQMHBUF-İleti tanıtıcısını arabelleğe dönüştür” sayfa 721](#)
- [“MQSUB-Aboneliğin kaydedili” sayfa 779](#)
- [“MQSUBRQ-Abonelik isteği” sayfa 786](#)

UNIX platformlarında *man* sayfaları biçiminde çevrimiçi yardım bu çağrılar için sağlanır.

Not: Veri dönüştürme, MQXCNCV ve MQ_DATA_CONV_EXIT ile ilişkili çağrılar [“Veri dönüştürme çıkışı” sayfa 895](#).

Çağrı açıklamalarında kullanılan kurallar

Her çağrı için, bu konular derlemesi, programlama dilinden bağımsız bir biçimde çağrı parametrelerinin ve kullanımının açıklamasını verir. Bunu, desteklenen programlama dillerinin her birinde çağrı tipik çağrıları ve parametrelerinin tipik bildirimleri izler.

Önemli: IBM MQ API çağrılarını kodlarken, ilgili tüm parametrelerin (aşağıdaki bölümlerde açıklandığı gibi) sağlandığından emin olmanız gerekir. Bunun yapılmaması öngörülemez sonuçlar üretebilir.

Her aramanın açıklaması aşağıdaki bölümleri içerir:

Arama adı

Arama adının ardından, aramanın amacına ilişkin kısa bir açıklama gelir.

Parametreler

Her değiştirge için, adı parantez () içindeki veri tipi izler. ve aşağıdakilerden biri:

Giriş

Arama yaparken parametrede bilgi girersiniz.

output

Çağrı tamamlandığında ya da başarısız olduğunda kuyruk yöneticisi parametredeki bilgileri döndürür.

giriş/çıkış

Çağrıyı yaptığınızda parametrede bilgi girersiniz ve çağrı tamamlandığında ya da başarısız olduğunda kuyruk yöneticisi bilgileri değiştirir.

Örneğin:

Compcode (MQLONG)-çıkış

Bazı durumlarda, veri tipi bir yapıdır. Her durumda, "[Temel veri tipleri](#)" sayfa 234 içinde veri tipi ya da yapısı hakkında daha fazla bilgi vardır.

Her çağrıdaki son iki parametre bir tamamlanma kodu ve bir neden kodudur. Tamamlanma kodu, aramanın başarıyla mı, kısmen mi, yoksa hiç mi tamamlandığını gösterir. Neden kodunda kısmi başarı ya da aramanın başarısız olmasıyla ilgili daha fazla bilgi verilmiştir. Her tamamlanma ve neden koduyla ilgili daha fazla bilgi için bkz. "[Dönüş kodları](#)" sayfa 863.

Kullanım notları

Aramayı nasıl kullanacağınızı ve kullanımıyla ilgili kısıtlamaları açıklayan ek bilgiler.

Çevirici dili çağırma

Çevirici dilde çağırmanın tipik çağırımı ve parametrelerinin bildirilmesi.

C çağırısı

Çağrıyı tipik olarak çağırma ve parametrelerinin C olarak bildirilmesi.

COBOL çağırısı

COBOL dilinde, çağırım ve parametrelerinin bildirilmesi için tipik bir çağrı.

PL/I çağırısı

PL/I içinde, çağırmanın tipik çağırılması ve parametrelerinin bildirilmesi.

Tüm parametreler başvuru yoluyla iletilir.

Visual Basic çağırısı

Visual Basic 'de çağırmanın tipik çağırması ve parametrelerinin bildirilmesi.

Diğer gösterim kuralları şunlardır:

Değişmezler

Değişmezlerin adları büyük harfle gösterilir; örneğin, MQOO_OUTPUT. Öneki aynı olan bir sabitler kümesi şu şekilde gösterilir: MQIA_*. Bir değişmezin değeri için bkz. "[Değişmezler](#)" sayfa 61 .

Diziler

Bazı çağrılarda, parametreler sabit büyüklüklere sahip olmayan karakter dizilerinden oluşan dizilerdir. Bu parametrelerin açıklamalarında, küçük harf n değeri sayısal bir sabiti temsil eder. Bu değiştirgeye ilişkin bildirim kodladığınızda, n değerini, gerek duyduğunuz sayısal değerle değiştirin.

C dilindeki aramaların kullanılması

Değer temelinde, *yalnızca giriş* ve MQHCONN, MQHOBJ, MQHMSG ya da MQLONG tipinde değiştirgeler geçirilir. Diğer tüm parametreler için, parametrenin *address* değeri geçirilir.

Bir işlevi her çağırıldığında adres tarafından geçirilen tüm parametreleri belirtmenize gerek yoktur. Belirli bir parametreye gerek duymadığınız durumlarda, parametre verilerinin adresi yerine, işlev çağırısında parametre olarak boş değerli bir gösterge belirtin. Bunun mümkün olduğu parametreler, çağrı açıklamalarında tanımlanır.

Çağrı değeri olarak parametre döndürülmez; C terminolojisinde bu, tüm çağrılarının voiddöndürdüğü anlamına gelir.

Arabellek deęiřtirgesinin bildirilmesi

MQGET, **MQPUT** ve **MQPUT1** çağrılarının her biri, tanımlanmamış bir veri tipine sahip bir parametreye sahiptir: *Buffer* parametresi. Uygulamanın ileti verilerini göndermek ve almak için bu parametreyi kullanın.

Bu sıralamaya ilişkin parametreler, C örneklerinde MQBYTE dizileri olarak gösterilir. Parametreleri bu şekilde bildirebilirsiniz, ancak genellikle bunları iletideki verilerin düzenini tanımlayan belirli bir yapı olarak bildirmek daha uygundur. İşlev prototipi, parametreyi işaretçi-geçersiz olarak bildirir; böylece, çağrı çağrısında parametre olarak herhangi bir veri türünün adresini belirtebilirsiniz.

Pointer-to-void, tanımlanmamış biçimde verilere yönelik bir işaretçidir. Şu şekilde tanımlanır:

```
typedef void *PMQVOID;
```

MQBACK-Geri dönüş deęişiklikleri

MQBACK çağrısı kuyruk yöneticisine, son eşitleme noktasından bu yana oluşan tüm iletilerin ve iletilerin geriletildiğini gösterir.

Bir iş biriminin parçası olarak konan iletiler silinir; iş biriminin bir parçası olarak alınan iletiler kuyrukta yeniden kullanılır.

- z/OS' da bu çağrı yalnızca toplu iş programları (IMS toplu DL/I programları da içinde olmak üzere) tarafından kullanılır.

Sözdizimi

MQBACK (*Hconn*, *Compcode*, *Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

Düzenleme Kodu

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_OUTCOME_PENDING

(2124, X'84C') Geri alma işleminin sonucu beklemede.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CF_STRUC_IN_USE

(2346, X'92A') Coupling-facility yapısı kullanımda.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_ENVIRONMENT_ERROR

(2012, X'7DC') Çağrısı ortamda geçerli değil.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_OBJECT_HASARLI

(2101, X'835 ') Nesne zarar gördü.

MQRC_OUTCOME_MIXED

(2123, X'84B') Kesinleştirme ya da geri yükleme işleminin sonucu karışık.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_STORAGE_MEDIUM_FULL

(2192, X'890 ') Dış depolama ortamı dolu.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlara ilişkin ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#)

Kullanım notları

1. Bu çağrıyı yalnızca kuyruk yöneticisinin kendisi iş birimini koordine ettiğinde kullanabilirsiniz. Bu şöyle olabilir:
 - Değişikliklerin yalnızca MQ kaynaklarını etkilediği yerel bir iş birimi.
 - Değişikliklerin diğer kaynak yöneticilerine ait kaynakları etkileyebileceği ve MQ kaynaklarını etkileyebileceği genel bir iş birimi.Yerel ve genel iş birimleriyle ilgili daha fazla ayrıntı için bkz. [“MQBEGIN-İş birimini başlat” sayfa 625.](#)
2. Kuyruk yöneticisinin iş birimini koordine etmediği ortamlarda, MQBACK yerine uygun geri alma çağrılarını kullanın. Ortam, uygulamanın olağandışı sonlanmasından kaynaklanan örtük bir geri dönüşü de destekleyebilir.
 - z/OS' da aşağıdaki çağrıları kullanın:
 - İş birimi yalnızca MQ kaynaklarını etkiliyorsa, toplu iş programları (IMS toplu DL/I programları da içinde olmak üzere) MQBACK çağrılarını kullanabilir. Ancak, iş birimi hem MQ kaynaklarını, hem de

diğer kaynak yöneticilerine ait kaynakları etkiliyorsa (örneğin, Db2), z/OS Recoverable Resource Service (RRS) tarafından sağlanan SRRBACK çağrısına bakın. SRRBACK çağrısı, RRS koordinasyonu için etkinleştirilen kaynak yöneticilerine ait kaynaklarda yapılan değişiklikleri geri çevirir.

- CICS uygulamaları, iş birimini geri almak için EXEC CICS SYNCPOINT ROLLBACK komutunu kullanmalıdır. CICS uygulamaları için MQBACK çağrılarını kullanmayın.
 - IMS uygulamaları (toplu DL/I programlarından başka), iş birimini geri almak için ROLB gibi IMS çağrılarını kullanmalıdır. IMS uygulamaları (toplu DL/I programları dışında) için MQBACK çağrılarını kullanmayın.
 - IBM işletim sistemi üzerinde, kuyruk yöneticisi tarafından eşgüdümlü yerel iş birimleri için bu çağrıyı kullanın. Bu, iş düzeyinde bir kesinleştirme tanımının var olmaması gerektiği, yani iş için **CMTSCOPE (*JOB)** parametresini içeren STRCMTCTL komutunun yayınlanmamış olması gerektiği anlamına gelir.
3. Bir uygulama, bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması uygulamanın olağan şekilde mi, yoksa olağandışı olarak mı bittiğine bağlıdır. Daha fazla ayrıntı için [“MQDISC-Kuyruk yöneticisinin bağlantısını kes” sayfa 676](#) içindeki kullanım notlarına bakın.
4. Bir uygulama, mantıksal ileti gruplarına ya da bölümlerine ileti girdiğinde ya da aldığı anda, kuyruk yöneticisi, son başarılı MQPUT ve MQGET çağrılarına ilişkin ileti grubu ve mantıksal iletiyle ilgili bilgileri saklar. Bu bilgiler kuyruk tanıtıcısıyla ilişkilidir ve aşağıdaki gibi şeyleri içerir:
- MQMD 'deki *GroupId*, *MsgSeqNumber*, *Offset* ve *MsgFlags* alanlarının değerleri.
 - İletinin bir iş biriminin parçası olup olmadığı.
 - MQPUT çağrısı için: İletinin kalıcı mı, kalıcı mı olduğu.
- Kuyruk yöneticisi, aşağıdakilerin her biri için bir küme olmak üzere üç grup ve bölüm bilgisi kümesini tutar:
- Son başarılı MQPUT çağrısı (bu, bir iş biriminin parçası olabilir).
 - Kuyruktan bir iletiyi kaldıran son başarılı MQGET çağrısı (bu, bir iş biriminin parçası olabilir).
 - Kuyruktaki bir iletiye göz atan son başarılı MQGET çağrısı (bu, bir iş biriminin parçası olamaz).
5. MQGET çağrısıyla ilişkili bilgiler, yürürlükteki iş biriminde o kuyruk tanıtıcısı için ilk başarılı MQGET çağrıdan önce sahip olduğu değere geri yüklenir.
- İş birimi başlatıldıktan sonra uygulama tarafından güncellenen, ancak iş birimi kapsamı dışında kalan kuyruklarda, iş birimi getirildiyse, grup ve bölüm bilgileri geri yüklenmez.
- Bir iş birimi getirildiğinde grup ve bölüm bilgilerinin önceki değerine geri yüklenmesi, uygulamanın çeşitli iş birimlerine birçok kesim içeren büyük bir ileti grubu ya da büyük bir mantıksal ileti yaymasına ve iş birimlerinden birinin başarısız olması durumunda ileti grubunda ya da mantıksal iletide doğru noktada yeniden başlamasına olanak sağlar.
- Yerel kuyruk yöneticisinde yalnızca sınırlı kuyruk saklama alanı varsa, birkaç iş birimi kullanılması avantajlı olabilir. Ancak, bir sistem arızası ortaya çıkarsa, uygulamanın iletileri doğru noktada yeniden koymayı ya da almayı yeniden başlatabilmesi için yeterli bilgiyi koruması gerekir.
- Bir sistem hatasından sonra doğru noktada nasıl yeniden başlatılacağına ilişkin ayrıntılar için [“MQPMO-İleti koyma seçenekleri” sayfa 496](#) içinde açıklanan MQPMO_LOGICAL_ORDER seçeneğine ve [“MQGMO-Get-message seçenekleri” sayfa 368](#) içinde açıklanan MQGMO_LOGICAL_ORDER seçeneğine bakın.
- Geri kalan kullanım notları yalnızca kuyruk yöneticisi iş birimlerini koordine ettiği için geçerlidir.
6. Bir iş birimi, bağlantı tanıtıcısıyla aynı kapsama sahiptir. Belirli bir iş birimini etkileyen tüm MQ çağrıları aynı bağlantı tanıtıcısı kullanılarak gerçekleştirilmelidir. Farklı bir bağlantı tanıtıcısı kullanılarak verilen çağrılar (örneğin, başka bir uygulama tarafından verilen çağrılar) farklı bir iş birimini etkiler. Bağlantı tanıtıcılarının kapsamıyla ilgili bilgi için [“MQCONN-Bağlantı kuyruğu yöneticisi” sayfa 654](#) içinde açıklanan **Hconn** değiştirgesine bakın.
7. Yalnızca yürürlükteki iş biriminin bir parçası olarak konan ya da alınan iletiler bu çağrıdan etkilenir.

8. Bir iş birimi içinde MQGET, MQPUT ya da MQPUT1 çağrılarını veren, ancak hiçbir zaman kesinleştirme ya da geriletme çağrısı yapmayan uzun süreli bir uygulama, kuyrukları diğer uygulamalar tarafından kullanılmayan iletilerle doldurabilir. Bu olasılığa karşı koruma sağlamak için, denetiminin **MaxUncommittedMsgs** kuyruk yöneticisi özneliğini, kuyrukları dolduran, ancak beklenen ileti sistemi uygulamalarının düzgün çalışmasını sağlayacak kadar yüksek olan kaçak uygulamaları önleyecek kadar düşük bir değere ayarlaması gerekir.

C çağrısı

```
MQBACK (Hconn, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
MQHCONN  Hconn;      /* Connection handle */
MQLONG   CompCode;  /* Completion code */
MQLONG   Reason;    /* Reason code qualifying CompCode */
```

COBOL çağrısı

```
CALL 'MQBACK' USING HCONN, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağrısı

```
call MQBACK (Hconn, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl CompCode   fixed bin(31); /* Completion code */
dcl Reason     fixed bin(31); /* Reason code qualifying CompCode */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL MQBACK,(HCONN,COMPCODE,REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
HCONN      DS F Connection handle
COMPCODE   DS F Completion code
REASON     DS F Reason code qualifying COMPCODE
```

Visual Basic çağrısı

```
MQBACK Hconn, CompCode, Reason
```


Değiştirgeleri aşağıdaki gibi bildirin:

```
Dim Hconn As Long 'Connection handle'  
Dim CompCode As Long 'Completion code'  
Dim Reason As Long 'Reason code qualifying CompCode'
```

MQBEGIN-İş birimini başlat

MQBEGIN çağrısı, kuyruk yöneticisi tarafından koordine edilen ve dış kaynak yöneticilerini içerebilecek bir iş birimi başlatır.

Sözdizimi

MQBEGIN (*Hconn*, *BeginOptions*, *Compcode*, *Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıttıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

Hconn , paylaşılmayan bir bağlantı tanıttıcısı olmalıdır. Paylaşılan bir bağlantı tanıttıcısı belirtilirse, çağrı MQRC_HCONN_ERROR neden koduyla başarısız olur. Paylaşılan ve paylaşılmayan tanıttıcıları hakkında daha fazla bilgi için [“MQCNO-Bağlantı seçenekleri” sayfa 316](#) içindeki MQCNO_HANDLE_SHARE_* seçeneklerinin açıklamasına bakın.

BeginOptions

Tip: MQBO-giriş/çıkış

Bunlar, [“MQBO-Başlangıç seçenekleri” sayfa 279](#) içinde açıklandığı gibi MQBEGIN işlemini denetleyen seçeneklerdir.

Herhangi bir seçenek gerekmiyorsa, C ya da S/390 çevirici dilinde yazılan programlar, bir MQBO yapısının adresini belirtmek yerine boş bir değiştirge adresi belirtebilir.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_NO_EXTERNAL_KATILIMCILAR

(2121, X'849 ') Katılımcı kaynak yöneticisi kaydedilmedi.

MQRC_PARTICIPANT_NOT_VAR

(2122, X'84A') Kaynak yöneticisine katılım sağlanmıyor.

CompCode MQCC_FAILED ise:

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_BO_ERROR

(2134, X'856 ') Başlangıç seçenekleri yapısı geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_ENVIRONMENT_ERROR

(2012, X'7DC') Çağrısı ortamda geçerli değil.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_OPTIONS_HATA

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

MQRC_UOW_IN_PROGRESS

(2128, X'850 ') İş birimi zaten başlatıldı.

Bu kodlarla ilgili daha fazla bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. Kuyruk yöneticisi tarafından eşgüdümlü bir iş birimi başlatmak ve diğer kaynak yöneticilerinin sahip olduğu kaynaklarda değişiklik yapılmasını içerebilecek bir iş birimi başlatmak için MQBEGIN çağrısı kullanın. Kuyruk yöneticisi üç tip iş birimini destekler:
 - **Kuyruk yöneticisi eşgüdümlü yerel iş birimi:** Kuyruk yöneticisinin katılan tek kaynak yöneticisi olduğu ve kuyruk yöneticisinin iş birimi eşgüdümçüsü olarak hareket ettiği iş birimi.
 - Bu iş birimi tipini başlatmak için, ilk MQPUT, MQPUT1 ya da iş birimindeki MQGET çağrısında MQPMO_SYNCPOINT ya da MQGMO_SYNCPOINT seçeneğini belirtin.
 - Bu iş birimini kesinleştirmek ya da geri almak için MQCMIT ya da MQBACK çağrılarını kullanın.
 - **Kuyruk yöneticisi eşgüdümlü genel iş birimi:** Kuyruk yöneticisinin hem MQ kaynakları için hem de diğer kaynak yöneticilerine ait kaynaklar için iş birimi eşgüdümçüsü olarak hareket ettiği iş birimi. Bu kaynak yöneticileri, iş birimindeki kaynaklarda yapılan tüm değişikliklerin kesinleştirilmesini ya da birlikte gerilettilmesini sağlamak için kuyruk yöneticisiyle işbirliği yapar.
 - Bu iş birimi tipini başlatmak için MQBEGIN çağrısına bakın.
 - Bu iş birimi tipini kesinleştirmek ya da geri almak için MQCMIT ve MQBACK çağrılarını kullanın.
 - **Harici eşgüdümlü genel iş birimi:** Kuyruk yöneticisinin katılımcı olduğu, ancak kuyruk yöneticisinin iş birimi eşgüdümçüsü olarak çalışmadığı iş birimi. Bunun yerine, kuyruk yöneticisinin birlikte çalıştığı bir dış iş birimi koordinatörü vardır.

- Bu iş birimi tipini başlatmak için, dış iş birimi koordinatörü tarafından sağlanan ilgili çağrıyı kullanın.
İş birimini başlatmayı denemek için MQBEGIN çağrısı kullanılırsa, çağrı MQRC_ENVIRONMENT_ERROR neden koduyla başarısız olur.
 - Bu iş birimi tipini kesinleştirmek ya da geri almak için, dış iş birimi eşgüdümçüsü tarafından sağlanan kesinleştirme ve geri çağırma çağrılarını kullanın.
İş birimini kesinleştirmek ya da geri almak için MQCMIT ya da MQBACK çağrısı kullanırsanız, çağrı MQRC_ENVIRONMENT_ERROR neden koduyla başarısız olur.
2. Uygulama, bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması, uygulamanın olağan şekilde mi, yoksa olağandışı olarak mı bittiğine bağlıdır. Daha fazla ayrıntı için [“MQDISC-Kuyruk yöneticisinin bağlantısını kes” sayfa 676](#) içindeki kullanım notlarına bakın.
 3. Bir uygulama aynı anda yalnızca bir iş birimine katılabilir. MQBEGIN çağrısı, uygulama için var olan bir iş birimi varsa, neden kodu MQRC_UOW_IN_PROGRESS ile başarısız olur.
 4. MQBEGIN çağrısı, MQ MQI istemcisi ortamında geçerli değil. Çağrıyı kullanma girişimi MQRC_ENVIRONMENT_ERROR neden koduyla başarısız oldu.
 5. Kuyruk yöneticisi genel iş birimleri için iş birimi eşgüdümçüsü olarak hareket ederken, iş birimine katılabilecek kaynak yöneticileri kuyruk yöneticisi yapılandırma dosyasında tanımlanır.
 6. IBM işletim sisteminde, üç iş birimi tipi aşağıdaki gibi desteklenir:
 - **Kuyruk yöneticisi eşgüdümlü yerel iş birimi** yalnızca iş düzeyinde bir kesinleştirme tanımı yoksa, yani iş için **CMTSCOPE(*JOB)** parametresiyle birlikte STRCMTCTL komutu verilmemelidir.
 - **Kuyruk yöneticisi-eşgüdümlü genel iş birimi** desteklenmiyor.
 - **Harici eşgüdümlü genel iş birimi** yalnızca iş düzeyinde bir kesinleştirme tanımı varsa, yani iş için **CMTSCOPE(*JOB)** parametresiyle birlikte STRCMTCTL komutu verilmelidir. Bu yapıldıysa, IBM i COMMIT ve ROLLBACK işlemleri MQ kaynaklarının yanı sıra, katılan diğer kaynak yöneticilerine ait kaynaklar için de geçerlidir.

C çağrısı

```
MQBEGIN (Hconn, &BeginOptions, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
MQHCONN  Hconn;           /* Connection handle */
MQBO     BeginOptions;   /* Options that control the action of MQBEGIN */
MQLONG   CompCode;       /* Completion code */
MQLONG   Reason;         /* Reason code qualifying CompCode */
```

COBOL çağrısı

```
CALL 'MQBEGIN' USING HCONN, BEGINOPTIONS, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağırısı

```
call MQBEGIN (Hconn, BeginOptions, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

Visual Basic çağırısı

```
MQBEGIN Hconn, BeginOptions, CompCode, Reason
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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-Arabelleği ileti tanıtıcısına dönüştür

MQBUFMH işlev çağırısı, bir arabelleği ileti tanıtıcısına dönüştürür ve MQMHBUF çağırısının tersidir.

Bu çağrı, arabellekte bir ileti tanımlayıcısı ve MQRFH2 özelliklerini alır ve bunları bir ileti tanıtıcısı aracılığıyla kullanılabilir kılar. İleti verilerindeki MQRFH2 özellikleri isteğe bağlı olarak kaldırılır. Gerekliyse, ileti tanımlayıcısının *Encoding, CodedCharSetIdve Format* alanları, özellikler kaldırıldıktan sonra arabelleğin içeriğini doğru şekilde tanımlamak için güncellenir.

Sözdizimi

MQBUFMH (*Hconn, Hmsg, BufMsgHOpts, MsgDesc, BufferLength, Buffer, DataLength, Compcode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. **Hconn** değeri, **Hmsg** değiştirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eşleşmelidir.

İleti tanıtıcısı MQHC_UNASSOCIATED_HCONN kullanılarak yaratıldıysa, arabelleği ileti tanıtıcısına dönüştüren iş parçacığında geçerli bir bağlantı kurulmalıdır. Geçerli bir bağlantı kurulmazsa, çağrı MQRC_CONNECTION_BROKEN ile başarısız olur.

İt

Tip: MQHMQSG-giriş

Bu, arabelleğin gerekli olduğu ileti tanıtıcısıdır. Değer, önceki bir MQCRTMH çağırısı tarafından döndürüldü.

BufMsgHOpts

Tip: MQBMHO-giriş

MQBMHO yapısı, uygulamaların, ileti tanıtıcılarının arabelleklerden nasıl üretildiğini denetleyen seçenekleri belirtmesine olanak sağlar.

Ayrıntılar için bkz. [“MQBMHO-İleti işleme seçenekleri için arabellek” sayfa 275.](#)

MsgDesc

Tip: MQMD-giriş/çıkış

MsgDesc yapısı, ileti tanımlayıcı özelliklerini içerir ve arabellek alanının içeriğini açıklar.

Çağrı çıkışında, özellikler isteğe bağlı olarak arabellek alanından kaldırılır ve bu durumda, ileti tanımlayıcısı arabellek alanını doğru şekilde tanımlayacak şekilde güncellenir.

Bu yapıdaki veriler, uygulamanın karakter kümesi ve kodlamasında olmalıdır.

BufferLength

Tip: MQLONG-input

BufferLength, Arabellek alanının bayt cinsinden uzunluğudur.

Sıfır baytlık bir *BufferLength* geçerli ve arabellek alanının veri içermediğini gösterir.

Arabellek

Tip: MQBYTEExBufferLength-input/output

Bunlar, "MQBEGIN-İş birimini başlat" sayfa 625'te açıklandığı gibi MQBEGIN işlemini denetleyen seçeneklerdir.

Buffer, ileti arabelleğini içeren alanı tanımlar. Çoğu veri için, arabelleği 4 baytlık bir sınırdan hizalayın.

Buffer karakter ya da sayısal veri içeriyorsa, **MsgDesc** parametresindeki *CodedCharSetId* ve *Encoding* alanlarını veriler için uygun değerlere ayarlayın; bu, gerekirse verilerin dönüştürülmesini sağlar.

İleti arabelleğinde özellikler bulunursa, isteğe bağlı olarak bunlar kaldırılır; bunlar daha sonra çağrıdan döndüklerinde ileti tanıtıcısından kullanılabilir olur.

C programlama dilinde parametre, herhangi bir veri tipinin adresinin parametre olarak belirtileceği anlamına gelen bir işaretçi-to-void olarak bildirilir.

BufferLength parametresi sıfırsa, **Buffer** değiştirgesine başvurulamaz; bu durumda, C ya da System/390 çevirici dilinde yazılmış programlar tarafından geçirilen parametre adresi boş olabilir.

DataLength

Tip: MQLONG-output

Özelliklerin kaldırılabilir olduğu arabelleğin bayt cinsinden uzunluğu.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'089C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BMHO_ERROR

(2489, X'09B9') İleti işleme seçeneklerine ilişkin arabellek yapısı geçersiz.

MQRC_BUFFER_ERROR

(2004, X'07D4') Arabellek değiştirgesi geçersiz.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'07D5') Arabellek uzunluğu değiştirgesi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağırısı.

MQRC_CONNECTION_BROKEN

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_HMSG_HATA

(2460, X'099C') İleti tanıtıcısı geçersiz.

MQRC_MD_ERROR

(2026, X'07EA') İleti tanımlayıcısı geçersiz.

MQRC_MSG_HANDLE_IN_USE

(2499, X'09C3') İleti tanıtıcısı kullanımda.

MQRC_OPTIONS_HATA

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_RFH_ERROR

(2334, X'091E') MQRFH2 yapısı geçerli değil.

MQRC_RFH_FORMAT_HATA

(2421, X'0975 ') Özellikleri içeren bir MQRFH2 klasörü ayrıştırılmadı.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

MQBUFMH çağrılarını API çıkışları tarafından engellenemez-bir arabellek uygulama alanında ileti tanıtıcısına dönüştürülür; çağrı kuyruk yöneticisine ulaşmaz.

C çağırısı

```
MQBUFMH (Hconn, Hmsg, &BufMsgHOpts, &MsgDesc, BufferLength, Buffer,  
&DataLength, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağırısı

```
CALL 'MQBUFMH' USING HCONN, HMSG, BUFMSGHOPTS, MSGDESC, BUFFERLENGTH,  
BUFFER, DATALENGTH, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağırısı

```
call MQBUFMH (Hconn, Hmsg, BufMsgHOpts, MsgDesc, BufferLength, Buffer,  
DataLength, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQBUFMH, (HCONN, HMSG, BUFMSGHOPTS, MSGDESC, BUFFERLENGTH, BUFFER,  
DATALENGTH, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

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
COMPCODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMPCODE

MQCB-Geri çağırmaı yönet

MQCB çağırısı, belirtilen nesne tanıtıcısı için bir geri çağırma kaydeder ve geri çağırma üzerinde etkinleştirme ve değışiklikleri denetler.

Geri çağırma, belirli olaylar oluřtuęunda IBM MQ tarafından çağırılan bir kod parçasıdır (devingen olarak bağlanabilen bir işlevin adı olarak ya da işlev göstergesi olarak belirtilir).

Bir istemcide MQCB ve MQCTL kullanabilmek için, kanalın kararlařtırılan **SHARECNV** değıştirgesinin sıfır dıřında bir değeri kabul ettięi bir sunucuya bağlanmanız gerekir.

Tanımlanabilecek geri çağırma tipleri řunlardır:

İleti tüketicisi

Belirtilen seçim ölçütlerini karřılayan bir ileti, bir nesne tanıtıcısında kullanılabilir olduęunda, ileti tüketicisi geri çağırma işlevi çağırılır.

Her nesne tanıtıcısı için tek bir geri çağırma işlevi kaydedilebilir. Tek bir kuyruk birden çok seçim ölçütüyle okunacaksa, kuyruk birden çok kez açılmalı ve her bir tanıtıcıda bir tüketici işlevi kaydedilmelidir.

Olay işleyici

Olay işleyici, tüm geri çağırma ortamını etkileyen kořullar için çağırılır.

Bir olay kořulu (örneğin, bir kuyruk yöneticisi ya da bağlantı durdurma ya da susturma) oluřtuęunda işlev çağırılır.

İşlev, tek bir ileti tüketicisine özgü kořullar (örneğin, MQRC_GET_INHIBITED;) için çağırılmaz; ancak, geri çağırma işlevi olaęan řekilde sona ermezse çağırılır.

Sözdizimi

MQCB (*Hconn, Operation, CallbackDesc, Hobj, MsgDesc, GetMsgOpts, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriř

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağırısı tarafından döndürüldü.

z/OS for CICS uygulamalarında, *MQHC_DEF_HCONN* ' in bu yürütme birimiyle iliřkili bağlantı tanıtıcısını kullanması için ařaęıdaki özel değeri belirtebilirsiniz.

İşlem

Tip: MQLONG-input

Belirtilen nesne tanıtıcısı için tanımlanan geri çağırma sırasında işlenmekte olan işlem. Ařaęıdaki seçeneklerden birini belirtmelisiniz. birden çok seçenek belirtmek için değeri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değeri birleřtirin (programlama dili bit işlemlerini destekliyorsa).

MQOP_REGISTER

Belirtilen nesne tanıtıcısı için geri çağırma işlevini tanımlayın. Bu işlem, çağırılacak işlevi ve kullanılacak seçim ölçütlerini tanımlar.

Nesne tanıtıcısı için önceden tanımlanmış bir geri çağırma işlevi varsa, tanımlama değıştirilir. Geri çağırma değıştirilirken bir hata saptanırsa, işlevin kaydı kaldırılır.

Bir geri çağırma daha önce kayıttan kaldırıldıęı aynı geri çağırma işlevine kaydedilirse, bu bir yerine koyma işlemi olarak işlenir; ilk ya da son çağırılar çağırılmaz.

MQOP_SUSPEND ya da MQOP_RESUME ile MQOP_REGISTER kullanabilirsiniz.

MQOP_DEREGISTER

Nesne tanıtıcısı için ileti tüketimini durdurun ve geri çağırma için seçilenlerden tutamacı kaldırır.

İlişkili tanıtıcı kapatılırsa, bir geri çağırmanın kaydı otomatik olarak kaldırılır.

MQOP_DEREGISTER bir tüketici içinden çağrılırsa ve geri çağırmanın tanımlı bir durdurma çağrısı varsa, bu çağrı tüketiciden döndükten sonra çağrılır.

Bu işlem kayıtlı tüketicisi olmayan bir *Hobj* için yayınlandıysa, çağrı MQRC_CALLBACK_NOT_REGISTERED ile geri döner.

MQOP_SUSPEND

Nesne tanıtıcısı için ileti tüketimini askıya alır.

Bu işlem bir olay işleyiciye uygulanırsa, olay işleyici askıya alınmış durumdayken olayları alamaz ve askıya alınmış durumdayken kaçırılan olaylar, sürdürüldüğünde işleme sağlanmaz.

Askıya alındığında, tüketici işlevi denetim tipi geri çağırılmaları almaya devam eder.

MQOP_RESUME

Nesne tanıtıcısı için iletileri tüketmeye devam edin.

Bu işlem bir olay işleyiciye uygulanırsa, olay işleyici askıya alınmış durumdayken olayları alamaz ve askıya alınmış durumdayken kaçırılan olaylar, sürdürüldüğünde işleme sağlanmaz.

CallbackDesc

Tip: MQCBD-giriş

Bu, uygulama tarafından kaydedilmekte olan geri çağırma işlevini ve kayıt sırasında kullanılan seçenekleri tanımlayan bir yapıdır.

Yapıya ilişkin ayrıntılar için [MQCBD](#) kısmına bakın.

Geri çağırma tanımlayıcısı yalnızca MQOP_REGISTER seçeneği için gereklidir; tanımlayıcı gerekli değilse, geçirilen parametre adresi boş değerli olabilir.

Hobj.

Tip: MQHOBJ-giriş

Bu tanıtıcı, iletinin tüketileceği nesne için oluşturulan erişimi gösterir. Bu, önceki bir [MQOPEN](#) ya da [MQSUB](#) çağrısından (**Hobj** değiştirilmesinde) döndürülen bir tanıtıcıdır.

Olay işleyici yordamı (MQCBT_EVENT_HANDLER) tanımlanırken *Hobj* gerekli değildir ve MQHO_NONE olarak belirtilmelidir.

Bir MQOPEN çağrısından *Hobj* döndürüldüyse, kuyruk aşağıdaki seçeneklerden biriyle ya da daha fazlasıyla açılmış olmalıdır:

- MQOO_INPUT_SHARED
- MQOO_INPUT_EXCLUSIVE
- MQOO_INPUT_AS_Q_DEF
- MQOO_GÖZ AT

MsgDesc

Tip: MQMD-giriş

Bu yapı, gerekli iletinin özniteliklerini ve alınan iletinin özniteliklerini açıklar.

MsgDesc değiştirgesi, tüketicinin gerektirdiği iletilerin özniteliklerini ve ileti tüketicisine geçirilecek MQMD sürümünü tanımlar.

MQMD ' deki *MsgId*, *CorrelId*, *GroupId*, *MsgSeqNumber* ve *Offset* , **GetMsgOpts** değiştirilmesinde belirtilen seçeneklere bağlı olarak ileti seçimi için kullanılır.

MQGMO_CONVERT seçeneğini belirtirseniz, ileti dönüştürme için *Encoding* ve *CodedCharSetId* kullanılır.

Ayrıntılar için bkz. [MQMD](#) .

MsgDesc , MQOP_REGISTER için ve herhangi bir alan için varsayılan dışında değerlere gereksinim duyarsanız kullanılır. *MsgDesc* bir olay işleyici için kullanılmaz.

Tanımlayıcı gerekli değilse, geçirilen parametre adresi boş değerli olabilir.

Çakışan seçicilerle aynı kuyruğa birden çok tüketici kaydedilirse, her ileti için seçilen tüketicinin tanımlanmamış olduğunu unutmayın.

GetMsgSeçmeler

Tip: MQGMO-giriş

GetMsgOpts parametresi, ileti tüketicisinin iletileri nasıl aldığını denetler. Bir MQGET çağrısında kullanıldığında, aşağıdaki durumlar dışında, bu değiştirgenin tüm seçeneklerinin "MQGMO-Get-message seçenekleri" sayfa 368 içinde açıklandığı gibi anlamları vardır:

MQGMO_SET_SIGNAL

Bu seçeneğine izin verilmez.

MQGMO_BROWSE_FIRST, MQGMO_BROWSE_NEXT, MQGMO_MARK_*

Bir göz atma tüketicisine gönderilen iletilerin sırası, bu seçeneklerin birleşimleri tarafından belirlenmiştir. Önemli birleşimler şunlardır:

MQGMO_BROWSE_FIRST

Kuyruktaki ilk ileti tekrar tekrar tüketiciye teslim edilir. Bu, tüketici geri çağırmada iletiyi yıkıcı bir şekilde tükettiğinde kullanışlıdır. Bu seçeneği dikkatli kullanın.

MQGMO_BROWSE_NEXT

Yürürlükteki imleç konumundan kuyruğun sonuna ulaşıncaya kadar, kuyruktaki her ileti tüketiciye verilir.

MQGMO_BROWSE_FIRST + MQGMO_BROWSE_NEXT

İmleç, kuyruğun başlangıcını belirtir. Daha sonra, imleç kuyruğun sonuna ulaşıncaya kadar her ileti tüketiciye verilir.

MQGMO_BROWSE_FIRST + MQGMO_MARK_*

Kuyruğun başından başlayarak, tüketiciye kuyrukta ilk işaretlenmemiş ileti verilir; bu ileti daha sonra bu tüketici için işaretlenir. Bu birleşim, tüketicinin geçerli imleç noktasının arkasına eklenen yeni iletileri alabilmesini sağlar.

MQGMO_BROWSE_NEXT + MQGMO_MARK_*

İmleç konumundan başlayarak, tüketiciye kuyrukta imlenmemiş sonraki ileti verilir; bu ileti daha sonra bu tüketici için işaretlenir. İletiler geçerli imleç konumunun arkasındaki kuyruğa eklenebildiği için bu bileşimi dikkatli bir şekilde kullanın.

MQGMO_BROWSE_FIRST + MQGMO_BROWSE_NEXT + MQGMO_MARK_*

Bu bileşime izin verilmez. Kullanılırsa, çağrı MQRC_OPTIONS_ERROR döndürür.

MQGMO_NO_WAIT, MQGMO_WAIT ve WaitInterval

Bu seçenekler, tüketicinin nasıl çağrılacağını denetler.

MQGMO_NO_WAIT

MQRC_NO_MSG_AVAILABLE ile tüketici hiçbir zaman çağrılmadı. Tüketici yalnızca iletiler ve olaylar için çağrılır.

Sıfır WaitInterval ile MQGMO_WAIT

MQRC_NO_MSG_AVAILABLE kodu, kullanılabilir ileti olmadığında ve tüketici başlatıldığında ya da tüketici son "ileti yok" neden kodundan bu yana en az bir ileti teslim edildiğinde tüketiciye iletilir.

Bu, sıfır bekleme aralığı belirtildiğinde tüketicinin meşgul döngüde yoklama yapmasını önler.

MQGMO_WAIT ve pozitif bir WaitInterval

Tüketici, belirtilen bekleme aralığından sonra MQRC_NO_MSG_AVAILABLE neden koduyla çağrılır. Bu çağrı, tüketiciye iletilerin teslim edilip edilmediğine bakılmaksızın yapılır. Bu, kullanıcının sağlıklı işletim bildirimini ya da toplu iş tipi işlemesi gerçekleştirmesini sağlar.

MQWI_UNLIMITED MQGMO_WAIT ve WaitInterval

Bu, MQRC_NO_MSG_AVAILABLE döndürülmeden önce sonsuz bir bekleme belirler.

MQRC_NO_MSG_AVAILABLE ile tüketici hiçbir zaman çağrılmadı.

GetMsgOpts yalnızca MQOP_REGISTER için ve herhangi bir alan için varsayılan dışında değerler gerekiyorsa kullanılır. *GetMsgOpts* bir olay işleyici için kullanılmaz.

GetMsgOpts gerekli değilse, geçirilen parametre adresi boş değerli olabilir. Bu deęiřtirgenin kullanılması, MQGMO_FAIL_IF_QUIESCING ile birlikte MQGMO_DEFAULT belirtilmesiyle aynıdır.

MQGMO yapısında bir ileti özellikleri tanıtıcısı sağlanırsa, MQGMO yapısında tüketici geri çağırısına geçirilen bir kopya sağlanır. MQCB çağırısından döndüğünüzde, uygulama ileti özellikleri tanıtıcısını silebilir.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Aşağıdaki listede yer alan neden kodları, kuyruk yöneticisinin **Reason** deęiřtirgesi için döndürebileceęi kodlardır.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Baędařtırıcı kullanılmıyor.

MQRC_ADAPTER_CONV_LOAD_ERROR (UYGULAMA_HATASI)

(2133, X'855 ') Veri dönüřtürme hizmetleri modülleri yüklenemiyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Baędařtırıcı hizmet modülü yüklenemiyor.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

(2183, X'887 ') API çıkışı yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluęu parametresi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağırısı.

MQRC_CALLBACK_LINK_HATA

(2487, X'9B7') Geri çağırma tipi alanı yanlış.

MQRC_CALLBACK_NOT_REGISTERED

(2448, X' 990 ') Kayıtlı geri arama olmadığı için kaydı kaldırılmıyor, askıya alınamıyor ya da sürdürülemiyor.

MQRC_CALLBACK_ROUTINE_HATA

(2486, X'9B6') *CallbackFunction* ya da *CallbackName* belirtilmelidir, ancak her ikisi de belirtilmemelidir.

MQRC_CALLBACK_TYPE_HATA

(2483, X'9B3') Geri çağırma tipi alanı yanlış.

MQR_CBD_OPTIONS_HATA
(2484, X'9B4') Yanlış MQR_CBD seçenekleri alanı.

MQR_CICS_WAIT_FAILED
(2140, X'85C') Bekleme isteği CICS tarafından reddedildi.

MQR_CONNECTION_BROKEN
(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQR_CONNECTION_NOT_YETKILI
(2217, X'8A9') Bağlantı için yetkili değil.

MQR_CONNECTION_QUIESCING
(2202, X'89A') Bağlantı susturuyor.

MQR_CONNECTION_DURDURULUYOR
(2203, X'89B') Bağlantı kapatılıyor.

MQR_CORREL_ID_HATA
(2207, X'89F') İlinti-tanıtıcı hatası.

MQR_DATA_LENGTH_HATA
(2010, X'7DA') Veri uzunluğu parametresi geçersiz.

MQR_ENVIRONMENT_ERROR
(2012, X'7DC') Çağrısı ortamda geçerli değil.

MQR_FUNCTION_NOT_SUPPORTED
(2298, X'8FA') İstenen işlev yürürlükteki ortamda kullanılmıyor.

MQR_GET_INHIBE
(2016, X'7E0') Kuyruk için engellendi.

MQR_GLOBAL_UOW_CONFLICT
(2351, X'92F') Genel iş birimleri çakışması.

MQR_GMO_HATA
(2186, X'88A') Get-message seçenekleri yapısı geçersiz.

MQR_HANDLE_IN_USE_FOR_UOW
(2353, X' 931 ') Genel iş birimi için kullanımda.

MQR_HCONN_HATA
(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQR_HOBJ_HATA
(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

MQR_TUTARSIZ_GÖZAT
(2259, X'8D3') Tutarsız göz atma belirtimi.

MQR_TUTARSIZ_UOW
(2245, X'8C5') İş birimi belirtimi tutarsız.

MQR_INVALID_MSG_UNDER_CURSOR
(2246, X'8C6') İmleç altındaki ileti alma için geçerli değil.

MQR_LOCAL_UOW_CONFLICT
(2352, X' 930 ') Genel iş birimi yerel iş birimiyle çakışıyor.

MQR_MATCH_OPTIONS_HATA
(2247, X'8C7') Eşleştirme seçenekleri geçerli değil.

MQR_MAX_MSG_LENGTH_ERROR
(2485, X'9B4') *MaxMsgLength* alanı yanlış.

MQR_MD_ERROR
(2026, X'7EA') İleti tanımlayıcısı geçersiz.

MQR_MODULE_ENTRY_NOT_FOUND
(2497, X'9C1') Belirtilen işlev giriş noktası modülde bulunamadı.

MQRC_MODULE_INVALID

(2496, X'9C0') Modül bulundu, ancak yanlış tipte; 32 bit, 64 bit ya da geçerli bir dinamik bağlantı kitaplığı değil.

MQRC_MODULE_NOT_FOUND

(2495, X'9BF') Modül arama yolunda bulunamadı ya da yükleme yetkisi yok.

MQRC_MSG_SEQ_NUMBER_ERROR (QRC_MSG_SEQ_NUMBER_ERROR)

(2250, X'8CA') İleti sıra numarası geçersiz.

MQRC_MSG_TOKEN_HATA

(2331, X'91B') İleti simgesi kullanımı geçersiz.

MQRC_NO_MSG_AVAILABLE

(2033, X'7F1') İleti yok.

MQRC_NO_MSG_UNDER_CURSOR

(2034, X'7F2') Göz at imleci iletide konumlandırılmadı.

MQRC_NOT_OPEN_FOR_BROWSE

(2036, X'7F4') Kuyruk göz atmak için açık değil.

MQRC_NOT_OPEN_FOR_INPUT

(2037, X'7F5') Kuyruk giriş için açık değil.

MQRC_OBJECT_CHANGED

(2041, X'7F9') Nesne tanımı açıldıktan sonra değiştirildi.

MQRC_OBJECT_HASARLI

(2101, X'835 ') Nesne zarar gördü.

MQRC_OPERATION_HATA

(2206, X'89E') API Çağrısı 'nda işlem kodu yanlış.

MQRC_OPTIONS_HATA

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_PAGESET_HATA

(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRC_Q_SILINDI

(2052, X'804 ') Kuyruk silindi.

MQRC_Q_INDEX_TYPE_HATA

(2394, X'95A') Kuyruk yanlış dizin tipine sahip.

MQRC_Q_MGR_AD_HATASI

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRC_Q_MGR_YOK

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.

MQRC_Q_MGR_QUIESCING

(2161, X'871 ') Kuyruk yöneticisi susturuyor.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_SIGNAL_HENÜZ tamamlanmamış

(2069, X'815 ') Bu tutamaç için bekleyen işaret.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_SUPPRESSED_BY_EXIT

(2109, X'83D') Çağrı çıkış programı tarafından engellendi.

MQRC_SYNCPOINT_LIMIT_ULAŞILDI

(2024, X'7E8') Yürürlükteki iş birimi içinde başka ileti işlenmez.

MQRC_SYNCPOINT_NOT_YOK

(2072, X'818 ') Eşitleme noktası desteği kullanılamıyor.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

MQRC_UOW_ENLISTMENT_HATA

(2354, X' 932 ') Genel iş birimine kayıt başarısız oldu.

MQRC_UOW_MIX_NOT_DESTEKLENMIYOR

(2355, X' 933 ') İş birimi çağrılarının karışımı desteklenmez.

MQRC_UOW_YOK

(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.

MQRC_WAIT_INTERVAL_ERROR

(2090, X'82A') MQGMO' daki bekleme aralığı geçersiz.

MQRC_ERROR_gmo_sürümü

(2256, X'8D0') Yanlış MQGMO sürümü sağlandı.

MQRC_ERROR_md_version

(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. MQCB, kuyrukta var olan belirtilen ölçütlerle eşleşen her ileti için çağrılacak işlemi tanımlamak için kullanılır. İşlem işlendiğinde, ileti kuyruktan kaldırılır ve tanımlı ileti tüketicisine iletilir ya da iletiyi almak için kullanılan bir ileti simgesi sağlanır.
2. MQCB, MQCTL ile tüketime başlamadan önce geri çağırma yordamlarını tanımlamak için kullanılabilir ya da bir geri çağırma yordamından kullanılabilir.
3. MQCB ' yi bir geri çağırma yordamının dışından kullanmak için, önce MQCTL kullanarak ileti tüketimini askıya almanız ve daha sonra tüketimi sürdürmeniz gerekir.
4. MQCB, IMS bağdaştırıcısında desteklenmiyor.

İleti tüketici geri çağırma sırası

Bir tüketiciyi, tüketicinin yaşam çevrimi boyunca temel noktalarda geri çağırma üzere yapılandırabilirsiniz. Örneğin:

- Tüketici ilk kez kaydedildiğinde,
- bağlantı başlatıldığında,
- bağlantı durdurulduğunda ve
- Tüketici bir MQCLOSE tarafından belirtik ya da örtük olarak kayıttan kaldırıldığında.

<i>Çizelge 541. MQCTL fiil tanımlamaları</i>	
Komut	Anlamı
MQCTL (BAŞLAT)	MQOP_START İşlemi kullanılarak MQCTL çağırısı
MQCTL (DURDUR)	MQOP_STOP İşlemi kullanılarak MQCTL çağırısı
MQCTL (BEKLE)	MQOP_START_WAIT İşlemini Kullanarak MQCTL çağırısı

Bu, tüketicinin tüketiciyle ilişkili durumu korumasını sağlar. Bir uygulama tarafından geri çağırma istendiğinde, tüketici çağırma kuralları aşağıdaki gibidir:

KAYDOL

Her zaman geri çağırmanın ilk tipidir.

MQCB (REGISTER) çağrısıyla aynı iş parçacığında her zaman çağrılır.

START

MQCTL (START) komutu ile her zaman zamanuyumlu olarak çağrılır.

- Tüm START geri çağrılar, MQCTL (START) komutu döndürülmeden önce tamamlanır.

THREAD_AFFINITY istenirse, ileti teslimiyle aynı iş parçacığında.

Örneğin, MQCTL (START) sırasında önceki bir geri çağırma MQCTL (STOP) komutu verirse, başlatma ile çağrı garanti edilmez.

DUR

Bağlantı yeniden başlatılıncaya kadar, bu çağrıdan sonra başka ileti ya da olay teslim edilmez.

Uygulama daha önce START ya da ileti ya da olay için çağrıldıysa, STOP garanti edilir.

DEREGISTER

Her zaman geri çağırmanın son türüdür.

Uygulamanızın START ve STOP geri çağrılarında iş parçacığı tabanlı başlatma ve temizleme gerçekleştirdiğinden emin olun. REGISTER ve DEREGISTER geri çağrılarıyla iş parçacığı tabanlı olmayan kullanıma hazırlama ve temizleme yapabilirsiniz.

Belirtilen iş parçacığının kullanım ömrü ve kullanılabilirliği hakkında herhangi bir varsayımda bulunmayın. Örneğin, DEREGISTER ' e son çağrıdan sonra canlı kalan bir iş parçacığına güvenmeyin. Benzer şekilde, THREAD_AFFINITY değerini kullanmamayı seçtiğinizde, bağlantı her başlatıldığında iş parçacığının var olduğunu varsaymayın.

Uygulamanızın iş parçacığı özelliklerine ilişkin belirli gereksinimleri varsa, her zaman buna göre bir iş parçacığı yaratabilir ve MQCTL (WAIT) kullanabilir. Bu, zamanuyumsuz ileti teslimi için iş parçacığını IBM MQ 'e' bağışlama ' etkisine sahiptir.

İleti tüketici bağlantısı kullanımı

Bir tüketiciyi, tüketicinin yaşam çevrimi boyunca temel noktalarda geri çağırma üzere yapılandırabilirsiniz. Örneğin:

- Tüketici ilk kez kaydedildiğinde,
- bağlantı başlatıldığında,
- bağlantı durdurulduğunda ve
- Tüketici bir MQCLOSE tarafından belirttik ya da örtük olarak kayıttan kaldırıldığında.

<i>Çizelge 542. MQCTL fiil tanımlamaları</i>	
Komut	Anlamı
MQCTL (BAŞLAT)	MQOP_START İşlemi kullanılarak MQCTL çağrısı
MQCTL (DURDUR)	MQOP_STOP İşlemi kullanılarak MQCTL çağrısı
MQCTL (BEKLE)	MQOP_START_WAIT İşlemini Kullanarak MQCTL çağrısı

Bu, tüketicinin tüketiciyle ilişkili durumu korumasını sağlar. Bir uygulama tarafından geri çağırma istendiğinde, tüketici çağırma kuralları aşağıdaki gibidir:

KAYDOL

Her zaman geri çağırmanın ilk tipidir.

MQCB (REGISTER) çağrısıyla aynı iş parçacığında her zaman çağrılır.

START

MQCTL (START) komutu ile her zaman zamanuyumlu olarak çağrılır.

- Tüm START geri çağrılar, MQCTL (START) komutu döndürülmeden önce tamamlanır.

THREAD_AFFINITY istenirse, ileti teslimiyle aynı iş parçacığında.

Örneğin, MQCTL (START) sırasında önceki bir geri çağırma MQCTL (STOP) komutu verirse, başlatma ile çağrı garanti edilmez.

DUR

Bağlantı yeniden başlatılıncaya kadar, bu çağrıdan sonra başka ileti ya da olay teslim edilmez.

Uygulama daha önce START ya da ileti ya da olay için çağrıldıysa, STOP garanti edilir.

DEREGISTER

Her zaman geri çağırmanın son türüdür.

Uygulamanızın START ve STOP geri çağrılarında iş parçacığı tabanlı başlatma ve temizleme gerçekleştirdiğinden emin olun. REGISTER ve DEREGISTER geri çağrılarıyla iş parçacığı tabanlı olmayan kullanıma hazırlama ve temizleme yapabilirsiniz.

Belirtilen iş parçacığının kullanım ömrü ve kullanılabilirliği hakkında herhangi bir varsayımda bulunmayın. Örneğin, DEREGISTER ' e son çağrıdan sonra canlı kalan bir iş parçacığına güvenmeyin. Benzer şekilde, THREAD_AFFINITY değerini kullanmamayı seçtiğinizde, bağlantı her başlatıldığında iş parçacığının var olduğunu varsaymayın.

Uygulamanızın iş parçacığı özelliklerine ilişkin belirli gereksinimleri varsa, her zaman buna göre bir iş parçacığı yaratabilir ve MQCTL (WAIT) kullanabilir. Bu, zamanuyumsuz ileti teslimi için iş parçacığını IBM MQ 'e' bağışlama ' etkisine sahiptir.

C çağırısı

```
MQCB (Hconn, Operation, CallbackDesc, Hobj, MsgDesc,  
GetMsgOpts, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağırısı

```
CALL 'MQCB' USING HCONN, OPERATION, CBDESC, HOBJ, MSGDESC,  
GETMSGOPTS, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```


PL/I çağırısı

```
call MQCB(Hconn, Operation, CallbackDesc, Hobj, MsgDesc, GetMsgOpts,  
          CompCode, Reason)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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-Geri çağırma işlevi

MQCB_FUNCTION işlev çağırısı, olay işleme ve zamanuyumsuz ileti tüketimi için geri çağırma işlevidir.

MQCB_FUNCTION çağrı tanımlaması yalnızca, geri çağırma işlevine geçirilen değiştirgeleri tanımlamak için sağlanır. Kuyruk yöneticisi tarafından MQCB_FUNCTION adlı bir giriş noktası sağlanmadı.

Çağrılacak gerçek işlevin belirtimi, [MQCB](#) çağırısına bir giriştir ve [MQCBD](#) yapısından geçirilir.

Sözdizimi

MQCB_FUNCTION (*Hconn, MsgDesc, GetMsgOpts, Arabellek, Bağlam*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanırtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağırısı tarafından döndürüldü. z/OS ' da CICS uygulamaları için MQCONN çağırısı atlanabilir ve Hconn için belirtilen şu değer:

MQHC_DEF_CONN

Varsayılan bağlantı tanırtıcısı.

MsgDesc

Tip: MQMD-giriş

Bu yapı, alınan iletinin özniteliklerini açıklar.

Ayrıntılar için bkz. [“MQMD-İleti tanımlayıcı” sayfa 420.](#)

İletilen MQMD sürümü, tüketici işlevini tanımlayan MQCB çağırısında geçirilen sürümle aynı.

MQMD ' nin adresi, MQMD yerine İleti Tanırtıcısı döndürülmesini istemek için sürüm 4 MQGMO kullanıldıysa, boş karakter olarak geçirilir.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; bir olay işleyici işleviyle ilgili değildir.

GetMsgSeçmeler

Tip: MQGMO-giriş

İleti tüketicisinin işlemlerini denetlemek için kullanılan seçenekler. Bu değiştirge, döndürülen iletiye ilişkin ek bilgi de içerir.

Ayrıntılar için bkz. [MQGMO](#) .

İletilen MQGMO sürümü, desteklenen en son sürümdür.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; bir olay işleyici işleviyle ilgili değildir.

Arabellek

Tip: MQBYTExBufferUzunluk-giriş

Bu, ileti verilerini içeren alandır.

Bu çağrı için herhangi bir ileti yoksa ya da ileti ileti verisi içermezse, *Buffer* adresi boş değer olarak geçirilir.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; bir olay işleyici işleviyle ilgili değildir.

Bağlam

Tip: MQCBC-giriş/çıkış

Bu yapı, geri çağırma işlevlerine bağlam bilgisi sağlar. Ayrıntılar için bkz. "[MQCBC-Geri çağırma bağlamı](#)" sayfa 281.

Kullanım notları

1. Geri çağırma yordamlarınız iş parçacığını geciktirebilecek ya da engelleyebilecek hizmetler (örneğin, MQGET ve bekleme) kullanıyorsa, diğer geri çağırma işlemlerinin dağıtılması gecikebilir.
2. Bir geri çağırma yordamının her çağırılması için ayrı bir iş birimi otomatik olarak oluşturulmaz; bu nedenle, yordamlar bir kesinleştirme çağırısı verebilir ya da mantıksal bir iş kümesi işleninceye kadar kesinleştirmeyi erteleyebilir. İş kümesi kesinleştirildiğinde, son eşitleme noktasından bu yana çağrılan tüm geri çağırma işlevlerine ilişkin iletileri kesinleştirir.
3. CICS LINK ya da CICS START tarafından çağrılan programlar, kanal taşıyıcıları olarak bilinen adlandırılmış nesnelere aracılığıyla CICS hizmetlerini kullanarak parametreleri alır. Taşıyıcı adları, değiştirge adlarıyla aynıdır. Daha fazla bilgi için CICS belgelerinize bakın.
4. Geri çağırma yordamları bir MQDISC çağırısı verebilir, ancak kendi bağlantıları için veremez. Örneğin, bir geri çağırma yordamı bağlantı yarattıysa, bağlantının bağlantısını da kesebilir.
5. Bir geri çağırma yordamı, genel olarak, her seferinde aynı iş parçacığından çağrılmasına güvenmemelidir. Gerekiyorsa, bağlantı başlatıldığında MQCTLO_THREAD_AFFINITY komutunu kullanın.
6. Bir geri çağırma yordamı sıfır olmayan bir neden kodu aldığı anda, uygun işlemi gerçekleştirmelidir.
7. MQCB_FUNCTION, IMS bağdaştırıcısı içinde desteklenmiyor.

MQCLOSE-Nesneyi kapat

MQCLOSE çağırısı bir nesneye erişimi durdurarak MQOPEN ve MQSUB çağrılarının tersini oluşturur.

Sözdizimi

MQCLOSE (*Hconn*, *Hobj*, *Seçenekler*, *CompCode*, *Neden*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağırısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağrılarını atlayabilir ve *Hconn*'in aşağıdaki değeri belirtebilirsiniz:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

Hobj.

Tip: MQHOBJ-giriş/çıkış

Bu tanıtıcı, kapatılan nesneyi gösterir. Nesne herhangi bir tipte olabilir. Önceki bir MQOPEN çağrısı *Hobj* değerini döndürdü.

Çağrı başarıyla tamamlandığında, kuyruk yöneticisi bu parametreyi ortam için geçerli bir tanıtıcı olmayan bir değere ayarlar. Bu değer:

MQHO_UNUSABLE_HOBJ

Kullanılmayan nesne tanıtıcısı.

z/OS' da *Hobj* , tanımlanmamış bir değere ayarlanır.

Seçenekler

Tip: MQLONG-giriş

Bu parametre, nesnenin nasıl kapatılacağını denetler.

Yalnızca kalıcı dinamik kuyruklar ve abonelikler birden çok şekilde kapatılabilir; bunların alıkonması ya da silinmesi gerekir; bunlar, MQQDT_PERMANENT_DYNAMIC değerine sahip **DefinitionType** özniteliğine sahip kuyruklardır (bkz. “Kuyruklara ilişkin öznitelikler” sayfa 826 içinde açıklanan **DefinitionType** özniteliğine bakın). Kapatma seçenekleri bu konuda özetlenir.

Sürekli abonelikler alıkonabilir ya da kaldırılabilir; bunlar MQSO_DURABLE seçeneğiyle MQSUB çağrısı kullanılarak yaratılır.

Tanıtıcıyı yönetilen bir hedefe kapatırken (MQSO_MANAGED seçeneğini kullanan bir MQSUB çağrısında döndürülen **Hobj** değiştirgesi) kuyruk yöneticisi, ilişkili abonelik de kaldırıldığında alınmamış yayınları temizler. Abonelik, bir MQSUB çağrısında döndürülen **Hsub** değiştirgesindeki MQCO_REMOVE_SUB seçeneği kullanılarak kaldırılır. MQCO_REMOVE_SUB ögesinin kalıcı olmayan bir abonelik için MQCLOSE üzerinde varsayılan davranış olduğunu unutmayın.

Bir tutamacı yönetilmeyen bir hedefe kapatırken, yayınların gönderildiği kuyruğu temizlemek sizin sorumluluğunuzdadır. Önce MQCO_REMOVE_SUB kullanarak aboneliği kapatın ve sonra ileti kalmayınca kadar kuyruktaki iletileri işleyin.

Aşağıdaki seçeneklerden yalnızca birini belirtmeniz gerekir:

Dinamik kuyruk seçenekleri: Bu seçenekler, kalıcı dinamik kuyrukların nasıl kapatılacağını denetler.

MQCO_DELETE

Aşağıdakilerden biri doğruysa, kuyruk silinir:

- Bu, önceki bir MQOPEN çağrısı tarafından yaratılan kalıcı bir dinamik kuyruktur ve kuyruktaki ileti yoktur ve kuyruk için bekleyen kesinleştirilmemiş alma ya da yerleştirme istekleri yoktur (yürürlükteki görev ya da başka bir görev için).
- *Hobj* ögesini döndüren, MQOPEN çağrısıyla yaratılan geçici dinamik kuyruktur. Bu durumda, kuyruktaki tüm iletiler temizlenir.

Hobj ' in bir MQSUB çağrısında döndürüldüğü vaka da içinde olmak üzere diğer tüm durumlarda, çağrı MQRC_OPTION_NOT_VALID_FOR_TYPE neden koduyla başarısız olur ve nesne silinmez.

z/OS' ta, kuyruk mantıksal olarak silinmiş dinamik bir kuyruksa ve bu, kuyruk için son tanıtıcıysa, fiziksel olarak silinir. Daha fazla ayrıntı için bkz. “Kullanım notları” sayfa 647 .

MQCO_DELETE_PURGE

Aşağıdakilerden biri doğruysa, kuyruk silinir ve üzerindeki iletiler temizlenir:

- Önceki bir MQOPEN çağrısı tarafından yaratılan kalıcı dinamik bir kuyruktur ve kuyruk için (yürürlükteki görev ya da başka bir görev için) kesinleştirilmemiş alma ya da koyma isteği yoktur.
- *Hobj* ögesini döndüren, MQOPEN çağrısıyla yaratılan geçici dinamik kuyruktur.

Hobj ' in bir MQSUB çağrısında döndürüldüğü vaka da içinde olmak üzere diğer tüm durumlarda, çağrı MQRC_OPTION_NOT_VALID_FOR_TYPE neden koduyla başarısız olur ve nesne silinmez.

<i>Çizelge 543. Farklı nesne tipleri için seçenekleri kapat</i>			
Nesne ya da kuyruk tipi	MQCO_NONE	MQCO_DELETE	MQCO_DELETE_PURGE
Kuyruk dışındaki bir nesne	Alıkondur	Geçerli değil	Geçerli değil
Önceden tanımlı kuyruk	Alıkondur	Geçerli değil	Geçerli değil
Kalıcı dinamik kuyruk	Alıkondur	Boşsa ve bekleyen güncelleme yoksa silindi	İletiler silindi; bekleyen güncelleme yoksa kuyruk silindi
Geçici dinamik kuyruk (kuyruk yaratan tarafından çağrı yayınlandı)	Silindi	Silindi	Silindi
Geçici dinamik kuyruk (kuyruk yaratan tarafından çağrı yayınlanmadı)	Alıkondur	Geçerli değil	Geçerli değil
Dağıtım listesi	Alıkondur	Geçerli değil	Geçerli değil
Yönetilen abonelik hedefi	Alıkondur	Geçerli değil	Geçerli değil
Dağıtım listesi (abonelik kaldırıldı)	İletiler silindi; kuyruk silindi	Geçerli değil	Geçerli değil

Abonelik kapatma seçenekleri: Bu seçenekler, tutamaç kapatıldığında sürekli aboneliklerin kaldırılıp kaldırılmayacağını ve uygulama tarafından okunmayı bekleyen yayınların temizlenip temizlenmeyeceğini denetler. Bu seçenekler yalnızca, MQSUB çağrısının **Hsub** değiştirilmesinde döndürülen bir nesne tanıtıcısıyla kullanılmak üzere geçerlidir.

MQCO_KEEP_SUB

Abonelik tanıtıcısı kapatıldı, ancak yapılan abonelik korundu. Yayınlar, abonelikte belirtilen hedefe gönderilmeye devam eder. Bu seçenek, abonelik MQSO_DURABLE seçeneğiyle yapıldıysa geçerlidir.

Abonelik dayanıklıysa, MQCO_KEEP_SUB varsayılan değerdir.

MQCO_REMOVE_SUB

Abonelik kaldırılır ve abonelik tanıtıcısı kapatılır.

MQSUB çağrısının **Hobj** değiştirilmesi **Hsub** değiştirilmesinin kapatılmasıyla geçersiz kılınmaz ve geri kalan yayınları almak üzere MQGET ya da MQCB için kullanılmaya devam edebilir. MQSUB çağrısının **Hobj** değiştirilmesi de kapatıldığında, bu bir yönetilen hedefse, alınmamış yayınlar kaldırılır.

Abonelik dayanıklı değilse, MQCO_REMOVE_SUB varsayılan değerdir.

MQCO_REMOVE_SUB işleminin başarıyla tamamlanması, işlemin tamamlandığı anlamına gelmez. Bu aramanın tamamlanıp tamamlanmadığını denetlemek için Dağıtılmış ağlara ilişkin zamanuysuz komutların tamamlanıp tamamlanmadığını denetleme başlıklı konudaki DELETE SUB adımına bakın.

Bu abonelik kapatma seçenekleri aşağıdaki tablolarda özetlenmiştir.

<i>Çizelge 544. Sürekli abonelik tanıtıcısını kapatma, ancak aboneliği koruma seçenekleri</i>	
Görev	Abonelik kapatma seçeneği
Yayınları MQOPENed tutamacında tut	MQCO_KEEP_SUB
MQOPENed tanıtıcısındaki yayınları kaldır	İşleme izin verilmiyor
Yayınları MQSO_MANAGED tutamacında tut	MQCO_KEEP_SUB
MQSO_MANAGED tanıtıcısındaki yayınları kaldır	İşleme izin verilmiyor

Sürekli bir abonelik tanıtıcısını kapatıp aboneliğini kaldırarak ya da kalıcı olmayan bir abonelik tanıtıcısını kapatarak aboneliğinizi kaldırmak için aşağıdaki abonelik kapatma seçeneklerini kullanın:

Çizelge 545. Aboneliği kaldırma seçenekleri	
Görev	Abonelik kapatma seçeneği
Yayınları MQOPENed tutamacında tut	MQCO_REMOVE_SUB
MQOPENed tanıtıcısındaki yayınları kaldır	İşleme izin verilmiyor
Yayınları MQSO_MANAGED tutamacında tut	MQCO_REMOVE_SUB

Önden okuma seçenekleri: Aşağıdaki seçenekler, bir uygulama tarafından istenmeden önce istemciye gönderilen ve henüz uygulama tarafından tüketilmeyen kalıcı olmayan iletilere ne olacağını denetler. Bu iletiler, uygulama tarafından istenmeyi bekleyen istemci önden okuma arabelleğinde saklanır ve MQCLOSE tamamlanmadan önce atılabilir ya da kuyruktan tüketilebilir.

MQCO_IMMEDIATE

Nesne hemen kapatılır ve istenmeden önce istemciye gönderilen iletiler atılır ve herhangi bir uygulama tarafından tüketilemez. Bu varsayılan değerdir.

MQCO_QUIESCE

Nesneyi kapatma isteğinde bulunursanız, ancak bir uygulama bunları istemeden önce istemciye gönderilen iletiler istemci önden okuma arabelleğinde bulunmaya devam ediyorsa, MQCLOSE çağrısı MQRC_READ_AHEAD_MSGS uyarısıyla geri döner ve nesne tanıtıcısı geçerli kalır.

Daha sonra uygulama, ileti almak için nesne tanıtıcısını kullanmaya devam edebilir ve daha sonra, nesneyi yeniden kapatabilir. İstemciye, istekte bulunan bir uygulamadan önce başka ileti gönderilmez, önden okuma kapatılmış olur.

Son MQGET çağrısı ile MQCO_IMMEDIATE kullanıldıysa atılacak aşağıdaki MQCLOSE arasında bir ileti olabileceğinden, uygulamaların istemcinin önden okuma arabelleğinde başka ileti olmadığı bir noktaya ulaşmaya çalışmak yerine MQCO_QUIESCE kullanmaları önerilir.

Zamanuyumsuz bir geri çağırma işlevinden MQCO_QUIESCE içeren bir MQCLOSE yayınlandıysa, önden okuma iletileriyle aynı davranış uygulanır. MQRC_READ_AHEAD_MSGS uyarısı döndürülürse, geri çağırma işlevi en az bir kez daha çağrılır. Önden okunan son ileti geri çağırma işlevine geçirildiğinde, MQCBC ConsumerFlags alanı MQCBCF_READA_BUFFER_EMPTY olarak ayarlanır.

Varsayılan seçenek: Önceden açıklanan seçeneklerin hiçbirine gerek duymazsanız, aşağıdaki seçeneği kullanabilirsiniz:

MQCO_NONE

İsteğe bağlı kapatma işlemi gerekmiyor.

Bu, aşağıdakiler için belirtilmelidir:

- Kuyruklar dışındaki nesnelere
- Önceden tanımlı kuyruklar
- Geçici devingen kuyruklar (ancak, *Hobj* 'in kuyruğu yaratan MQOPEN çağrısıyla döndürülen tanıtıcı olmadığı durumlarda).
- Dağıtım listeleri

Yukarıdaki tüm durumlarda, nesne korunur ve silinmez.

Geçici bir dinamik kuyruk için bu seçenek belirtilirse:

- Kuyruk, *Hobj* ; döndüren MQOPEN çağrısıyla yaratıldıysa silinir. kuyruktaki iletiler temizlenir.
- Diğer tüm durumlarda kuyruk (ve üzerindeki iletiler) alıkonur.

Kalıcı dinamik kuyruk için bu seçenek belirtilirse, kuyruk alıkonur ve silinmez.

z/OS' ta, kuyruk mantıksal olarak silinmiş dinamik bir kuyruksa ve bu, kuyruk için son tanıtıcıysa, fiziksel olarak silinir. Daha fazla ayrıntı için bkz. “Kullanım notları” sayfa 647 .

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Listelenen neden kodları, kuyruk yöneticisinin **Reason** parametresi için döndürebileceği kodlardır.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_INCOMPLETE_GROUP

(2241, X'8C1') İleti grubu tamamlanmadı.

MQRC_INCOMPLETE_MSG

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

MQRC_READ_AHEAD_MSGS

(nnnn, X'xxx ') İstemci, henüz uygulama tarafından kullanılmayan önden okuma iletileri aldı.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenmiyor.

MQRC_API_EXIT_HATA

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_LOAD_ERROR (MQRC_API_EXIT_HATASI)

(2183, X'887 ') API çıkışı yüklenmiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CF_NOT_YOK

(2345, X' 929 ') Bağlama olanağı kullanılmıyor.

MQRC_CF_STRUC_FAILED

(2373, X' 945 ') Bağlama tesisi yapısı başarısız oldu.

MQRC_CF_STRUC_IN_USE

(2346, X'92A') Kupon kullanım olanağı yapısı.

MQRC_CICS_WAIT_FAILED

(2140, X'85C') Bekleme isteği CICStarafından reddedildi.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_CONNECTION_NOT_YETKILI

(2217, X'8A9') Bağlantı için yetkili değil.

MQRC_CONNECTION_DURDURULUYOR

(2203, X'89B') Bağlantı kapatılıyor.

MQRC_DB2_NOT_AVAILABLE

(2342, X' 926 ') Db2 altsistemi kullanılamıyor.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_HOBJ_HATA

(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_OBJECT_HASARLI

(2101, X'835 ') Nesne zarar gördü.

MQRC_OPTION_NOT_VALID_FOR_TYPE

(2045, X'7FD') MQOPEN ya da MQCLOSE çağrısında: seçeneği nesne tipi için geçerli değil.

MQRC_OPTIONS_HATA

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_PAGESET_HATA

(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRC_Q_MGR_AD_HATASI

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRC_Q_MGR_YOK

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_Q_NOT_EMPTY

(2055, X'807 ') Kuyruk bir ya da daha fazla ileti ya da kesinleştirilmemiş koyma ya da alma isteği içeriyor.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_SECURITY_HATA

(2063, X'80F') Güvenlik hatası oluştu.

MQRC_STORAGE_NOT_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_SUPPRESSED_BY_EXIT

(2109, X'83D') Çağrı çıkış programı tarafından engellendi.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. Bir uygulama MQDISC çağrısıyla ilgili bir çağrıda bulunduğu ya da olağan dışı ya da olağandışı bir şekilde sona erdiğinde, uygulama tarafından açılan ve hala açık olan nesnelere MQCO_NONE seçeneğiyle otomatik olarak kapatılır.
2. Kapatılan nesne bir *kuyruk* ise aşağıdaki noktalar geçerlidir:
 - Kuyruktaki işlemler bir iş biriminin parçası olarak gerçekleştirilirse, kuyruk, eşitleme noktasının sonucunu etkilemeden, eşitleme noktasından önce ya da sonra kapatılabilir. Kuyruk tetiklenirse, kuyruğu kapatmadan önce geriye işleme gerçekleştirilmesi bir tetikleyici iletisinin yayınlanmasına

neden olabilir. Tetikleyici iletilerine ilişkin ek bilgi için [Tetikleyici iletilerinin özellikleribaşlıklı konuya](#) bakın.

- Kuyruk MQOO_BROWSE seçeneğiyle açıldıysa, göz atma imleci yok edilir. Kuyruk daha sonra MQOO_BROWSE seçeneğiyle yeniden açılırsa, yeni bir göz atma imleci yaratılır (bkz. [MQOO_BROWSE](#)).
- MQCLOSE çağrısı sırasında bu tanıtıcı için bir ileti kilitliyse, kilit serbest bırakılır (bkz. [MQGMO_LOCK](#)).
- z/OS sistemlerinde, kuyruk tanıtıcısıyla ilgili olarak MQGMO_SET_SIGNAL seçeneği bekleyen bir MQGET isteği varsa, istek iptal edilir ([MQGMO_SET_SIGNAL](#) konusuna bakın). Aynı kuyruğa ilişkin, ancak farklı tanıtıcılara (*Hobj*) karşı yerleştirilen sinyal istekleri etkilenmez (dinamik bir kuyruk silinmedikçe, bu durumda bunlar da iptal edilir).

3. Kapatılan nesne bir *dinamik kuyruk* (kalıcı ya da geçici) ise aşağıdaki noktalar geçerlidir:

- Devingen bir kuyruk için, ilgili MQOPEN çağrısında belirtilen seçeneklerden bağımsız olarak MQCO_DELETE ve MQCO_DELETE_PURGE seçeneklerini belirtebilirsiniz.
- Dinamik bir kuyruk silindiğinde, kuyruk için bekleyen MQGMO_WAIT seçeneğiyle tüm MQGET çağrıları iptal edilir ve neden kodu MQRC_Q_DELETED döndürülür. Bkz. [MQGMO_WAIT](#).

Uygulamalar silinmiş bir kuyruğa erişemese de, kuyruk sistemden kaldırılmaz ve kuyruğa başvuran tüm tanıtıcıları kapatıncaya ve kuyruğu etkileyen tüm iş birimleri kesinleştirilinceye ya da yedekleninceye kadar, ilişkili kaynaklar serbest bırakılmaz.

z/OS sistemlerinde, mantıksal olarak silinmiş ancak henüz sistemden kaldırılmamış bir kuyruk, silinen kuyrukla aynı ada sahip yeni bir kuyruk yaratılmasını önler; MQOPEN çağrısı bu durumda MQRC_NAME_IN_USE neden koduyla başarısız olur. Ayrıca, bu tür bir kuyruk, uygulamalar tarafından erişilemese de MQSC komutları kullanılarak da görüntülenebilir.

- Kalıcı bir dinamik kuyruk silindiğinde, MQCLOSE çağrısında belirtilen *Hobj* tanıtıcısı, kuyruğu yaratan MQOPEN çağrısının döndürdüğü tanıtıcı değilse, MQOPEN çağrısının geçerliliğini denetlemek için kullanılan kullanıcı kimliğinin kuyruğu silme yetkisi olduğu denetlenir. MQOPEN çağrısında MQOO_ALTERNATE_USER_AUTHORITY seçeneği belirtildiyse, denetlenen kullanıcı kimliği *AlternateUserId* olur.

Bu denetim aşağıdaki durumda gerçekleştirilmez:

- Belirtilen tanıtıcı, kuyruğu yaratan MQOPEN çağrısıyla döndürülen tanıtıcı.
- Silinmekte olan kuyruk geçici bir dinamik kuyruk.
- Geçici bir dinamik kuyruk kapatıldığında, MQCLOSE çağrısında belirtilen *Hobj* tanıtıcısı, kuyruğu yaratan MQOPEN çağrısıyla döndürülen tanıtıcıysa, kuyruk silinir. Bu durum, MQCLOSE çağrısında belirtilen kapatma seçeneklerinden bağımsız olarak oluşur. Kuyrukta ileti varsa, bunlar atılır; rapor iletisi oluşturulmaz.

Kuyruğu etkileyen kesinleştirilmemiş iş birimleri varsa, kuyruk ve iletileri silinmeye devam eder, ancak iş birimleri başarısız olmaz. Ancak, daha önce açıklandığı gibi, iş birimleriyle ilişkili kaynaklar, her bir iş birimi kesinleştirilinceye ya da geri çekilinceye kadar serbest bırakılmaz.

4. Kapatılan nesne bir *dağıtım listesi* ise aşağıdaki noktalar geçerlidir:

- Dağıtım listesi için geçerli olan tek kapatma seçeneği MQCO_NONE 'dir; başka seçenekler belirtilirse, çağrı MQRC_OPTIONS_ERROR ya da MQRC_OPTION_NOT_VALID_FOR_TYPE neden koduyla başarısız olur.
- Bir dağıtım listesi kapatıldığında, listedeki kuyruklar için tek tek tamamlanma kodları ve neden kodları döndürülmez; tanılama amacıyla yalnızca çağrıya ilişkin **CompCode** ve **Reason** parametreleri kullanılabilir.

Kuyruklardan biri kapatılırken bir hata oluşursa, kuyruk yöneticisi işlemeye devam eder ve dağıtım listesinde kalan kuyrukları kapatmayı dener. Çağrıya ilişkin **CompCode** ve **Reason** parametreleri, hatayı açıklayan bilgileri döndürecek şekilde ayarlanır. Kuyrukların çoğu başarıyla kapatılmış olsa da, tamamlama kodunun MQCC_FAILED olması mümkündür. Hatayla karşılaşan kuyruk saptanmadı.

Birden çok kuyrukta bir hata varsa, **CompCode** ve **Reason** parametrelerinde hangi hatanın bildirildiği tanımlanmaz.

C çağırısı

```
MQCLOSE (Hconn, &Hobj, Options, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağırısı

```
CALL 'MQCLOSE' USING HCONN, HOBJ, OPTIONS, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağırısı

```
call MQCLOSE (Hconn, Hobj, Options, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQCLOSE,(HCONN,HOBJ,OPTIONS,COMPCODE,REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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
```

Visual Basic çağrısı

```
MQCLOSE Hconn, Hobj, Options, CompCode, Reason
```


Değiştirgeleri aşağıdaki gibi bildirin:

```
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-Değişiklikleri kesinleştir

MQCMIT çağrısı, kuyruk yöneticisine uygulamanın bir eşitleme noktasına ulaştığını ve son eşitleme noktasından bu yana oluşan tüm iletilerin kalıcı hale getirileceğini belirtir.

Bir iş biriminin parçası olarak konan iletiler diğer uygulamalar tarafından kullanılabilir; bir iş biriminin parçası olarak alınan iletiler silinir.

-  z/OS üzerinde, çağrı yalnızca toplu iş programları (IMS toplu DL/I programları da içinde olmak üzere) tarafından kullanılır.

Sözdizimi

MQCMIT (*Hconn*, *CompCode*, *Reason*(Neden))

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıttıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Listelenen neden kodları, kuyruk yöneticisinin **Reason** parametresi için döndürebileceği kodlardır.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_BACKED_OUT

(2003, X'7D3') Çalışma birimi geri çekildi.

MQRC_OUTCOME_PENDING

(2124, X'84C') Kesinleştirme işleminin sonucu beklemede.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CALL_INTERRUPTED

(2549, X'9F5') MOPUT ya da MQCMIT kesildi ve yeniden bağlanma işlemi kesin bir sonucu yeniden oluşturamıyor.

MQRC_CF_STRUC_IN_USE

(2346, X'92A') Coupling-facility yapısı kullanımda.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_ENVIRONMENT_ERROR

(2012, X'7DC') Çağrısı ortamda geçerli değil.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_OBJECT_HASARLI

(2101, X'835 ') Nesne zarar gördü.

MQRC_OUTCOME_MIXED

(2123, X'84B') Kesinleştirme ya da geri yükleme işleminin sonucu karışık.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RECONNECT_FAILED

(2548, X'9F4') Yeniden bağlandıktan sonra, yeniden bağlanabilir bir bağlantıya ilişkin tutamaçlar yeniden alınırken bir hata oluştu.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_STORAGE_MEDIUM_FULL

(2192, X'890 ') Dış depolama ortamı dolu.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. Bu çağrıyı yalnızca kuyruk yöneticisinin kendisi iş birimini koordine ettiğinde kullanın. Bu şöyle olabilir:

- Değişikliklerin yalnızca IBM MQ kaynaklarını etkilediği yerel bir iş birimi.
- Değişikliklerin diğer kaynak yöneticilerine ait kaynakları etkileyebileceği ve IBM MQ kaynaklarını etkileyebileceği genel bir iş birimi.

Yerel ve genel iş birimleriyle ilgili daha fazla ayrıntı için bkz. [“MQBEGIN-İş birimini başlat” sayfa 625](#).

2. Kuyruk yöneticisinin iş birimini koordine etmediği ortamlarda, MQCMIT yerine uygun kesinleştirme çağrısı kullanılmalıdır. Ortam, uygulamanın olağan şekilde sonlandırılmasından kaynaklanan örtük bir kesinleştirmeyi de destekleyebilir.
 - z/OS' da aşağıdaki çağrıları kullanın:
 - İş birimi yalnızca IBM MQ kaynaklarını etkiliyorsa, toplu iş programları (IMS toplu DL/I programları da içinde olmak üzere) MQCMIT çağrılarını kullanabilir. Ancak, iş birimi diğer kaynak yöneticilerine ait IBM MQ kaynaklarını ve kaynaklarını etkiliyorsa (örneğin, Db2), z/OS Recoverable Resource Service (RRS) tarafından sağlanan SRRRCMIT çağrısına bakın. SRRRCMIT çağrısı, RRS eşgüdümü için etkinleştirilen kaynak yöneticilerine ait kaynaklarda yapılan değişiklikleri kesinleştirir.
 - CICS uygulamaları, iş birimini belirttik olarak kesinleştirmek için EXEC CICS SYNCPOINT komutunu kullanmalıdır. Diğer bir seçenek olarak, hareketin sona erdirilmesi, iş biriminin örtük olarak kesinleştirilmesiyle sonuçlanır. MQCMIT çağrısı CICS uygulamaları için kullanılamaz.
 - IMS uygulamaları (toplu DL/I programlarından başka) iş birimini kesinleştirmek için GU ve CHKP gibi IMS çağrıları kullanılmalıdır. MQCMIT çağrısı IMS uygulamaları (toplu DL/I programları dışında) için kullanılamaz.
 - IBM işletim sistemi üzerinde, kuyruk yöneticisi tarafından eşgüdümlü yerel iş birimleri için bu çağrıyı kullanın. Bu, iş düzeyinde bir kesinleştirme tanımının var olmaması gerektiği, yani iş için **CMTSCOPE(*JOB)** parametresini içeren STRCMTCTL komutunun yayınlanmamış olması gerektiği anlamına gelir.
3. Bir uygulama, bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması uygulamanın olağan şekilde mi, yoksa olağandışı olarak mı bittiğine bağlıdır. Daha fazla ayrıntı için bkz. [MQDISC kullanım notları](#) .
4. Bir uygulama, mantıksal ileti gruplarına ya da bölümlerine ileti girdiğinde ya da aldığı anda, kuyruk yöneticisi, son başarılı MQPUT ve MQGET çağrılarında ilişkin ileti grubu ve mantıksal iletiyle ilgili bilgileri saklar. Bu bilgiler kuyruk tanıtıcısıyla ilişkilidir ve aşağıdaki gibi şeyleri içerir:
 - MQMD ' deki *GroupId*, *MsgSeqNumber*, *Offset* ve *MsgFlags* alanlarının değerleri.
 - İletinin bir iş biriminin parçası olup olmadığı.
 - MQPUT çağrısı için: İletinin kalıcı mı, kalıcı mı olduğu.

Bir iş birimi kesinleştirildiğinde, kuyruk yöneticisi grup ve bölüm bilgilerini saklar ve uygulama iletileri yürürlükteki ileti grubuna ya da mantıksal iletiye yerleştirmeye ya da almaya devam edebilir.

Bir iş birimi kesinleştirildiğinde grup ve bölüm bilgilerinin tutulması, uygulamanın çeşitli iş birimlerinde birçok bölümden oluşan büyük bir ileti grubu ya da büyük bir mantıksal ileti dağıtmasına olanak sağlar. Yerel kuyruk yöneticisinin yalnızca sınırlı kuyruk saklama alanı varsa, birkaç iş birimi kullanılması avantajlıdır. Ancak, bir sistem arızası oluşursa, uygulamanın iletileri doğru noktada yeniden yerleştirmek ya da almak için yeterli bilgiyi koruması gerekir. Bir sistem hatasından sonra doğru noktada nasıl yeniden başlatılacağına ilişkin ayrıntılar için bkz. [MQPMO_LOGICAL_ORDER](#) ve [MQGMO_LOGICAL_ORDER](#).

Geri kalan kullanım notları yalnızca kuyruk yöneticisi iş birimlerini koordine ettiğinde geçerlidir:
5. Bir iş birimi, bağlantı tanıtıcısıyla aynı kapsama sahiptir; belirli bir iş birimini etkileyen tüm IBM MQ çağrılarının aynı bağlantı tanıtıcısı kullanılarak gerçekleştirilmesi gerekir. Farklı bir bağlantı tanıtıcısı kullanılarak verilen çağrılar (örneğin, başka bir uygulama tarafından verilen çağrılar) farklı bir iş birimini etkiler. Bağlantı tanıtıcılarının kapsamıyla ilgili bilgi için MQCONN içinde açıklanan **Hconn** değiştirilmesine bakın.
6. Yalnızca yürürlükteki iş biriminin bir parçası olarak konan ya da alınan iletiler bu çağrıdan etkilenir.
7. Bir iş birimi içinde MQGET, MQPUT ya da MQPUT1 çağrıları veren, ancak hiçbir zaman kesinleştirme ya da gerileme çağrısı yapmayan uzun süreli bir uygulama, kuyrukları diğer uygulamalar tarafından kullanılmayan iletilerle doldurabilir. Bunu önlemek için, denetimcinin **MaxUncommittedMsgs** kuyruk yöneticisi özniteliğini, kuyrukları dolduran, ancak beklenen ileti sistemi uygulamalarının düzgün çalışmasını sağlayacak kadar yüksek olan kaçak uygulamaları önleyecek kadar düşük bir değere ayarlaması gerekir.

8. **ALW** AIX, Linux, and Windows sistemlerinde **Reason** deęiřtirgesi MQRC_CONNECTION_BROKEN (MQCC_FAILED *CompCode* ile) ya da MQRC_UNEXPECTED_ERROR ise, iř birimi bařarıyla kesinleřtirilmiř olabilir.

C aęrısı

```
MQCMIT (Hconn, &CompCode, &Reason);
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
MQHCONN Hconn; /* Connection handle */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Reason code qualifying CompCode */
```

COBOL aęrısı

```
CALL 'MQCMIT' USING HCONN, COMPCODE, REASON.
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
** 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.
```

PL/I aęrısı

```
call MQCMIT (Hconn, CompCode, Reason);
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
dcl Hconn fixed bin(31); /* Connection handle */
dcl CompCode fixed bin(31); /* Completion code */
dcl Reason fixed bin(31); /* Reason code qualifying CompCode */
```

High Level Assembler (Yüksek Düzeyli çevirici) aęrısı

```
CALL MQCMIT,(HCONN,COMPCODE,REASON)
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
HCONN DS F Connection handle
COMPCODE DS F Completion code
REASON DS F Reason code qualifying COMPCODE
```

Visual Basic aęrısı

```
MQCMIT Hconn, CompCode, Reason
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
Dim Hconn As Long 'Connection handle'  
Dim CompCode As Long 'Completion code'  
Dim Reason As Long 'Reason code qualifying CompCode'
```

MQCONN-Bağlantı kuyruğu yöneticisi

MQCONN çağrısı, bir uygulama programını bir kuyruk yöneticisine bağlar.

Uygulamanın sonraki ileti kuyruklama çağrılarında kullandığı bir kuyruk yöneticisi bağlantı tanıtıcısı sağlar.

- z/OS' da CICS uygulamalarının bu çağrıyı yapması gerekmez. Bu uygulamalar, CICS sisteminin bağlı olduğu kuyruk yöneticisine otomatik olarak bağlanır. Ancak, MQCONN ve MQDISC çağrıları CICS uygulamalarından kabul edilir.
- IBM üzerinde, uygulamalar kuyruk yöneticisine bağlanmak için MQCONN ya da MQCONNX çağrıyı ve kuyruk yöneticisiyle bağlantıyı kesmek için MQDISC çağrıyı kullanmalıdır.

Yalnızca sunucu kuruluşunda istemci bağlantısı kurulamaz ve yalnızca istemci kuruluşunda yerel bağlantı kurulamaz.

Sözdizimi

MQCONN (*QMgrName*, *Hconn*, *CompCode*, *Reason*)

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Bu, uygulamanın bağlanmak istediği kuyruk yöneticisinin adıdır. Ad aşağıdaki karakterleri içerebilir:

- Büyük harf alfabetik karakterler (A-Z)
- Küçük harf alfabetik karakterler (a-z)
- Sayısal basamaklar (0-9)
- Nokta (.), eğik çizgi (/), altçizgi (_), yüzde (%)

Ad başta ya da gömülü boşluklar içermemelidir, ancak sonda boşluklar içerebilir. Addaki önemli verilerin sonunu göstermek için boş karakter kullanılabilir; boş değer ve ardından gelen karakterler boşluk olarak değerlendirilir. Belirtilen ortamlarda aşağıdaki kısıtlamalar geçerlidir:

- EBCDIC Katakana kullanan sistemlerde küçük harfli karakterler kullanılamaz.
- z/OS üzerinde, alt çizgiyle başlayan ya da biten adlar, işlemler ve denetim panoları tarafından işlenemez. Bu nedenle, bu tür isimlerden kaçınınız.
- IBM üzerinde, komutlarda belirtildiğinde küçük harf, eğik çizgi ya da yüzde işareti içeren adları tırnak içine alın. **QMgrName** değiştirilmesinde bu tırnak işaretlerini belirtmeyin.

Ad tamamen boşluklardan oluşuyorsa, *varsayılan* kuyruk yöneticisinin adı kullanılır. Ancak, IBM MQ MQI client uygulamalarındaki bölümde açıklanan boş kuyruk yöneticisi adlarının kullanımına dikkat edin.

QMgrName için belirtilen ad bir *bağlanabilir* kuyruk yöneticisinin adı ya da kuyruk yöneticisi grupları kullanılıyorsa, kuyruk yöneticisi grubunun adı olmalıdır.

z/OS üzerinde, bağlanabildiği kuyruk yöneticileri ortam tarafından belirlenir:

- CICS için yalnızca CICS sisteminin bağlı olduğu kuyruk yöneticisini kullanabilirsiniz. **QMgrName** parametresi yine de belirtilmelidir, ancak değeri yoksayılmıştır; boş karakterler uygun bir seçenektir.
- IMS için, IMS içinde SSM çizelgesinde listelenen yalnızca altsistem tanımlama çizelgesinde (CSQQDEFV), ve kuyruk yöneticileri bağlanabilir (kullanım notuna bakın [6](#)).
- z/OS toplu iş ve TSO için, yalnızca uygulamayla aynı sistemde bulunan kuyruk yöneticileri bağlanabilir (kullanım notuna bakın [6](#)).

Kuyruk paylaşım grupları: Birden çok kuyruk yöneticisinin bulunduğu ve bir kuyruk paylaşım grubu oluşturmak üzere yapılandırıldığı sistemlerde, kuyruk yöneticisi adı yerine *QMgrName* için kuyruk paylaşım grubunun adı belirtilebilir. Bu, uygulamanın, kuyruk paylaşım grubunda bulunan ve uygulamayla aynı z/OS görüntüsünde bulunan *herhangi bir* kuyruk yöneticisine bağlanmasını sağlar. Sistem, boş bir *QMgrName* kullanılması varsayılan kuyruk yöneticisi yerine kuyruk paylaşım grubuna bağlanacak şekilde de yapılandırılabilir.

QMgrName , kuyruk paylaşım grubunun adını belirtirse, ancak sistemde bu adı taşıyan bir kuyruk yöneticisi de varsa, öncekiyle bağlantı kurulur. Yalnızca bu bağlantı başarısız olursa, denenen kuyruk paylaşım grubundaki kuyruk yöneticilerinden biriyle bağlantı kurulur.

Bağlantı başarılı olursa, MQCONN ya da MQCONNX çağrısıyla döndürülen tanıtıcı değeri kullanarak, bağlantının kurulduğu kuyruk yöneticisine ait olan *tüm* kaynaklara (paylaşılan ve paylaşılmayan) erişebilirsiniz. Bu kaynaklara erişim, tipik yetki denetimlerine tabidir.

Uygulama eşzamanlı bağlantı kurmak için iki MQCONN ya da MQCONNX çağrısı yaparsa ve/ya da her iki çağrı kuyruk paylaşım grubunun adını belirtirse, ikinci çağrı ilk çağrıyla aynı kuyruk yöneticisine bağlandığında MQCC_WARNING ve MQRC_ALREADY_CONNECTED neden kodunu döndürür.

Kuyruk paylaşım grupları yalnızca z/OS üzerinde desteklenir. Bir kuyruk paylaşım grubuna bağlantı yalnızca toplu iş, RRS toplu işi, CICS ve TSO ortamlarında desteklenir. CICS için yalnızca CICS sisteminin bağlı olduğu kuyruk paylaşım grubunu kullanabilirsiniz. **QMgrName** değiştirgesini yine de belirtmeniz gerekir, ancak değeri yoksayılr; boş karakterler uygun bir seçenektir.



Uyarı: IMS , bir kuyruk paylaşım grubuna bağlanamıyor.

IBM MQ MQI client applications: IBM MQ MQI client uygulamaları için, her istemci-bağlantı kanalı tanımlaması için belirtilen kuyruk yöneticisi adıyla, başarılı oluncaya kadar bağlantı girişiminde bulunulur. Ancak, kuyruk yöneticisi belirtilen adla aynı ada sahip olmalıdır. Tümü boş bir ad belirtilirse, kuyruk yöneticisi adı boş olan her istemci bağlantı kanalı, biri başarılı oluncaya kadar denenir; bu durumda, kuyruk yöneticisinin gerçek adıyla ilgili bir denetim olmaz.

IBM MQ istemci uygulamaları z/OS ürününde desteklenmez, ancak z/OS , IBM MQ istemci uygulamalarının bağlanabileceği bir IBM MQ sunucusu olarak işlev görür.

IBM MQ MQI client Kuyruk yöneticisi grupları: Belirtilen ad bir yıldız işaretiyle (*) başlıyorsa, bağlantının kurulduğu kuyruk yöneticisinin adı uygulama tarafından belirtilenden farklı olabilir. Belirtilen ad (yıldız işareti olmadan), bağlantı için uygun kuyruk yöneticilerinden oluşan bir *grubu* tanımlar. Uygulama, bağlantı kurulabilecek bir bağlantı bulununcaya kadar her birini sırayla deneyerek gruptan bir tane seçer. Bağlantıların denendiği sıra, aday kanalların istemci kanal ağırlığı ve bağlantı benzerliği değerlerinden etkilenir. Gruptaki kuyruk yöneticilerinden hiçbiri bağlantı için uygun değilse, çağrı başarısız olur. Her kuyruk yöneticisi yalnızca bir kez denenir. Ad için tek başına bir yıldız imi belirtilirse, somutlama tanımlı bir varsayılan kuyruk yöneticisi grubu kullanılır.

Kuyruk yöneticisi grupları yalnızca MQ-istemcisi ortamında çalışan uygulamalar için desteklenir; istemci olmayan bir uygulama yıldız işaretiyle başlayan bir kuyruk yöneticisi adını belirtirse çağrı başarısız olur. Grup, gruptaki her kuyruk yöneticisiyle iletişim kurmak için, aynı kuyruk yöneticisi adıyla (yıldız imi olmadan belirtilen ad) birden çok istemci bağlantı kanalı tanımlaması sağlanarak tanımlanır. Varsayılan grup, her biri boş bir kuyruk yöneticisi adı olan bir ya da daha çok istemci bağlantı kanalı tanımlı sağlanarak tanımlanır (bu nedenle, tümü boş bir ad belirtilmesi, istemci uygulaması için ad için tek bir yıldız imi belirlenmesiyle aynı etkiye sahiptir).

Bir grubun bir kuyruk yöneticisine bağlandıktan sonra, bir uygulama, uygulamanın bağlandığı kuyruk yöneticisinin adını (*yemel kuyruk yöneticisi*) belirtmek için, iletideki ve nesne tanımlayıcılarındaki kuyruk yöneticisi adı alanlarında tipik olarak boşluk belirtebilir. Uygulamanın bu adı bilmesi gerekiyorsa, **QMgrName** kuyruk yöneticisi özneteliğini sorgulamak için MQINQ çağrısına bakın.

Bağlantı adına bir yıldız imi konması, uygulamanın gruptaki belirli bir kuyruk yöneticisine bağlanmasına bağlı olmadığı anlamına gelir. Uygun uygulamalar şunlardır:

- İletileri koyan, ancak ileti almayan uygulamalar.
- İstek iletileri koyan ve yanıt iletilerini *geçici dinamik* kuyruktan alan uygulamalar.

Uygun olmayan uygulamalar, belirli bir kuyruk yöneticisindeki belirli bir kuyruktan ileti alması gerekenlerdir; bu tür uygulamalar adın başına yıldız işareti koymamalıdır.

Bir yıldız işareti belirlerseniz, adın geri kalan kısmının uzunluk üst sınırı 47 karakterdir.

Bu değiştirgenin uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir.

Hconn.

Tip: MQHCONN-çıkış

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. Bunu, uygulama tarafından yayınlanan sonraki tüm ileti kuyruklama çağrılarında belirtin. MQDISC çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde geçerli olmaz.

IBM MQ artık mqm kitaplığını sunucu paketlerinin yanı sıra istemci paketleriyle de sağlar. Bu, mqm kitaplığında bulunan bir MQI çağrısı yapıldığında, bağlantı tipinin bir istemci mi, yoksa sunucu bağlantısı mı olduğunu görmek için denetlendiği ve ardından doğru temel çağrı yapıldığı anlamına gelir. Bu nedenle, Hconn iletilen bir çıkış artık mqm kitaplığına bağlanabilir, ancak bir istemci kuruluşunda kullanılabilir.

İşle kapsamı: Döndürülen tanıtıcı kapsamı, kuyruk yöneticisine (MQCONN ya da MQCONNX) bağlanmak için kullanılan çağrıya bağlıdır. Kullanılan çağrı MQCONNX ise, tanıtıcı kapsamı MQCNO yapısının *Options* alanında belirtilen MQCNO_HANDLE_SHARE_* seçeneğine de bağlıdır.

- Çağrı MQCONN ise ya da MQCNO_HANDLE_SHARE_NONE seçeneği belirtildiyse, döndürülen tanıtıcı *paylaşılmayan* bir tanıtıcıdır.

Paylaşılmayan bir tutamacın kapsamı, uygulamanın çalıştığı platform tarafından desteklenen en küçük koştur işleme birimidir (ayrıntılar için bkz. [Çizelge 546 sayfa 656](#)); tanıtıcı, çağrı yayınlandığı koştur işleme birimi dışında geçerli değildir.

- MQCNO_HANDLE_SHARE_BLOCK ya da MQCNO_HANDLE_SHARE_NO_BLOCK seçeneğini belirtirseniz, döndürülen tanıtıcı *paylaşılan* bir tanıtıcıdır.

Paylaşılan tanıtıcı kapsamı, çağrının verildiği iş parçacığına sahip olan işlemdir; tanıtıcı, o işleme ait olan herhangi bir iş parçacığından kullanılabilir. Her platform iş parçacıklarını desteklemez.

- MQCONN ya da MQCONNX çağrısı MQCC_FAILED ' e eşit tamamlama koduyla başarısız olursa, Hconn değeri tanımsız olur.

Çizelge 546. Çeşitli platformlardaki paylaşılmayan tutamaçların kapsamı	
Hizmet olarak sunulan	Paylaşılmayan tanıtıcı kapsamı
z/OS	<ul style="list-style-type: none">• CICS: CICS görevi• IMS: görev, bir sonraki eşitleme noktasına kadar (görevin alt görevleri hariç)• z/OS toplu iş ve TSO: görev (görevin alt görevleri hariç)
IBM i	İş
AIX and Linux	Başlık
32 bit Windows uygulamaları	Başlık
64 bit Windows uygulamaları	Başlık

z/OS ' da CICS uygulamaları için döndürülen değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_ALREADY_CONNECTED

(2002, X'7D2') Uygulama zaten bağlı.

MQRC_CLUSTER_EXIT_LOAD_ERROR (Çıkış)

(2267, X'8DB') Küme iş yükü çıkışı yüklenemiyor.

MQRC_SSL_ALREADY_KULLANIMA_HAZIRLANDI

(2391, X' 957 ') SSL zaten başlatıldı.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_CONN_LOAD_ERROR (UYGULAMA_HATASI)

(2129, X'851 ') Bağdaştırıcı bağlantı modülü yüklenemiyor.

MQRC_ADAPTER_DEFS_HATA

(2131, X'853 ') Bağdaştırıcı altsistemi tanımlama modülü geçersiz.

MQRC_ADAPTER_DEFS_LOAD_ERROR

(2132, X'854 ') Bağdaştırıcı altsistemi tanımlama modülü yüklenemiyor.

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ADAPTER_STORAGE_SHORTAGE

(2127, X'84F') Bağdaştırıcı için bellek yetersiz.

MQRC_ANOTHER_Q_MGR_CONNECTED

(2103, X'837 ') Başka bir kuyruk yöneticisi zaten bağlandı.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_INIT_ERROR (UYGULAMA_HATASI)

(2375, X' 947 ') API çıkışını başlatma başarısız oldu.

MQRC_API_EXIT_TERM_HATA

(2376, X' 948 ') API çıkışı sonlandırılmadı.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CONN_ID_IN_USE

(2160, X'870 ') Bağlantı tanıtıcısı kullanımda.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_CONNECTION_HATA

(2273, X'8E1') MQCONN çağrısı işlenirken hata oluştu.

MQRC_CONNECTION_KULLANILAMIYOR

(2568, X'A08') Kuyruk yöneticisi yürürlükteki kuruluştaki istenen bağlantı tipini sağlayamadığı zaman MQCONN ya da MQCONNX çağrısında oluşur. Yalnızca sunucu kuruluşunda istemci bağlantısı kurulamaz. Yalnızca istemci kuruluşunda yerel bağlantı kurulamaz.

MQRC_CONNECTION QUIESCING

(2202, X'89A') Bağlantı susturuyor.

MQRC_CONNECTION_DURDURULUYOR

(2203, X'89B') Bağlantı kapatılıyor.

MQRC_CRYPTO_HARDWARE_HATA

(2382, X'94E') Şifreleme donanımı yapılandırma hatası.

MQRC_DUPLICATE_RECOV_COORD

(2163, X'873 ') Kurtarma koordinatörü var.

MQRC_ENVIRONMENT_ERROR

(2012, X'7DC') Çağrısı ortamda geçerli değil.

Buna ek olarak, MQCONNX çağrısında, bir CICS ya da IMS uygulamasından [“MQCSP-Güvenlik değiştirgeleri”](#) sayfa 334 denetim öbeğini iletiyor.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_ANASISTEM_YOK

(2538, X'9EA') Bir istemciden kuyruk yöneticisine bağlanmak için MQCONN çağrısı yayınlandı, ancak uzak sisteme etkileşim ayırma girişimi başarısız oldu.

MQRC_INSTALLATION_MISMATCH

(2583, X'A17') Kuyruk yöneticisi kuruluşu ile seçilen kitaplık arasında uyumsuzluk var.

MQRC_KEY_REPOSITORY_HATA

(2381, X'94D') Anahtar havuzu geçersiz.

MQRC_MAX_CONNS_LIMIT_SINIRINA ulaşıldı

(2025, X'7E9') Bağlantı sayısı üst sınırına ulaşıldı.

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_OPEN_FAILED

(2137, X'859 ') Nesne başarıyla açılmadı.

MQRC_Q_MGR_AD_HATASI

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRC_Q_MGR_YOK

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.

MQRC_Q_MGR QUIESCING

(2161, X'871 ') Kuyruk yöneticisi susturuyor.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_SECURITY_HATA

(2063, X'80F') Güvenlik hatası oluştu.

MQRC_SSL_INITIALIZION_HATA

(2393, X' 959 ') SSL kullanıma hazırlama hatası.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. MQCONN çağrısı kullanılarak bağlantı kurulan kuyruk yöneticisine *yerel kuyruk yöneticisi* adı verilir.
2. Yerel kuyruk yöneticisinin sahip olduğu kuyruklar, uygulamada yerel kuyruklar olarak görünür. İletiler takılabilir ve bu kuyruklardan ileti alabilir.

Yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun sahip olduğu paylaşılan kuyruklar, uygulamaya yerel kuyruklar olarak görünür. İletiler takılabilir ve bu kuyruklardan ileti alabilir.


Uzak kuyruk yöneticilerinin iyeliğindeki kuyruklar uzak kuyruklar olarak görünür. Bu kuyruklara ileti koymak mümkündür, ancak bu kuyruklardan ileti almak mümkün değildir.
3. Bir uygulama çalışırken kuyruk yöneticisi başarısız olursa, uygulamanın sonraki IBM MQ çağrılarında kullanılacak yeni bir bağlantı tanıtıcısı elde etmek için MQCONN çağrısını yeniden yürütmesi gerekir. Uygulama, çağrı başarılı oluncaya kadar düzenli aralıklarla MQCONN çağrısına devam edebilir.

Bir uygulama kuyruk yöneticisine bağlı olup olmadığından emin değilse, uygulama bağlantı tanıtıcısı elde etmek için güvenli bir şekilde bir MQCONN çağrısı verebilir. Uygulama zaten bağlıysa, döndürülen tanıtıcı önceki MQCONN çağrısıyla aynıdır, ancak tamamlama kodu MQCC_WARNING ve neden kodu MQRC_ALREADY_CONNECTED.
4. Uygulama IBM MQ çağrılarını kullanmayı bitirdiğinde, uygulamanın kuyruk yöneticisiyle bağlantısını kesmek için MQDISC çağrılarını kullanması gerekir.
5. MQCONN çağrısı MQCC_FAILED ' e eşit tamamlama koduyla başarısız olursa, Hconn değeri tanımsız olur.
6. z/OS'ta:
 - Toplu iş, TSO ve IMS uygulamaları, diğer IBM MQ çağrılarını kullanmak için MQCONN çağrılarını yayınlamalıdır. Bu uygulamalar eşzamanlı olarak birden çok kuyruk yöneticisine bağlanabilir.

Kuyruk yöneticisi başarısız olursa, yeni bir bağlantı tanıtıcısı elde etmek için uygulamanın kuyruk yöneticisi yeniden başlatıldıktan sonra çağrıyı yeniden yürütmesi gerekir.

IMS uygulamaları MQCONN çağrılarını, önceden bağlanmış olsalar bile, yinelenen olarak yürütebilse de, çevrimiçi ileti işleme programları (MPPs) için bu önerilmez.
 - CICS uygulamalarının diğer IBM MQ çağrılarını kullanmak için MQCONN çağrısı yayınlamak zorunda değildir, ancak isterlerse bunu yapabilir; hem MQCONN çağrısı hem de MQDISC çağrısı kabul edilir. Ancak, aynı anda birden çok kuyruk yöneticisiyle bağlantı kurulamaz.

Kuyruk yöneticisi başarısız olursa, kuyruk yöneticisi yeniden başlatıldığında bu uygulamalar otomatik olarak yeniden bağlanır ve MQCONN çağrısına gerek yoktur.
7. z/OS' da kullanılabilir kuyruk yöneticilerini tanımlamak için:
 - Toplu iş uygulamalarında, sistem programcılarını CSQBDEF makrosunu kullanarak varsayılan kuyruk yöneticisi adını ya da kuyruk paylaşım grubu adını tanımlayan bir birim (CSQBDEFV) yaratabilir.
 - IMS uygulamalarında, sistem programcılarını kullanılabilir kuyruk yöneticilerinin adlarını tanımlayan ve varsayılan kuyruk yöneticisini belirten bir modül (CSQQDEFV) yaratmak için CSQQDEFX makrosunu kullanabilir.
Ayrıca, her kuyruk yöneticisi IMS denetim bölgesinde ve o kuyruk yöneticisine erişen her bağımlı bölgede tanımlanmalıdır. Bunu yapmak için IMS içinde bir altsistem üyesi oluşturmanız gerekir. PROCLIB kitaplığı ve altsistem üyesini ilgili IMS bölgelerine tanıtır. Bir uygulama, IMS bölgesi için altsistem üyesinde tanımlı olmayan bir kuyruk yöneticisine bağlanmayı denerse, uygulama olağandışı biter.

 Bu makroları kullanma hakkında daha fazla bilgi için bkz. [Müşteri kullanımı için tasarlanmış makrolar](#).

8. IBM üzerinde, olağandışı sona eren programların kuyruk yöneticisiyle bağlantısı otomatik olarak kesilmez. MQCONN ya da MQCONNX çağrısının tamamlama kodu MQCC_WARNING ve neden kodu MQRC_ALREADY_CONNECTED döndürme olasılığına izin verecek uygulamaları yazın. Bu durumda döndürülen bağlantı tanıtıcısını olağan biçimde kullanın.

C çağrısı

```
MQCONN (QMgrName, &Hconn, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
MQCHAR48 QMgrName; /* Name of queue manager */
MQHCONN  Hconn;    /* Connection handle */
MQLONG   CompCode; /* Completion code */
MQLONG   Reason;   /* Reason code qualifying CompCode */
```

COBOL çağrısı

```
CALL 'MQCONN' USING QMGRNAME, HCONN, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağrısı

```
call MQCONN (QMGrName, Hconn, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL MQCONN, (QMGRNAME, HCONN, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
QMGRNAME DS CL48 Name of queue manager
HCONN DS F Connection handle
COMPCODE DS F Completion code
REASON DS F Reason code qualifying COMPCODE
```

Visual Basic çağrısı

```
MQCONN QMgrName, Hconn, CompCode, Reason
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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-Connect kuyruk yöneticisi (genişletilmiş)

MQCONNX çağrısı, bir uygulama programını bir kuyruk yöneticisine bağlar. Sonraki IBM MQ çağrılarında uygulama tarafından kullanılan bir kuyruk yöneticisi bağlantı tanıtıcısı sağlar.

MQCONNX çağrısı MQCONN çağrısına benzer, ancak MQCONNX, çağrılarının çalışma şeklini denetlemek için seçeneklerin belirtilmesine izin verir.

- Bu çağrı, tüm IBM MQ sistemlerinde ve bu sistemlere bağlı IBM MQ istemcilerinde desteklenir.

Yalnızca sunucu kuruluşunda istemci bağlantısı kurulamaz ve yalnızca istemci kuruluşunda yerel bağlantı kurulamaz.

Sözdizimi

MQCONNX (*QMgrName, ConnectOpts, Hconn, CompCode, Reason*)

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Ayrıntılar için “MQCONN-Bağlantı kuyruğu yöneticisi” sayfa 654 içinde açıklanan **QMgrName** değiştirgesine bakın.

ConnectOpts

Tip: MQCNO-giriş/çıkış

Ayrıntılar için bkz. “MQCNO-Bağlantı seçenekleri” sayfa 316.

Hconn.

Tip: MQHCONN-çıkış

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. Bunu, uygulama tarafından yayınlanan sonraki tüm ileti kuyruklama çağrılarında belirtin. MQDISC çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde geçerli olmaz.

IBM MQ artık mqm kitaplığını sunucu paketlerinin yanı sıra istemci paketleriyle de sağlar. Bu, mqm kitaplığında bulunan bir MQI çağrısı yapıldığında, bağlantı tipinin bir istemci mi, yoksa sunucu bağlantısı mı olduğunu görmek için denetlendiği ve ardından doğru temel çağrı yapıldığı anlamına gelir. Bu nedenle, *Hconn* iletilen bir çıkış artık mqm kitaplığına bağlanabilir, ancak bir istemci kuruluşunda kullanılabilir.

İşle kapsamı: Döndürülen tanıtıcı kapsamı, kuyruk yöneticisine (MQCONN ya da MQCONNX) bağlanmak için kullanılan çağrıya bağlıdır. Kullanılan çağrı MQCONNX ise, tanıtıcı kapsamı MQCNO yapısının *Options* alanında belirtilen MQCNO_HANDLE_SHARE_* seçeneğine de bağlıdır.

- Çağrı MQCONN ise ya da MQCNO_HANDLE_SHARE_NONE seçeneği belirtildiyse, döndürülen tanıtıcı *paylaşılmayan* bir tanıtıcıdır.

Paylaşılmayan bir tutamacın kapsamı, uygulamanın çalıştığı platform tarafından desteklenen en küçük koşt işleme birimidir (ayrıntılar için bkz. [Çizelge 547 sayfa 662](#)); tanıtıcı, çağrı yayınlandığı koşt işleme birimi dışında geçerli değildir.

- MQCNO_HANDLE_SHARE_BLOCK ya da MQCNO_HANDLE_SHARE_NO_BLOCK seçeneğini belirtirseniz, döndürülen tanıtıcı *paylaşılan* bir tanıtıcıdır.

Paylaşılan tanıtıcı kapsamı, çağrının verildiği iş parçacığına sahip olan işlemdir; tanıtıcı, o işleme ait olan herhangi bir iş parçacığından kullanılabilir. Her platform iş parçacıklarını desteklemez.

- MQCONN ya da MQCONNX çağrısı MQCC_FAILED ' e eşit tamamlama koduyla başarısız olursa, Hconn değeri tanımsız olur.

<i>Çizelge 547. Çeşitli platformlardaki paylaşılmayan tutamaçların kapsamı</i>	
Hizmet olarak sunulan	Paylaşılmayan tanıtıcı kapsamı
z/OS	<ul style="list-style-type: none">• CICS: CICS görevi• IMS: görev, bir sonraki eşitleme noktasına kadar (görevin alt görevleri hariç)• z/OS toplu iş ve TSO: görev (görevin alt görevleri hariç)
IBM i	İş
AIX and Linux	Başlık
32 bit Windows uygulamaları	Başlık
64 bit Windows uygulamaları	Başlık

z/OS ' da CICS uygulamaları için döndürülen değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

CompCode

Tip: MQLONG-output

Ayrıntılar için “MQCONN-Bağlantı kuyruğu yöneticisi” [sayfa 654](#) içinde açıklanan **CompCode** değiştirgesine bakın.

Neden

Tip: MQLONG-output

MQCONN ve MQCONNX çağrılarıyla aşağıdaki kodlar döndürülebilir. MQCONNX çağrısının döndüreceği ek kodların listesi için aşağıdaki kodlara bakın.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_ALREADY_CONNECTED

(2002, X'7D2') Uygulama zaten bağlı.

MQRC_CLUSTER_EXIT_LOAD_ERROR (Çıkış)

(2267, X'8DB') Küme iş yükü çıkışı yüklenemiyor.

MQRC_SSL_ALREADY_KULLANIMA_HAZIRLANDI

(2391, X' 957 ') SSL zaten başlatıldı.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_CONN_LOAD_ERROR (UYGULAMA_HATASI)

(2129, X'851 ') Bağdaştırıcı bağlantı modülü yüklenemiyor.

MQRC_ADAPTER_DEFS_HATA

(2131, X'853 ') Baędařtırıcı altsistemi tanımlama modülü geersiz.

MQRC_ADAPTER_DEFS_LOAD_ERROR

(2132, X'854 ') Baędařtırıcı altsistemi tanımlama modülü yüklenemiyor.

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Baędařtırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Baędařtırıcı hizmet modülü yüklenemiyor.

MQRC_ADAPTER_STORAGE_SHORTAGE

(2127, X'84F') Baędařtırıcı için bellek yetersiz.

MQRC_ANOTHER_Q_MGR_CONNECTED

(2103, X'837 ') Bařka bir kuyruk yöneticisi zaten baęlandı.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API ıkışı bařarisız oldu.

MQRC_API_EXIT_INIT_ERROR (UYGULAMA_HATASI)

(2375, X' 947 ') API ıkışını bařlatma bařarisız oldu.

MQRC_API_EXIT_TERM_HATA

(2376, X' 948 ') API ıkışı sonlandırılmadı.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluęu parametresi geersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI aęırısı.

MQRC_CONN_ID_IN_USE

(2160, X'870 ') Baęlantı tanıtıcısı kullanımda.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine baęlantı kaybedildi.

MQRC_CONNECTION_HATA

(2273, X'8E1') MQCONN aęırısı işlenirken hata oluştu.

MQRC_CONNECTION_KULLANILAMIYOR

(2568, X'A08') Kuyruk yöneticisi yürürlükteki kuruluřta istenen baęlantı tipini saęlayamadıęı zaman MQCONN ya da MQCONNX aęırısında oluşur. Yalnızca sunucu kuruluřunda istemci baęlantısı kurulamaz. Yalnızca istemci kuruluřunda yerel baęlantı kurulamaz.

MQRC_CONNECTION QUIESCING

(2202, X'89A') Baęlantı susturuyor.

MQRC_CONNECTION DURDURULUYOR

(2203, X'89B') Baęlantı kapatılıyor.

MQRC_CRYPTO_HARDWARE_HATA

(2382, X'94E') řifreleme donanımı yapılandırma hatası.

MQRC_DUPLICATE_RECOV_COORD

(2163, X'873 ') Kurtarma koordinatörü var.

MQRC_ENVIRONMENT_ERROR

(2012, X'7DC') aęırısı ortamda geerli deęil.

Buna ek olarak, MQCONNX aęırısında, bir CICS ya da IMS uygulamasından “MQCSP-Güvenlik deęiřtirgeleri” sayfa 334 denetim öbeęini iletiyor.

MQRC_HCONN_HATA

(2018, X'7E2') Baęlantı tanıtıcısı geersiz.

MQRC_ANASISTEM_YOK

(2538, X'9EA') Bir istemciden kuyruk yöneticisine bağlanmak için MQCONN çağrısı yayınlandı, ancak uzak sisteme etkileşim ayırma girişimi başarısız oldu.

MQRC_INSTALLATION_MISMATCH

(2583, X'A17') Kuyruk yöneticisi kurulumu ile seçilen kitaplık arasında uyumsuzluk var.

MQRC_KEY_REPOSITORY_HATA

(2381, X'94D') Anahtar havuzu geçersiz.

MQRC_MAX_CONNS_LIMIT_SINIRINA ulaşıldı

(2025, X'7E9') Bağlantı sayısı üst sınırına ulaşıldı.

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_OPEN_FAILED

(2137, X'859 ') Nesne başarıyla açılmadı.

MQRC_Q_MGR_AD_HATASI

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRC_Q_MGR_YOK

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.

MQRC_Q_MGR_QUIESCING

(2161, X'871 ') Kuyruk yöneticisi susturuyor.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_SECURITY_HATA

(2063, X'80F') Güvenlik hatası oluştu.

MQRC_SSL_INITIALIZION_HATA

(2393, X' 959 ') SSL kullanıma hazırlama hatası.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

MQCONN çağrısıyla aşağıdaki ek neden kodları döndürülebilir:

CompCode MQCC_FAILED ise:

MQRC_AIR_HATA

(2385, X' 951 ') Kimlik doğrulama bilgileri kaydı geçerli değil.

MQRC_AUTH_INFO_CONN_NAME_ERROR (KUYRUK_AD_HATASI)

(2387, X' 953 ') Kimlik doğrulama bilgileri bağlantı adı geçersiz.

MQRC_AUTH_INFO_REC_COUNT_HATA

(2383, X'94F') Kimlik doğrulama bilgileri kayıt sayısı geçersiz.

MQRC_AUTH_INFO_REC_HATA

(2384, X' 950 ') Kimlik doğrulama bilgileri kayıt alanları geçerli değil.

MQRC_AUTH_INFO_TYPE_ERROR

(2386, X' 952 ') Kimlik doğrulama bilgisi tipi geçerli değil.

MQRC_CD_ERROR

(2277, X'8E5') Kanal tanımı geçerli değil.

MQRC_CLIENT_CONN_ERROR (MQRC_CLIENT_HATASI)

(2278, X'8E6') İstemci bağlantısı alanları geçersiz.

MQRC_CNO_HATA

(2139, X'85B') Bağlantı seçenekleri yapısı geçersiz.

MQRC_CONN_TAG_IN_USE

(2271, X'8DF') Bağlantı etiketi kullanımda.

MQRC_CONN_TAG_NOT_USABLE

(2350, X'92E') Bağlantı etiketi kullanılamıyor.

MQRC_CSP_HATA

(2595, X'A23') MQCSP yapısı geçerli değil.

V 9.3.4 MQRC_FUNCTION_NOT_SUPPORTED

(2298, X'8FA') İstenen işlev yürürlükteki ortamda kullanılamıyor.

MQRC_LDAP_PASSWORD_HATASI

(2390, X' 956 ') LDAP parolası geçerli değil.

MQRC_LDAP_USER_NAME_ERROR (MQRC_LDAP_USER_NAME_ERROR)

(2388, X' 954 ') LDAP kullanıcı adı alanları geçerli değil.

MQRC_LDAP_USER_NAME_LENGTH_ERR

(2389, X' 955 ') LDAP kullanıcı adı uzunluğu geçersiz.

MQRC_OPTIONS_HATA

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_SCO_ERROR

(2380, X'94C') SSL yapılandırma seçenekleri yapısı geçerli değil.

MQRC_SSL_CONFIG_HATA

(2392, X' 958 ') SSL yapılandırma hatası.

MQRC_TOKEN_ZAMANDAMGASI_GEÇERLİ değil

(2064, X'810 ') Kimlik doğrulama belirteci henüz geçerli değil ya da süresi dolmuş.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

Visual Basic programlama dili için a'ya § daki nokta kullanılır:

- **ConnectOpts** değiştirgesi MQCNO tipinde olarak bildirildi. Uygulama bir IBM MQ MQI clientolarak çalışıyorsa ve istemci-bağlantı kanalının değiştirgelerini belirtmek istiyorsanız, uygulamanın MQCNO yapısı yerine çağrıda bir MQCNOCD yapısı belirtebilmesi için **ConnectOpts** değiştirgesini Anytipinde olarak bildirin. Ancak bu, **ConnectOpts** parametresinin doğru veri tipi olduğundan emin olmak için denetlenemediği anlamına gelir.

C çağırısı

```
MQCONN (QMgrName, &ConnectOpts, &Hconn, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağırısı

```
CALL 'MQCONN' USING QMGRNAME, CONNECTOPTS, HCONN, COMPCODE,
REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

** Name of queue manager
01 QMGRNAME PIC X(48).
** Options that control the action of MQCONN
01 CONNECTOPTS.
COPY CMQCNNOV.
** 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.

```

PL/I çağırısı

```
call MQCONN (QMgrName, ConnectOpts, Hconn, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

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 */

```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQCONN, (QMGRNAME,CONNECTOPTS,HCONN,COMPCODE,REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

QMGRNAME DS CL48 Name of queue manager
CONNECTOPTS CMQCNNOA , Options that control the action of MQCONN
HCONN DS F Connection handle
COMPCODE DS F Completion code
REASON DS F Reason code qualifying COMPCODE

```

Visual Basic çağırısı

```
MQCONN QMgrName, ConnectOpts, Hconn, CompCode, Reason
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

Dim QMgrName As String*48 'Name of queue manager'
Dim ConnectOpts As MQCNO 'Options that control the action of'
'MQCONN'
Dim Hconn As Long 'Connection handle'
Dim CompCode As Long 'Completion code'
Dim Reason As Long 'Reason code qualifying CompCode'

```

MQCRTMH-İleti tanıtıcısı yarat

MQCRTMH çağırısı bir ileti tanıtıcısı döndürür.

Bir uygulama sonraki ileti kuyruklama çağrılarında MQCRTMH çağırısı kullanabilir:

- İleti tanıtıcısının bir özelliğini ayarlamak için [MQSETMP](#) çağırısını kullanın.
- İleti tanıtıcısının bir özelliğinin değerini sorgulamak için [MQINQMP](#) çağırısını kullanın.
- İleti tanıtıcısının bir özelliğini silmek için [MQDLTMP](#) çağırısını kullanın.

İleti tanıtıcısı MQPUT ve MQPUT1 çağrılarında, ileti tanıtıcısının özelliklerini, konmakta olan iletiyle ilişkilendirmek için kullanılabilir. Benzer şekilde, MQGET çağrısında bir ileti tanıtıcısı belirtilerek, MQGET çağrısı tamamlandığında ileti tanıtıcısı kullanılarak, alınmakta olan iletinin özelliklerine erişilebilir.

İleti tanıtıcısını silmek için [MQDLTMH](#) komutunu kullanın.

Sözdizimi

MQCRTMH (*Hconn, CrtMsgHOpts, Hmsg, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü. Kuyruk yöneticisine yönelik bağlantı geçerliliğini keserse ve ileti tanıtıcısında IBM MQ çağrısı çalışmıyorsa, iletiyi silmek için [MQDLTMH](#) örtük olarak çağrılır.

Diğer bir seçenek olarak, aşağıdaki değeri belirtebilirsiniz:

MQHC_UNASSOCIATED_HCONN

Bağlantı tanıtıcısı, belirli bir kuyruk yöneticisine yönelik bir bağlantıyı göstermiyor.

Bu değer kullanıldığında, kendisine ayrılan saklama alanını serbest bırakmak için ileti tanıtıcısı [MQDLTMH](#) ' e yönelik belirtik bir çağrıyla silinmelidir; IBM MQ ileti tanıtıcısını hiçbir zaman örtük olarak silmez.

İleti tanıtıcısını yaratan iş parçacığında kurulmuş bir kuyruk yöneticisiyle en az bir geçerli bağlantı olmalıdır; tersi durumda, çağrı MQRC_HCONN_ERROR ile başarısız olur.

Tek bir sistemde birden çok kuruluşu olan bir ortamda, MQHC_UNASSOCIATED_HCONN değeri, sürece yüklenen ilk kuruluşla birlikte kullanımla sınırlıdır. İleti tanıtıcısı farklı bir kuruluşla sağlanırsa, MQRC_HMSG_NOT_AVAILABLE neden kodu döndürülür.

z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve *Hconn* için aşağıdaki değeri belirtebilirsiniz:

MQHC_DEF_CONN

Varsayılan bağlantı tanıtıcısı

CrtMsgHOpts

Tip: MQCMHO-giriş

MQCRTMH işlemini denetleyen seçenekler. Ayrıntılar için bkz. [MQCMHO](#) .

İt

Tip: MQHMSG-çıkış

Çıkışta, ileti tanıtıcısının özelliklerini ayarlamak, sorgulamak ve silmek için kullanılacak bir ileti tanıtıcısı döndürülür. Başlangıçta ileti tanıtıcısı özellik içermiyor.

İleti tanıtıcısının ilişkili bir ileti tanımlayıcısı da vardır. Başlangıçta bu, varsayılan değerleri içerir. İlişkili ileti tanımlayıcı alanlarının değerleri MQSETMP ve MQINQMP çağrıları kullanılarak ayarlanabilir ve sorilebilir. MQDLTMP çağrısı, ileti tanımlayıcısının bir alanını varsayılan değerine geri çevirir.

Hconn değiştirgesi MQHC_UNASSOCIATED_HCONN değeri olarak belirtilirse, döndürülen ileti tanıtıcısı MQGET, MQPUT ya da MQPUT1 çağrılarında işlem birimi içindeki herhangi bir bağlantıyla kullanılabilir, ancak bir kerede tek bir IBM MQ çağrısı tarafından kullanılabilir. İkinci bir IBM MQ çağrısı aynı ileti tanıtıcısını kullanmayı denediğinde tanıtıcı kullanılırsa, ikinci IBM MQ çağrısı MQRC_MSG_HANDLE_IN_USE neden koduyla başarısız olur.

Hconn değiştirgesi MQHC_UNASSOCIATED_HCONN değilse, döndürülen ileti tanıtıcısı yalnızca belirtilen bağlantıda kullanılabilir.

Bu ileti işlecinin kullanıldığı sonraki MQI çağrılarında aynı *Hconn* değiştirge değeri kullanılmalıdır:

- MQDLTMH
- MQSETMP
- MQINQMP
- MQDLTMP
- MQMHBUF
- MQBUFMH

İleti tanıtıcısı için MQDLTMH çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde, döndürülen ileti tanıtıcısı geçerli olmaktan çıkar. İleti tanıtıcısı yaratıldığında belirli bir bağlantı sağlanırsa ve kuyruk yöneticisine yönelik bağlantı geçerliliğini keserse (örneğin, MQDBC çağrılırsa) MQDLTMH örtük olarak çağrılır.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'089C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CMHO_HATA

(2461, X'099D') İleti tanıtıcısı yaratma seçenekleri yapısı geçersiz.

MQRC_CONNECTION_BROKEN

(2273, X'7D9') Kuyruk yöneticisiyle bağlantı kesildi.

MQRC_HANDLE_NOT_YOK

(2017, X'07E1') Başka tanıtıcı yok.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_HMSG_HATA

(2460, X'099C') İleti tanıtıcısı göstergesi geçersiz.

MQRC_OPTIONS_HATA

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. İletiler ve neden kodları.

C

```
MQCRTMH (Hconn, &CrtMsgHOpts, &Hmsg, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
MQHCONN  Hconn;          /* Connection handle */
MQCMHO   CrtMsgHOpts;   /* Options that control the action of MQCRTMH */
MQHMSG   Hmsg;          /* Message handle */
MQLONG   CompCode;      /* Completion code */
MQLONG   Reason;        /* Reason code qualifying CompCode */
```

COBOL

```
CALL 'MQCRTMH' USING HCONN, CRTMSGHOPTS, HMSG, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
HCONN          DS      F      Connection handle
CRTMSGHOPTS    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-Denetim geri çağrılar

MQCTL çağırısı, geri çağırımlar üzerinde denetleme işlemleri gerçekleştirir ve bir bağlantı için açılan nesne tanıtıcıları.

Sözdizimi

MQCTL (*Hconn, İşlem, ControlOpts, CompCode, Neden*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağırısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağırısı atlanabilir ve *Hconn*'in aşağıdaki özel değeri belirtebilirsiniz:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

İşlem

Tip: MQLONG-input

Belirtilen nesne tanıtıcısı için tanımlanan geri çağırma sırasında işlenmekte olan işlem. Aşağıdaki seçeneklerden birini ve yalnızca birini belirtmeniz gerekir:

MQOP_START

Belirtilen bağlantı tanıtıcısı için, tanımlı tüm ileti tüketicisi işlevleri için ileti tüketimini başlatın.

Geri çağırımlar, uygulama iş parçacıklarından farklı olan, sistem tarafından başlatılan bir iş parçacığında çalıştırılır.

Bu işlem, sisteme sağlanan bağlantı tanıtıcısının denetimini sağlar. Tüketici iş parçacığından başka bir iş parçacığı tarafından verilebilecek tek MQI çağrılarları şunlardır:

- MQOP_STOP İşlemiyle MQCTL
- MQOP_SUSPEND İşlemiyle MQCTL
- MQDISC-HConn bağlantısını kesmeden önce MQOP_STOP İşlemiyle MQCTL 'yi gerçekleştirir.

Bağlantı tanıtıcısı başlatılırken bir IBM MQ API çağırısı verilirse ve çağrı bir ileti tüketicisi işlevinden kaynaklanmazsa, MQRC_HCONN_ASYNC_ACTIVE döndürülür.

MQCBCT_START_CALL sırasında bir ileti tüketicisi bağlantıyı durdurursa, MQCTL çağırısı MQRC_CONNECTION_DURDURULDU hata nedeni koduyla döner.

Bu, bir tüketici işlevinde yayınlanabilir. Geri çağırma yordamına aynı bağlantı için tek amacı, önceden yayınlanan bir MQOP_STOP işlemini iptal etmektir.

Bu seçenek, şu ortamlarda desteklenmez: CICS on z/OS ya da uygulama, iş parçacıklı olmayan bir IBM MQ kitaplığıyla bağlıysa.

MQOP_START_WAIT

Belirtilen bağlantı tanıtıcısı için, tanımlı tüm ileti tüketicisi işlevleri için ileti tüketimini başlatın.

İleti tüketicileri aynı iş parçacığında çalışır ve denetim, şu zamana kadar MQCTL 'yi çağırana döndürülmez:

- MQCTL MQOP_STOP ya da MQOP_SUSPEND işlemlerinin kullanımıyla serbest bırakıldı ya da
- Tüm tüketici yordamlarının kaydı kaldırıldı ya da askıya alındı.

Tüm tüketicilerin kaydı kaldırılırsa ya da askıya alınırsa, örtük bir MQOP_STOP işlemi yayınlanır.

Bu seçenek, yürürlükteki bağlantı tanıtıcısı ya da başka bir bağlantı tanıtıcısı için, bir geri çağırma yordamı içinden kullanılamaz. Çağrı girişiminde bulunulursa, MQRC_ENVIRONMENT_ERROR ile geri döner.

MQOP_START_WAIT işlemi sırasında herhangi bir zamanda kayıtlı, askıya alınmamış tüketiciler yoksa, çağrı MQRC_NO_CALLBACKS_ACTIVE neden koduyla başarısız olur.

MQOP_START_WAIT işlemi sırasında bağlantı askıya alındıysa, MQCTL çağrısı MQRC_CONNECTION_SUSPENDED uyarı nedeni kodunu döndürür; bağlantı 'başlatıldı' olarak kalır.

Uygulama MQOP_STOP ya da MQOP_RESUME komutunu vermeyi seçebilir. Bu örnekte, MQOP_RESUME işlemi öbekleri.

Bu seçenek, tek iş parçacıklı istemcide desteklenmez.

MQOP_DUR

İletilerin tüketilmesini durdurun ve bu seçenek tamamlanmadan önce tüm tüketicilerin işlemlerini tamamlamalarını bekleyin. Bu işlem, bağlantı tanıtıcısını serbest bırakır.

Bir geri çağırma yordamı içinden verilirse, yordam çıkıncaya kadar bu seçenek yürürlüğe girmez. Okunan iletilere ilişkin tüketici yordamları tamamlandıktan sonra ve (istenirse) geri çağırma yordamlarına yönelik durdurma çağrılarında sonra başka ileti tüketici rutinleri çağrılmaz.

Bir geri çağırma yordamı dışında verilirse, okunan iletilere ilişkin tüketici yordamları tamamlanıncaya kadar ve geri çağırma durdurma çağrılarında (istendiye) sonra, denetim çağırana dönmez. Ancak geri çağırma kayıtlı olmaya devam ediyor.

Bu işlevin önden okuma iletileri üzerinde etkisi yoktur. Teslim edilecek başka ileti olup olmadığını saptamak için, tüketicilerin geri çağırma işlevinden MQCLOSE (MQCO_QUIESCE) çalıştırdığından emin olmanız gerekir.

MQOP_SUSPEND

İleti tüketimini duraklatır. Bu işlem, bağlantı tanıtıcısını serbest bırakır.

Bu, uygulamaya ilişkin iletilerin okunuşunda herhangi bir etkiye sahip değildir. İletileri uzun süre kullanmayı durdurmak istiyorsanız, kuyruğu kapatmayı ve tüketim devam ederken yeniden açmayı düşünün.

Bir geri çağırma yordamı içinden verilirse, yordam çıkıncaya kadar yürürlüğe girmez. Yürürlükteki yordam çıktıktan sonra başka ileti tüketicisi yordamı çağrılmayacak.

Bir geri çağırma dışında verilirse, denetim, yürürlükteki tüketici yordamı tamamlanıncaya ve başka çağrılınca kadar çağırana dönmez.

MQOP_RESUME

İletileri tüketmeye devam edin.

Bu seçenek genellikle ana uygulama iş parçacığından yayınlanır, ancak aynı yordamda daha önce yayınlanan bir askıya alma isteğini iptal etmek için bir geri çağırma yordamı içinden de kullanılabilir.

Bir MQOP_START_WAIT işlemi sürdürmek için MQOP_RESUME kullanılırsa, işlem öbekler.

ControlOpts

Tip: MQCTLO-giriş

MQCTL işlemi denetleyen seçenekler

Yapının ayrıntıları için bkz. [MQCTLO](#) .

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_CONV_LOAD_ERROR (UYGULAMA_HATASI)

(2133, X'855 ') Veri dönüştürme hizmetleri modülleri yüklenemiyor.

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

(2183, X'887 ') API çıkışı yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

MQRC_CALLBACK_LINK_HATA

(2487, X'9B7') Geri çağırma yordamı çağırılmıyor

MQRC_CALLBACK_NOT_KAYITLI

(2448, X' 990 ') Kayıtlı geri çağırma olmadığı için kayıt kaldırılmıyor, askıya alınamıyor ya da devam ettirilemiyor

MQRC_CALLBACK_ROUTINE_HATA

(2486, X'9B6') Bir MQOP_REGISTER çağrısında hem CallbackFunction , hem de CallbackName belirtildi.

Ya da CallbackFunction ya da CallbackName belirtildi, ancak kayıtlı geri çağırma işleviyle eşleşmiyor.

MQRC_CALLBACK_TYPE_HATA

(2483, X'9B3') CallBackTip alanı yanlış.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CBD_ERROR

(2444, X'98C') Seçenek öbeği yanlış.

MQRC_CBD_OPTIONS_HATA

(2484, X'9B4') Yanlış MQCBD seçenekleri alanı.

MQRC_CICS_WAIT_FAILED

(2140, X'85C') Bekleme isteği CICStarafından reddedildi.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_CONNECTION_NOT_YETKILI

(2217, X'8A9') Bağlantı için yetkili değil.

MQR_CONNECTION_QUIESCING
(2202, X'89A') Baęlantı susturuyor.

MQR_CONNECTION_DURDURULUYOR
(2203, X'89B') Baęlantı kapatılıyor.

MQR_CORREL_ID_HATA
(2207, X'89F') İliinti-tanıtıcı hatası.

MQR_ENVIRONMENT_ERROR
(2012, X'7DC') Çaęrısı ortamda geçerli deęil.

MQR_FUNCTION_NOT_SUPPORTED
(2298, X'8FA') İstlenen işlev yürürlükteki ortamda kullanılamıyor.

MQR_GET_INHIBE
(2016, X'7E0') Kuyruk için engellendi.

MQR_GLOBAL_UOW_CONFLICT
(2351, X'92F') Genel iş birimleri çakışması.

MQR_GMO_HATA
(2186, X'88A') Get-message seçenekleri yapısı geçersiz.

MQR_HANDLE_IN_USE_FOR_UOW
(2353, X' 931 ') Genel iş birimi için kullanımda.

MQR_HCONN_HATA
(2018, X'7E2') Baęlantı tanıtıcısı geçersiz.

MQR_HOBJ_HATA
(2019, X'7E3') Nesne tanıtıcısı geçerli deęil.

MQR_TUTARSIZ_GÖZAT
(2259, X'8D3') Tutarsız göz atma belirtimi.

MQR_TUTARSIZ_UOW
(2245, X'8C5') İş birimi belirtimi tutarsız.

MQR_INVALID_MSG_UNDER_CURSOR
(2246, X'8C6') İmleç altındaki ileti için geçerli deęil.

MQR_LOCAL_UOW_CONFLICT
(2352, X' 930 ') Genel iş birimi yerel iş birimiyle çakışıyor.

MQR_MATCH_OPTIONS_HATA
(2247, X'8C7') Eşleştirme seçenekleri geçerli deęil.

MQR_MAX_MSG_LENGTH_ERROR
(2485, X'9B5') Yanlış MaxMsgUzunluk alanı

MQR_MD_ERROR
(2026, X'7EA') İleti tanımlayıcısı geçersiz.

MQR_MODULE_ENTRY_NOT_FOUND
(2497, X'9C1') Belirtilen işlev giriş noktası modülde bulunamadı.

MQR_MODULE_INVALID
(2496, X'9C0') Modül bulundu, ancak yanlış tipte (32 bit/64 bit) ya da geçerli bir dll deęil.

MQR_MODULE_NOT_FOUND
(2495, X'9BF') Modül arama yolunda bulunamadı ya da yükleme yetkisi yok.

MQR_MSG_ID_HATA
(2206, X'89E') İleti tanıtıcısı hatası.

MQR_MSG_SEQ_NUMBER_ERROR (QRC_MSG_SEQ_NUMBER_ERROR)
(2250, X'8CA') İleti sıra numarası geçersiz.

MQR_MSG_TOKEN_HATA
(2331, X'91B') İleti simgesi kullanımı geçersiz.

MQR_NOT_OPEN_FOR_BROWSE
(2036, X'7F4') Kuyruk göz atmak için açık deęil.

MQRC_NOT_OPEN_FOR_INPUT
(2037, X'7F5') Kuyruk giriş için açık değil.

MQRC_OBJECT_CHANGED
(2041, X'7F9') Nesne tanımı açıldıktan sonra değiştirildi.

MQRC_OBJECT_HASARLI
(2101, X'835 ') Nesne zarar gördü.

MQRC_OPERATION_HATA
(2488, X'9B8') API Çağrısı 'nda İşlem Kodu Yanlış

MQRC_OPTIONS_HATA
(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_PAGESET_HATA
(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRC_Q_SILINDI
(2052, X'804 ') Kuyruk silindi.

MQRC_Q_INDEX_TYPE_HATA
(2394, X'95A') Kuyruk yanlış dizin tipine sahip.

MQRC_Q_MGR_AD_HATASI
(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRC_Q_MGR_YOK
(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.

MQRC_Q_MGR QUIESCING
(2161, X'871 ') Kuyruk yöneticisi susturuyor.

MQRC_Q_MGR DURDURULUYOR
(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM
(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_SIGNAL_HENÜZ tamamlanmamış
(2069, X'815 ') Bu tutamaç için bekleyen işaret.

MQRC_STORAGE_KULLANILAMIYOR
(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_SUPPRESSED_BY_EXIT
(2109, X'83D') Çağrı çıkış programı tarafından engellendi.

MQRC_SYNCPOINT_NOT_YOK
(2072, X'818 ') Syncpoint desteği kullanılmıyor.

MQRC_UNEXPECTED_HATASI
(2195, X'893 ') Beklenmeyen hata oluştu.

MQRC_UOW_ENLISTMENT_HATA
(2354, X' 932 ') Genel iş birimine kayıt başarısız oldu.

MQRC_UOW_MIX_NOT_DESTEKLENMIYOR
(2355, X' 933 ') İş birimi çağrılarının karışımı desteklenmez.

MQRC_UOW_YOK
(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.

MQRC_WAIT_INTERVAL_ERROR
(2090, X'82A') MQGMO' daki bekleme aralığı geçersiz.

MQRC_ERROR_gmo_sürümü
(2256, X'8D0') Yanlış MQGMO sürümü sağlandı.

MQRC_ERROR_md_version
(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. Geri çağırma yordamları, çağırdıkları tüm hizmetlerden gelen yanıtları denetlemelidir ve yordam çözülemeyen bir koşul saptarsa, geri çağırma yordamına yinelenen çağrılarını önlemek için bir MQCB MQOP_DEREGISTER komutu vermelidir.
2. IBM MQgüncellemeleri de dahil olmak üzere, bir XA Transaction Manager 'in genel hareketleri yönettiği bir uygulamada zamanuyumsuz tüketim kullanıyorsanız, aşağıdaki ek noktaları göz önünde bulundurmanız gerekir:
 - a. Bir **HConn** için, oluşturulduktan sonra, **xa_open** çağırıldıktan sonra MQCTL (MQOP_START) çağrılmaz. Bunun nedeni, **HConn** 'in bir XA bağlamına bağlanmış olması ve bu nedenle zamanuyumsuz tüketim mekanizması tarafından kullanılan ayrı iş parçacığında ya da iş parçacıklarında erişilememesi olabilir.
 - b. Bu senaryoda MQCTL (MQOP_START) çağrılırsa, çağrı MQRC_ASYNC_XA_CONFLICT (2350) neden koduyla başarısız olur.
 - c. Bir **HConn** için, oluşturulduktan sonra, **xa_open** çağırıldıktan sonra MQCTL (MQOP_START_WAIT) çağrılmalıdır. Bunun nedeni, bu zamanuyumsuz tüketim mekanizmasını başlatma yönteminin, **HConn** için MQCTL çağrısının yapıldığı iş parçacığında daha fazla geri çağırmaya neden olmasıdır. Bu nedenle, **HConn** ile iş parçacığı arasındaki bağlantı kaybolmaz.
3. **z/OS** z/OS üzerinde, İşlem MQOP_START olduğunda:
 - Zamanuyumsuz geri çağırma yordamlarını kullanan programlara z/OS UNIX System Services (z/OS UNIX) programını kullanma yetkisi verilmelidir.
 - Zamanuyumsuz geri çağırma yordamlarını kullanan Dil Ortamı (LE) programları POSIX(ON) LE çalıştırma zamanı seçeneğini kullanmalıdır.
 - Zamanuyumsuz geri çağırma yordamlarını kullanan LE dışı programlar z/OS UNIX pthread_create arabirimini (çağrılabilir hizmet BPX1PTC) kullanmamalıdır.
4. **z/OS** MQCTL, IMS bağdaştırıcısında desteklenmiyor.

Not: CICS içinde MQOP_START desteklenmez. Bunun yerine, MQOP_START_WAIT işlev çağrılarını kullanın.

C çağrısı

```
MQCTL (Hconn, Operation, &ControlOpts, &CompCode, &Reason)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağrısı

```
CALL 'MQCTL' USING HCONN, OPERATION, CTLOPTS, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağırısı

```
call MQCTL(Hconn, Operation, CtlOpts, CompCode, Reason)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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-Kuyruk yöneticisinin bağlantısını kes

MQDISC çağırısı, kuyruk yöneticisi ile uygulama programı arasındaki bağlantıyı kesiyor ve MQCONN ya da MQCONNX çağırısının tersi.

- z/OSüzerinde, zamanuyumsuz ileti tüketimi, olay işleme ya da geri çağırma kullanan tüm uygulamalar, ana denetim iş parçacığı sona ermeden önce bir MQDISC çağırısı yayınlamalıdır. Daha fazla ayrıntı için bkz. [IBM MQ iletilerinin zamanuyumsuz tüketimi](#).
- z/OSüzerinde, CICS uygulamalarının kuyruk yöneticisiyle bağlantısını kesmek için bu çağırısı yapması gerekmez.

Bir CICS uygulaması bu çağırısı yaparsa, aşağıdakilerden birini belirterek daha önce bir MQCONNX çağırısı yapılmadıysa bu çağırısı etkili olmaz:

```
MQCNO_SERIALIZE_CONN_TAG_Q_MGR  
MQCNO_SERIALIZE_CONN_TAG_QSG  
MQCNO_RESTRICT_CONN_TAG_Q_MGR ya da  
MQCNO_RESTRICT_CONN_TAG_QSG
```

seçenekler, bu durumda tüm açık nesne tanıtıcıları kapatılır.

Sözdizimi

MQDISC (*Hconn*, *CompCode*, *Reason*(Neden))

Parametreler

Hconn.

Tip: MQHCONN-giriş/çıkış

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağırısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağrılarını atlayabilir ve *Hconn*için aşağıdaki değeri belirtebilirsiniz:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

Çağrı başarıyla tamamlandığında, kuyruk yöneticisi *Hconn* değerini ortam için geçerli bir tanıtıcı olmayan bir değere ayarlar. Bu değer:

MQHC_UNUSABLE_HCONN

Kullanılamaz bağlantı tanıtıcısı.

z/OS' da *Hconn* , tanımlanmamış bir değere ayarlanır.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdaki kodlardan biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_BACKED_OUT

(2003, X'7D3') Çalışma birimi geri çekildi.

MQRC_CONN_TAG_SERBEST BIRAKILMAMIŞ

(2344, X' 928 ') Bağlantı etiketi serbest bırakılmadı.

MQRC_OUTCOME_PENDING

(2124, X'84C') Kesinleştirme işleminin sonucu beklemede.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_DISC_LOAD_HATA

(2138, X'85A') Bağdaştırıcı bağlantı kesmesi modülü yüklenemiyor.

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_INIT_ERROR (UYGULAMA_HATASI)

(2375, X' 947 ') API çıkışını başlatma başarısız oldu.

MQRC_API_EXIT_TERM_HATA

(2376, X' 948 ') API çıkışı sonlandırılmadı.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_CONNECTION_DURDURULUYOR

(2203, X'89B') Bağlantı kapatılıyor.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_OUTCOME_MIXED

(2123, X'84B') Kesinleştirme ya da geri yükleme işleminin sonucu karışık.

MQRC_PAGESET_HATA

(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRC_Q_MGR_AD_HATASI

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRC_Q_MGR_YOK

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. Bağlantı hala o bağlantı altında açık nesnelere olduğunda bir MQDISC çağrısı verilirse, kuyruk yöneticisi bu nesnelere kapatır ve kapatma seçenekleri MQCO_NONE olarak ayarlanır.
2. Uygulama bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması uygulamanın nasıl sona erdiğine bağlıdır:
 - a. Uygulama, sona erdirmeden önce MQDISC çağrısı yaparsa:
 - Kuyruk yöneticisi eşgüdümlü bir iş birimi için, kuyruk yöneticisi uygulama adına MQCMIT çağrısı yayınlar. Mümkünse iş birimi kesinleştirildi ve kesinleştirilmediyse geri çekildi.
 - Dışarıdan eşgüdümlü bir iş biriminde, iş biriminin durumunda herhangi bir değişiklik yoktur; ancak, kuyruk yöneticisi genellikle, iş birimi koordinatörü tarafından istendiğinde iş biriminin kesinleştirilmesi gerektiğini belirtir.

z/OS, CICS, IMS (toplu iş DL/1 programları dışında) ve RRS uygulamaları bu şekilde olur.
 - b. Uygulama olağan bir şekilde sona ererse, ancak MQDISC çağrısı yayınlanmazsa, yapılan işlem ortama bağlıdır:
 - z/OS sistemlerinde, MQ Java ya da MQ JMS uygulamaları dışında, 2a notunda açıklanan işlemler gerçekleştirilir.
 - Diğer tüm durumlarda, 2c notunda açıklanan işlemler gerçekleşir.

Ortamlar arasındaki farklar nedeniyle, iş birimini kesinleştirmek ya da sona erdirmeden önce geri çıkarmak istediğiniz uygulamaların olduğundan emin olun.
 - c. Uygulama MQDISC çağrısı yayınlanmadan *olağandışı şekilde* sona ererse, iş birimi geriletir.
3. z/OS üzerinde aşağıdaki noktalar geçerlidir:
 - CICS uygulamalarının kuyruk yöneticisiyle bağlantısını kesmek için MQDISC çağrısına gerek yoktur; CICS sisteminin kendisi kuyruk yöneticisine bağlanır ve MQDISC çağrısı bu bağlantıyı etkilemez.
 - CICS, IMS (toplu DL/1 programları dışında) ve RRS uygulamaları, bir dış iş birimi koordinatörü tarafından koordine edilen iş birimlerini kullanır. Sonuç olarak, MQDISC çağrısı, çağrı yayınlandığında var olan iş biriminin durumunu (varsa) etkilemez.

Ancak, *Yapar* MQDISC çağrısı, uygulama tarafından yayınlanan daha önceki bir MQCONNX çağrısıyla bağlantıyla ilişkilendirilmiş *ConnTag* bağlantı etiketinin kullanımının sona erdiğini gösterir. MQDISC çağrısı yayınlandığında bağlantı etiketine gönderme yapan etkin bir iş birimi varsa, çağrı tamamlama kodu MQCC_WARNING ve neden kodu MQRC_CONN_TAG_NOT_ALLOWED ile tamamlanır. Dış iş birimi eşgüdümcüsü iş birimini çözümleyinceye kadar bağlantı etiketi yeniden kullanım için kullanılamaz.

Not: CICSiçinde MQOP_START desteklenmez. Bunun yerine, MQOP_START_WAIT iřlev çağrılarını kullanın.

C çağrısı

```
MQDISC (&Hconn, &CompCode, &Reason);
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
MQHCONN  Hconn;      /* Connection handle */
MQLONG   CompCode;   /* Completion code */
MQLONG   Reason;     /* Reason code qualifying CompCode */
```

COBOL çağrısı

```
CALL 'MQDISC' USING HCONN, COMPCODE, REASON.
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
** 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.
```

PL/I çağrısı

```
call MQDISC (Hconn, CompCode, Reason);
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl CompCode   fixed bin(31); /* Completion code */
dcl Reason     fixed bin(31); /* Reason code qualifying CompCode */
```

System/390 çevirici çağrısı

```
CALL MQDISC,(HCONN,COMPCODE,REASON)
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
HCONN      DS F Connection handle
COMPCODE   DS F Completion code
REASON     DS F Reason code qualifying COMPCODE
```

Visual Basic çağrısı

```
MQDISC Hconn, CompCode, Reason
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
Dim Hconn As Long 'Connection handle'
```

Dim CompCode As Long 'Completion code'
Dim Reason As Long 'Reason code qualifying CompCode'

MQDLTMH-İleti silme tanıtıcısı

MQDLTMH çağrısı bir ileti tanıtıcısını siler ve MQCRTMH çağrısının tersidir.

Sözdizimi

MQDLTMH (*Hconn, Hmsg, DltMsgHOpts, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir.

Değer, **Hmsg** değiştirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eşleşmelidir.

İleti tanıtıcısı MQHC_UNASSOCIATED_HCONN kullanılarak yaratıldıysa, ileti tanıtıcısını silme iş parçacığında geçerli bir bağlantı kurulmalıdır; tersi durumda, çağrı MQRC_CONNECTION_BROKEN ile başarısız olur.

İlt

Tip: MQHMSG-giriş/çıkış

Bu, silinecek ileti tanıtıcısıdır. Değer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

Çağrı başarıyla tamamlandığında, tanıtıcı ortam için geçersiz bir değere ayarlanır. Bu değer:

MQHM_HABLE_HMSG

Kullanılmayan ileti tanıtıcısı.

Aynı ileti tanıtıcısı iletilen başka bir IBM MQ çağrısı devam ediyorsa, ileti tanıtıcısı silinemez.

DltMsgHOpts

Tip: MQDMHO-giriş

Ayrıntılar için bkz. [MQDMHO](#) .

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'089C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CONNECTION_BROKEN

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_DMHO_ERROR

(2462, X'099E') İleti silme tanıtıcısı seçenekleri yapısı geçersiz.

MQRC_HMSG_HATA

(2460, X'099C') İleti tanıtıcısı göstergesi geçersiz.

MQRC_MSG_HANDLE_IN_USE

(2499, X'09C3') İleti tanıtıcısı kullanımda.

MQRC_OPTIONS_HATA

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

C çağrısı

```
MQDLTMH (Hconn, &Hmsg, &DltMsgHOpts, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağrısı

```
CALL 'MQDLTMH' USING HCONN, HMSG, DLTMSGHOPTS, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağrısı

```
call MQDLTMH (Hconn, Hmsg, DltMsgHOpts, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL MQDLTMH, (HCONN, HMSG, DLTMMSGHOPTS, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

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-İleti özelliğini sil

MQDLTMP çağrısı bir özelliği ileti tanıtıcısından siler ve MQSETMP çağrısının tersidir.

Sözdizimi

MQDLTMP (*Hconn, Hmsg, DltPropOpts, Name, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. Değer, **Hmsg** değiştirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eşleşmelidir.

İleti tanıtıcısı MQHC_UNASSOCIATED_HCONN kullanılarak yaratıldıysa, ileti tanıtıcısını silen iş parçacığında geçerli bir bağlantı kurulmalıdır; tersi durumda, çağrı MQRC_CONNECTION_BROKEN ile başarısız olur.

İtt

Tip: MQHMSG-giriş

Bu, silinecek özelliği içeren ileti tanıtıcısıdır. Değer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

DltProp(DltProp) Seçmeler

Tip: MQDMPO-giriş

Ayrıntılar için [MQDMPO](#) veri tipine bakın.

Ad

Tip: MQCHARV-giriş

Silinecek özelliğin adı. Özellik adlarıyla ilgili ek bilgi için [Özellik adları](#) konusuna bakın.

Özellik adında genel arama karakterlerine izin verilmez.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_PROPERTY_NOT_VAR

(2471, X'09A7') Özellik kullanılamıyor.

MQRC_RFH_FORMAT_HATA

(2421, X'0975 ') Özellikleri içeren bir MQRFH2 klasörü ayrıştırılmadı.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'089C') Bağdaştırıcı kullanılamıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'0852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'086D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CONNECTION_BROKEN

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_DMPO_ERROR

(2481, X'09B1') İleti silme özelliği seçenekleri yapısı geçersiz.

MQRC_HMSG_HATA

(2460, X'099C') İleti tanıtıcısı geçersiz.

MQRC_MSG_HANDLE_IN_USE

(2499, X'09C3') İleti tanıtıcısı kullanımda.

MQRC_OPTIONS_HATA

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_NAMETY_NAME_ERROR

(2442, X'098A') Özellik adı geçersiz.

MQRC_SOURCE_CCSID_HATA

(2111, X'083F') Özellik adı kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'0893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

C çağrısı

```
MQDLTMP (Hconn, Hmsg, &DltPropOpts, &Name, &CompCode, &Reason)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

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 */

```

COBOL çağırısı

```
CALL 'MQDLTMP' USING HCONN, HMSG, DLTPROPOPTS, NAME, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

** 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 CMQCHRVV.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON   PIC S9(9) BINARY.

```

PL/I çağırısı

```
call MQDLTMP (Hconn, Hmsg, DltPropOpts, Name, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

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 */

```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQDLTMP, (HCONN, HMSG, DLTPROPOPTS, NAME, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

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-İletiyi al

MQGET çağırısı, MQOPEN çağırısı kullanılarak açılan bir yerel kuyruktan ileti alır.

Sözdizimi

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

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve *Hconn* için belirtilen şu değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

Hobj.

Tip: MQHOBJ-giriş

Bu tanıtıcı, iletinin alınacağı kuyruğu gösterir. *Hobj* değeri önceki bir MQOPEN çağrısı tarafından döndürüldü. Kuyruk aşağıdaki seçeneklerden biriyle ya da daha fazlasıyla açılmış olmalıdır (ayrıntılar için bkz. [“MQOPEN-Nesneyi aç” sayfa 725](#)):

- MQOO_INPUT_SHARED
- MQOO_INPUT_EXCLUSIVE
- MQOO_INPUT_AS_Q_DEF
- MQOO_GÖZ AT

MsgDesc

Tip: MQMD-giriş/çıkış

Bu yapı, gerekli iletinin özniteliklerini ve alınan iletinin özniteliklerini açıklar. Ayrıntılar için bkz. [“MQMD-İleti tanımlayıcı” sayfa 420](#).

BufferLength ileti uzunluğundan azsa, *MsgDesc* kuyruk yöneticisi tarafından doldurulur; **GetMsgOpts** değiştirilmesinde MQGMO_ACCEPT_TRUNCATED_MSG belirtilip belirtilmediğini belirler (bkz. [MQGMO-Options alanı](#)).

Uygulama version-1 MQMD 'yi sağlarsa, döndürülen iletide uygulama ileti verilerinin başına bir MQMDE öneki eklenmiş, ancak yalnızca MQMDE' deki alanlardan birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa. MQMDE içindeki tüm alanların varsayılan değerleri varsa, MQMDE atlanır. MQMD 'deki *Biçim* alanındaki MQFMT_MD_EXTENSION biçim adı, bir MQMDE' nin var olduğunu gösteriyor.

MsgHandle alanında geçerli bir ileti tanıtıcısı belirtildiyse, uygulamanın MQMD yapısı sağlaması gerekmez. Bu alanda hiçbir şey sağlanmazsa, iletinin tanımlayıcısı, ileti tanıtıcılarıyla ilişkili tanımlayıcıdan alınır.

Uygulama MQMD yapısı yerine bir ileti tanıtıcısı sağlarsa ve MQGMO_PROPERTIES_FORCE_MQRFH2değerini belirtirse, çağrı MQRC_MD_ERROR neden koduyla başarısız olur. Uygulama bir MQMD yapısı sağlamazsa ve MQGMO_PROPERTIES_AS_Q_DEF belirtiliyorsa ve **PropertyControl** kuyruk özneliği MQPROP_FORCE_MQRFH2ise, MQRC_MD_ERROR neden koduyla çağrı da başarısız olur.

Eşleştirme seçenekleri belirtilirse ve ileti tanıtıcısıyla ilişkili ileti tanımlayıcısı kullanılıyorsa, eşleştirme için kullanılan giriş alanları ileti tanıtıcısından gelir.

GetMsgSeçmeler

Tip: MQGMO-giriş/çıkış

Ayrıntılar için bkz. [“MQGMO-Get-message seçenekleri” sayfa 368](#).

BufferLength

Tip: MQLONG-input

Bu, *Buffer* alanının bayt cinsinden uzunluğudur. Veri içermeyen iletiler için ya da ileti kuyruktan kaldırılacaksa ve atılan veriler için sıfır belirtin (bu durumda MQGMO_ACCEPT_TRUNCATED_MSG belirtmeniz gerekir).

Not: Kuyruktan okunabilecek en uzun iletinin uzunluğu **MaxMsgLength** kuyruk özniteliği tarafından verilir; bkz. [“Kuyruklara ilişkin öznitelikler” sayfa 826.](#)

Arabellek

Tip: MQBYTEExBufferUzunluk-çıkış

Bu, ileti verilerini içerecek alandır. Arabelleği, iletideki verilerin niteliğine uygun bir sınırdan hizalayın. 4 baytlık hizalama, çoğu ileti için uygundur (IBM MQ üstbilgi yapılarını içeren iletiler de içinde olmak üzere), ancak bazı iletiler daha sıkı hizalama gerektirebilir. Örneğin, 64 bit ikili tamsayı içeren bir ileti 8 baytlık hizalama gerektirebilir.

BufferLength ileti uzunluğundan azsa, iletinin olabildiğince çoğu **Buffer**içine taşınır. **GetMsgOpts** değiştirgesinde MQGMO_ACCEPT_TRUNCATED_MSG belirtilirse bu durum oluşur (ek bilgi için [MQGMO-Options alanına](#) bakın).

Buffer içindeki verilerin karakter kümesi ve kodlaması, **MsgDesc** parametresinde döndürülen *CodedCharSetId* ve *Encoding* alanları tarafından verilir. Bu değerler alıcının gerektirdiği değerlerden farklıysa, alıcının uygulama ileti verilerini gereken karakter kümesine ve kodlamaya dönüştürmesi gerekir. İleti verilerini dönüştürmek için MQGMO_CONVERT seçeneği (gerekliyorsa kullanıcı tarafından yazılmış bir çıkışla) kullanılabilir; bu seçeneğin ayrıntıları için bkz. [“MQGMO-Get-message seçenekleri” sayfa 368 .](#)

Not: MQGET çağrısındaki diğer tüm değiştirgeler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında (**CodedCharSetId** kuyruk yöneticisi özniteliği ve MQENC_NATIVE ile verilir) bulunur.

Arama başarısız olursa, arabelleğin içeriği değişmiş olabilir.

C programlama dilinde, parametre bir gösterge-to-void olarak bildirilir: herhangi bir veri tipinin adresi parametre olarak belirtilebilir.

BufferLength parametresi sıfırsa, *Buffer* değiştirgesine başvurulamaz; bu durumda, C ya da System/390 çevirici dilinde yazılmış programlar tarafından geçirilen parametre adresi boş olabilir.

DataLength

Tip: MQLONG-output

Bu, *iletideuygulama* verilerinin bayt cinsinden uzunluğudur. Bu değer *BufferLength*değerinden büyükse, **Buffer** değiştirgesinde yalnızca *BufferLength* byte döndürülür (ileti kesilir). Değer sıfırsa, ileti uygulama verisi içermez.

BufferLength ileti uzunluğundan azsa, kuyruk yöneticisi *DataLength* işlemini tamamlar ve **GetMsgOpts** değiştirgesinde MQGMO_ACCEPT_TRUNCATED_MSG belirtilip belirtilmediğini belirtir (ek bilgi için [MQGMO-Options alanı](#) konusuna bakın). Bu, uygulamanın ileti verilerini barındırmak için gereken arabellek büyüklüğünü saptamasını sağlar ve daha sonra, çağrıyı uygun büyüklükte bir arabellekle yeniden yürütür.

Ancak, MQGMO_CONVERT seçeneği belirtilirse ve dönüştürülen ileti verileri *Buffer*içine sığmayacak kadar uzunsa, *DataLength* için döndürülen değer:

- Kuyruk yöneticisi tanımlı biçimler için *dönüştürülmemiş* verilerin uzunluğu.

Bu durumda, verilerin türü dönüştürme sırasında genişletilmesine neden olursa, uygulamanın *DataLength*için kuyruk yöneticisi tarafından döndürülen değerden daha büyük bir arabellek ayırması gerekir.

- Uygulama tanımlı biçimler için, veri dönüştürme çıkışı tarafından döndürülen değer.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Listelenen neden kodları, kuyruk yöneticisinin **Reason** parametresi için döndürebileceği kodlardır. Uygulama MQGMO_CONVERT seçeneğini belirtiyorsa ve ileti verilerinin bazılarını ya da tümünü dönüştürmek için kullanıcı tarafından yazılan bir çıkış çağırılırsa, **Reason** parametresi için hangi değer döndürüleceğine çıkış karar verir. Sonuç olarak, belgelenen değerlerden başka değerler de olabilir.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_CONVERTED_MSG_TOO_BIG

(2120, X'848 ') Dönüştürülen veriler arabellek için çok büyük.

MQRC_CONVERTED_STRING_TOO_BIG

(2190, X'88E') Dönüştürülen dizgi alan için çok büyük.

MQRC_DBCS_HATA

(2150, X'866 ') DBCS dizgisi geçersiz.

MQRC_FORMAT_HATASI

(2110, X'83E') İleti biçimi geçersiz.

MQRC_INCOMPLETE_GROUP

(2241, X'8C1') İleti grubu tamamlanmadı.

MQRC_INCOMPLETE_MSG

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

MQRC_INCONSISTENT_CCSID ' LER

(2243, X'8C3') İleti bölümlerinin CCSID' leri farklı.

MQRC_INENCOT_ENCODINGS

(2244, X'8C4') İleti bölümlerinin farklı kodlamaları var.

MQRC TUTARSIZ UOW

(2245, X'8C5') İş birimi belirtimi tutarsız.

MQRC_MSG_TOKEN_HATA

(2331, X'91B') İleti simgesi kullanımı geçersiz.

MQRC_NO_MSG_LOCKED

(2209, X'8A1') Kilitli ileti yok.

MQRC_NOT_CONVERTED

(2119, X'847 ') İleti verileri dönüştürülmedi.

MQRC_OPTIONS_CHANGED

(nnnn, X'xxx ') Tutarlılık için gerekli olan seçenekler değiştirildi.

MQRC_PARTIALLY_DÖNÜŞTÜRÜLDÜ

(2272, X'8E0') İleti verileri kısmen dönüştürüldü.

MQRC_SIGNAL_REQUEST_ACCEPTED

(2070, X'816 ') İleti döndürülmedi (ancak sinyal isteği kabul edildi).

MQRC_SOURCE_BUFFER_ERROR (QRC_SOURCE_BUFFER_ERROR)

(2145, X'861 ') Kaynak arabellek değiştirgesi geçersiz.

MQRC_SOURCE_CCSID_HATA

(2111, X'83F') Kaynak kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_SOURCE_DECIMAL_ENC_HATA

(2113, X'841 ') İletideki paketli ondalık kodlama tanınmıyor.

MQRC_SOURCE_FLOAT_ENC_HATA

(2114, X'842 ') İletide kayan nokta kodlaması tanınmıyor.

MQRC_SOURCE_INTEGER_ENC_HATA

(2112, X'840 ') Kaynak tamsayı kodlaması tanınmıyor.

MQRC_SOURCE_LENGTH_ERROR

(2143, X'85F') Kaynak uzunluğu parametresi geçersiz.

MQRC_TARGET_BUFFER_HATA

(2146, X'862 ') Hedef arabellek değiştirgesi geçersiz.

MQRC_TARGET_CCSID_HATA

(2115, X'843 ') Hedef kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_TARGET_DECIMAL_ENC_HATA

(2117, X'845 ') Alıcı tarafından belirtilen Paketlenmiş ondalık kodlama tanınmıyor.

MQRC_TARGET_FLOAT_ENC_HATA

(2118, X'846 ') Alıcı tarafından belirtilen kayan nokta kodlaması tanınmıyor.

MQRC_TARGET_INTEGER_ENC_HATA

(2116, X'844 ') Hedef tamsayı kodlaması tanınmıyor.

MQRC_TRUNCATED_MSG_ACCEPTED

(2079, X'81F') İzlenen ileti döndürüldü (işleme tamamlandı).

MQRC_TRUNCATED_MSG_FAILED

(2080, X'820 ') İzlenen ileti döndürüldü (işleme tamamlanmadı).

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_CONV_LOAD_ERROR (UYGULAMA_HATASI)

(2133, X'855 ') Veri dönüştürme hizmetleri modülleri yüklenemiyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

(2183, X'887 ') API çıkışı yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BACKED_OUT

(2003, X'7D3') Çalışma birimi geri çekildi.

MQRC_BUFFER_ERROR

(2004, X'7D4') Arabellek değiştirgesi geçersiz.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağırısı.

MQRC_CF_NOT_YOK

(2345, X' 929 ') Bağlama olanağı kullanılmıyor.

MQRC_CF_STRUC_FAILED

(2373, X' 945 ') Bağlama tesisi yapısı başarısız oldu.

MQRC_CF_STRUC_IN_USE
(2346, X'92A') Coupling-facility yapısı kullanılıyor.

MQRC_CF_STRUC_LIST_HDR_IN_USE
(2347, X'92B') Coupling-facility yapısı liste üstbilgisi kullanımda.

MQRC_CICS_WAIT_FAILED
(2140, X'85C') Bekleme isteği CICStarafından reddedildi.

MQRC_CONNECTION_BROKEN
(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_CONNECTION_NOT_YETKILI
(2217, X'8A9') Bağlantı için yetkili değil.

MQRC_CONNECTION_QUIESCING
(2202, X'89A') Bağlantı susturuyor.

MQRC_CONNECTION_DURDURULUYOR
(2203, X'89B') Bağlantı kapatılıyor.

MQRC_CORREL_ID_HATA
(2207, X'89F') İlinti-tanıtıcı hatası.

MQRC_DATA_LENGTH_HATA
(2010, X'7DA') Veri uzunluğu parametresi geçersiz.

MQRC_DB2_NOT_AVAILABLE
(2342, X' 926 ') Db2 altsistemi kullanılamıyor.

MQRC_GET_INHIBE
(2016, X'7E0') Kuyruk için engellendi.

MQRC_GLOBAL_UOW_CONFLICT
(2351, X'92F') Genel iş birimleri çakışması.

MQRC_GMO_HATA
(2186, X'88A') Get-message seçenekleri yapısı geçersiz.

MQRC_HANDLE_IN_USE_FOR_UOW
(2353, X' 931 ') Genel iş birimi için kullanımda.

MQRC_HCONN_HATA
(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_HOBJ_HATA
(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

MQRC_TUTARSIZ_GÖZAT
(2259, X'8D3') Tutarsız göz atma belirtimi.

MQRC_TUTARSIZ_UOW
(2245, X'8C5') İş birimi belirtimi tutarsız.

MQRC_INVALID_MSG_UNDER_CURSOR
(2246, X'8C6') İmleç altındaki ileti alma için geçerli değil.

MQRC_LOCAL_UOW_CONFLICT
(2352, X' 930 ') Genel iş birimi yerel iş birimiyle çakışıyor.

MQRC_MATCH_OPTIONS_HATA
(2247, X'8C7') Eşleştirme seçenekleri geçerli değil.

MQRC_MD_ERROR
(2026, X'7EA') İleti tanımlayıcısı geçersiz.

MQRC_MSG_ID_HATA
(2206, X'89E') İleti tanıtıcısı hatası.

MQRC_MSG_SEQ_NUMBER_ERROR (QRC_MSG_SEQ_NUMBER_ERROR)
(2250, X'8CA') İleti sıra numarası geçersiz.

MQRC_MSG_TOKEN_HATA
(2331, X'91B') İleti simgesi kullanımı geçersiz.

MQRN_NO_MSG_AVAILABLE
(2033, X'7F1') İleti yok.

MQRN_NO_MSG_UNDER_CURSOR
(2034, X'7F2') Göz at imleci iletide konumlandırılmadı.

MQRN_NOT_OPEN_FOR_BROWSE
(2036, X'7F4') Kuyruk göz atmak için açık değil.

MQRN_NOT_OPEN_FOR_INPUT
(2037, X'7F5') Kuyruk giriş için açık değil.

MQRN_OBJECT_CHANGED
(2041, X'7F9') Nesne tanımı açıldıktan sonra değiştirildi.

MQRN_OBJECT_HASARLI
(2101, X'835 ') Nesne zarar gördü.

MQRN_OPTIONS_HATA
(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRN_PAGESET_HATA
(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRN_Q_SILINDI
(2052, X'804 ') Kuyruk silindi.

MQRN_Q_INDEX_TYPE_HATA
(2394, X'95A') Kuyruk yanlış dizin tipine sahip.

MQRN_Q_MGR_AD_HATASI
(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRN_Q_MGR_YOK
(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.

MQRN_Q_MGR_QUIESCING
(2161, X'871 ') Kuyruk yöneticisi susturuyor.

MQRN_Q_MGR_DURDURULUYOR
(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRN_RESOURCE_PROBLEM
(2102, X'836 ') Sistem kaynakları yetersiz.

MQRN_SECOND_MARK_NOT_ALLOWED
(2062, X'80E') Bir ileti zaten işaretli.

MQRN_SIGNAL_HENÜZ tamamlanmamış
(2069, X'815 ') Bu tutamaç için bekleyen işaret.

MQRN_SIGNAL1_ERROR
(2099, X'833 ') Sinyal alanı geçersiz.

MQRN_STORAGE_MEDIUM_FULL
(2192, X'890 ') Dış depolama ortamı dolu.

MQRN_STORAGE_KULLANILAMIYOR
(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRN_SUPPRESSED_BY_EXIT
(2109, X'83D') Çağrı çıkış programı tarafından engellendi.

MQRN_SYNCPOINT_LIMIT_ULAŞILDI
(2024, X'7E8') Yürürlükteki iş birimi içinde başka ileti işlenmez.

MQRN_SYNCPOINT_NOT_YOK
(2072, X'818 ') eşitleme noktası desteği kullanılamıyor.

MQRN_UNEXPECTED_HATASI
(2195, X'893 ') Beklenmeyen hata oluştu.

MQRN_UOW_ENLISTMENT_HATA
(2354, X' 932 ') Genel iş birimine kayıt başarısız oldu.

MQRC_UOW_MIX_NOT_DESTEKLENMIYOR

(2355, X'933') İş birimi çağrılarının karışımı desteklenmez.

MQRC_UOW_YOK

(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.

MQRC_WAIT_INTERVAL_ERROR

(2090, X'82A') MQGMO' daki bekleme aralığı geçersiz.

MQRC_ERROR_gmo_sürümü

(2256, X'8D0') Yanlış MQGMO sürümü sağlandı.

MQRC_ERROR_md_version

(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. Alınan ileti olağan durumda kuyruktan silinir. Bu silme, MQGET çağrısının bir parçası olarak ya da bir eşitleme noktasının parçası olarak gerçekleşebilir.

Göz atma seçenekleri şunlardır: MQGMO_BROWSE_FIRST, MQGMO_BROWSE_NEXT ve MQGMO_BROWSE_MSG_UNDER_CURSOR.

2. MQGMO_LOCK seçeneği göz atma seçeneklerinden biriyle belirtilirse, göz atılan ileti yalnızca bu tanıtıcı tarafından görülebilmesi için kilitletir.

MQGMO_UNLOCK seçeneği belirtilirse, önceden kilitletlenmiş bir iletinin kilidi açılır. Bu durumda ileti alınmaz ve **MsgDesc**, **BufferLength**, **Buffer** ve **DataLength** parametreleri denetlenmez ya da değiştirilmez.

3. MQGET çağrısı yayınlayan uygulamalar için, uygulama olağandışı bir şekilde sonlandırılırsa ya da çağrı işlenirken bağlantı kesilirse, alınan ileti kaybolabilir. Bu sorun, uygulama adına MQGET çağrısının verildiği kuyruk yöneticisiyle aynı altyapıda çalışan vekil, ileti kuyruktan kaldırıldıktan sonra vekil iletiyi uygulamaya döndürmek üzere oluncaya kadar uygulama kaybını saptayamamasından kaynaklanır. Bu sorun, hem kalıcı iletiler hem de kalıcı olmayan iletiler için ortaya çıkabilir.

İletileri bu şekilde kaybetme riskini ortadan kaldırmak için her zaman iş birimi içindeki iletileri alın. Yani, MQGET çağrısında MQGMO_SYNCPOINT seçeneğini belirterek ve ileti işleme tamamlandığında iş birimini kesinleştirmek ya da geri almak için MQCMIT ya da MQBACK çağrılarını kullanarak. MQGMO_SYNCPOINT belirtilirse ve istemci olağandışı bir şekilde sonlandırılırsa ya da bağlantı kesilirse, vekil, kuyruk yöneticisindeki iş birimini geri alır ve ileti kuyrukta yeniden kullanılır. Eşitleme noktalarıyla ilgili daha fazla bilgi için [IBM MQ uygulamalarda Syncpoint ile ilgili önemli noktalar](#) başlıklı konuya bakın.

Bu durum, kuyruk yöneticisiyle aynı altyapıda çalışan uygulamaların yanı sıra IBM MQ istemcileriyle de ortaya çıkabilir.

4. Bir uygulama belirli bir iletiye bir dizi ileti koyarsakuyruğun tek bir iş birimi içinde olması ve daha sonra bu iş birimini başarıyla kesinleştirmesi, iletilerin aşağıdaki gibi alınabilmesine neden olur:

- Kuyruk *paylaşılmayan bir kuyruksa* (yerel kuyruk), iş birimindeki tüm iletiler aynı anda kullanılabilir olur.
- Kuyruk bir *paylaşılan kuyruksa*, iş birimi içindeki iletiler yerleştirildikleri sırayla kullanılabilir, ancak tümü aynı anda kullanılamaz. Sistem ağır yüklendiğinde, iş birimindeki ilk iletinin başarıyla alınması mümkündür; ancak, iş birimindeki ikinci ya da sonraki iletinin MQGET çağrısı MQRC_NO_MSG_AVAILABLE ile başarısız olur. Bu sorun ortaya çıkarsa, uygulamanın kısa bir süre beklemesi ve işlemi yeniden denemesi gerekir.

5. Bir uygulama ileti gruplarını kullanmadan aynı kuyruğa ileti dizisi koyarsa, belirli koşullar yerine getirilirse, bu iletilerin sırası korunur. Ayrıntılar için bkz. [MQPUT kullanım notları](#). Koşullar karşılanıyorsa, iletiler alıcı uygulamaya gönderildikleri sırayla sunulur. Bu durumda:

- Kuyruktan ileti alan tek bir günlük nesnesi var.

Kuyruktan ileti alan iki ya da daha fazla uygulama varsa, bir sıraya ait iletileri tanımlamak için kullanılacak mekanizmayı gönderen ile aynı fikirde olmaları gerekir. Örneğin, gönderen iletilerdeki tüm CorrelId alanlarını bir sırayla, bu ileti sırası için benzersiz olan bir değere ayarlayabilir.

- Alıcı, örneğin belirli bir MsgId ya da CorrelId belirterek, alma sırasını kasıtlı olarak değiştirmez.

Gönderen uygulama iletileri ileti grubu olarak yerleştirirse, alan uygulama MQGET çağrısında MQGMO_LOGICAL_ORDER seçeneğini belirtiyorsa, iletiler alan uygulamaya doğru sırayla sunulur. İleti gruplarıyla ilgili ek bilgi için aşağıdaki konuya bakın:

- [MQMD- MsgFlags alanı](#)
- [MQPMO_LOGICAL_ORDER](#)
- [MQGMO_LOGICAL_ORDER](#)

Kullanıcı, eşitleme noktası altındaki bir gruptaki iletileri alıyorsa, işlemi bitirmeyi denemeden önce tüm grubun işlendiğinden emin olmalıdır.

6. Uygulamalar, **MsgDesc** değiştirgesinin Feedback alanında MQFB_QUIT geribildirim kodu için test etmeli ve bu değeri bulurlarsa sona ermelidir. Ek bilgi için [MQMD-Feedback field](#) (MQMD-Geribildirim alanı) konusuna bakın.
7. Hobj ile tanımlanan kuyruk MQOO_SAVE_ALL_CONTEXT seçeneğiyle açıldıysa ve MQGET çağrısındaki tamamlanma kodu MQCC_OK ya da MQCC_WARNING ise, Hobj kuyruk tanıtıcısıyla ilişkili bağlam, alınan iletinin bağlamına ayarlanır (MQGMO_BROWSE_FIRST, MQGMO_BROWSE_NEXT ya da MQGMO_BROWSE_MSG_UNDER_CURSOR seçeneği ayarlanmadıysa).

MQPMO_PASS_IDENTITY_CONTEXT ya da MQPMO_PASS_ALL_CONTEXT seçeneklerini belirterek, saklanmış bağlamı sonraki bir MQPUT ya da MQPUT1 çağrısında kullanabilirsiniz. Bu, alınan iletinin bağlamının tamamen ya da kısmen başka bir iletiye aktarılmasını sağlar (örneğin, ileti başka bir kuyruğa iletilendiğinde). İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

8. **GetMsgOpts** değiştirgesine MQGMO_CONVERT seçeneğini ekliyorsanız, veriler **Buffer** değiştirgesine yerleştirilmeden önce, uygulama ileti verileri alan uygulama tarafından istenen gösterimine dönüştürülür:
 - İletideki denetim bilgilerindeki Format alanı, uygulama verilerinin yapısını tanımlar ve iletidaki denetim bilgilerindeki CodedCharSetId ve Encoding alanları, karakter kümesi tanıtıcısını ve kodlamasını belirtir.
 - MQGET çağrısını yayınlayan uygulama, **MsgDesc** değiştirgesindeki CodedCharSetId ve Encoding alanlarında, uygulama ileti verilerinin dönüştürülecek karakter kümesi tanıtıcısını ve kodlamasını belirtir.

İleti verilerinin dönüştürülmesi gerektiğinde, dönüştürme, iletidaki denetim bilgisindeki Format alanının değerine bağlı olarak, kuyruk yöneticisi tarafından ya da kullanıcı tarafından yazılan bir çıkış tarafından gerçekleştirilir:

- Aşağıdaki biçim adları, kuyruk yöneticisi tarafından dönüştürülen biçimlerdir; bu biçimlere "yerleşik" biçimler denir:
 - MQFMT_ADMIN
 - MQFMT_CICS (yalnızca z/OS)
 - MQFMT_COMMAND_1
 - MQFMT_COMMAND_2
 - MQFMT_DEAD_LETTER_HEADER
 - MQFMT_DIST_HEADER
 - MQFMT_EVENT sürüm 1
 - MQFMT_EVENT sürüm 2 (yalnızca 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 (yalnızca z/OS)
 - MQFMT_XMIT_Q_HEADER
- MQFMT_NONE biçim adı, iletideki verilerin niteliğini tanımsız olarak gösteren özel bir değerdir. Sonuç olarak, kuyruk yöneticisi, ileti kuyruktan alındığında dönüştürme girişiminde bulunmaz.

Not: MQGET çağrısında MQFMT_NONE biçim adına sahip bir ileti için MQGMO_CONVERT belirtilirse ve iletinin karakter kümesi ya da kodlaması **MsgDesc** değıştirgesinde belirtilenden farklıysa, ileti **Buffer** değıştirgesinde döndürölür (başka hata olmadığı varsayılarak), ancak çağrı tamamlama kodu MQCC_WARNING ve neden kodu MQRC_FORMAT_ERROR ile tamamlanır.

İleti verilerinin niteliği, dönüştürme gerektirmediği anlamına geldiğinde ya da gönderen ve alan uygulamalar, ileti verilerinin gönderileceği form arasında anlaşmaya vardığında MQFMT_NONE özelliğini kullanabilirsiniz.

- Diğer tüm biçim adları, iletiyi dönüştürme için kullanıcı tarafından yazılan bir çıkışa geçirir. Çıkış, ortama özgü eklemeler dışında, biçimle aynı ada sahiptir. Kullanıcı tarafından belirtilen biçim adları IBM MQharfleriyle başlamamalıdır.

Veri dönüştürme çıkışının ayrıntıları için bkz. “Veri dönüştürme çıkışı” sayfa 895 .

İletideki kullanıcı verileri, desteklenen karakter kümeleri ve kodlamalar arasında dönüştürülebilir. Ancak, ileti bir ya da daha çok IBM MQ üstbilgi yapısı içeriyorsa, iletinin kuyruk adlarında geçerli olan herhangi bir karakter için çift baytlık ya da çok baytlık karakter içeren bir karakter kümesine dönüştürülemediğini unutmayın. Neden kodu MQRC_SOURCE_CCSD_ERROR ya da MQRC_TARGET_CCSD_ERROR, denirse sonuçlanır ve ileti dönüştürölmeden döndürölür. Unicode karakter kümesi UTF-16 , böyle bir karakter kümesinin bir örneğidir.

MQGET döndürölüşünde, aşağıdaki neden kodu iletinin başarıyla dönüştürölüğünü gösterir:

- MQRC_NONE

Aşağıdaki neden kodu, iletinin başarıyla dönüştürölmüş olabileceğini gösterir; uygulamanın öğrenmek için **MsgDesc** değıştirgesindeki CodedCharSetId ve Encoding alanlarını denetlemesi gerekir:

- MQRC_TRUNCATED_MSG_ACCEPTED

Diğer tüm neden kodları iletinin dönüştürölmediğini gösterir.

Not: Bu neden kodunun yorumlanması, yalnızca çıkış “Veri dönüştürme çıkışı” sayfa 895içinde açıklanan işleme yönergelerine uyuyorsa, kullanıcı tarafından yazılan bir çıkış tarafından gerçekleştirilen dönüştürmeler için geçerlidir.

9. İletileri almak için nesne yönelimli arabirimi kullanırken, MQGET çağrısına ilişkin ileti verilerini tutmak için bir arabellek belirtmemeyi seçebilirsiniz. Alma iletisi arabelleğinin büyüklüğünü sınırlamadan nesne yönelimli bir uygulama kullanarak bir ileti aldığınızda, uygulama MQRC_CONVERTED_MSG_TOO_BIG ile başarısız olmaz ve dönüştürölülen iletiyi alır. Bu, aşağıdaki ortamlar için geçerlidir:

- Tam olarak yönetilen uygulamalar da dahil olmak üzere .NET
- C++
- Java (IBM MQ classes for Java)

Not: Tüm istemciler için sharingConversations değeri sıfırsa ve arabellek dönüştürölülen iletiyi almak için çok küçükse, dönüştürölmemiş ileti döndürölür; neden kodu MQRC_CONVERTED_MSG_TOO_BIG. sharingConversations hakkında daha fazla bilgi için İstemci uygulamasında paylaşım sohbetleri kullanmabaşlıklı konuya bakın.

10. Yerleşik biçimler için kuyruk yöneticisi, MQGMO_CONVERT seçeneği belirtildiğinde iletideki karakter dizgilerinin varsayılan dönüşümünü gerçekleştirebilir. Varsayılan dönüştürme, kuyruk yöneticisinin dizgi verilerini dönüştürürken gerçek karakter kümesine yaklaşan, kuruluş tarafından belirtilen bir varsayılan karakter kümesi kullanmasını sağlar. Sonuç olarak, MQGET çağrısı MQCC_WARNING ve neden kodu MQRC_SOURCE_CCSID_ERROR ya da MQRC_TARGET_CCSID_ERROR ile tamamlanmak yerine MQCC_OK tamamlanma koduyla başarılı olabilir.

Not: Dizgi verilerini dönüştürmek için yaklaşık bir karakter kümesi kullanılmasının sonucu, bazı karakterlerin yanlış dönüştürülebilmesinden kaynaklanır. Bunu önlemek için, dizgide hem gerçek karakter kümesi hem de varsayılan karakter kümesi için ortak olan karakterleri kullanın.

Varsayılan dönüştürme hem uygulama iletisi verileri hem de MQMD ve MQMDE yapılarındaki karakter alanları için geçerlidir:

- Uygulama iletisi verilerinin varsayılan dönüşümü yalnızca aşağıdaki deyimlerin tümü true olduğunda gerçekleşir:
 - Uygulama MQGMO_CONVERT belirtiyor.
 - İleti, desteklenmeyen bir karakter kümesinden ya da karakter kümesine dönüştürülmesi gereken veriler içeriyor.
 - Kuyruk yöneticisi kurulduğunda ya da yeniden başlatıldığında varsayılan dönüştürme etkinleştirildi.
- Kuyruk yöneticisi için varsayılan dönüştürme etkinleştirildiyse, MQMD ve MQMDE yapılarındaki karakter alanlarının varsayılan dönüşümü gerektiği gibi gerçekleşir. MQGET çağrısında uygulama tarafından MQGMO_CONVERT seçeneği belirtilmese de dönüştürme gerçekleştirilir.

11. Visual Basic programlama dili için aYa ş daki noktalar kullanılır:

- **Buffer** değiştirgesinin büyüklüğü **BufferLength** değiştirgesiyle belirtilen uzunluktan azsa, çağrı başarısız olur; neden kodu MQRC_STORAGE_NOT_ALLOWED.
- **Buffer** değiştirgesi Stringtipinde olarak bildirildi. Kuyruktan alınacak veriler Stringtipinde değilse, MQGET yerine MQGETAny çağrısı.

MQGETAny çağrısının değiştirgeleri MQGET çağrısıyla aynıdır; ancak, **Buffer** değiştirgesi Anytipinde olarak bildirilir ve her tip verinin alınmasına izin verir. Ancak bu, Buffer ' in en az BufferLength bayt boyutunda olduğundan emin olmak için denetlenememesi anlamına gelir.

12. Önden okuma etkinleştirildiğinde tüm MQGET seçenekleri desteklenmez. Aşağıdaki çizelge, hangi seçeneklere izin verildiğini ve bunların MQGET çağrıları arasında değiştirilip değiştirilemeyeceğini gösterir.

Çizelge 548. Önden okuma etkinleştirildiğinde MQGET seçeneklerine izin verilir			
	Önden okuma etkinleştirildiğinde izin verilir ve MQGET çağrıları arasında değiştirilebilir	Önden okuma etkinleştirildiğinde izin verilir, ancak MQGET çağrıları arasında değiştirilemez ^a	Önden okuma etkinleştirildiğinde izin verilmeyen MQGET seçenekleri ^b
MQGET MD değerleri	MsgId ^c CorrelId ^c	Kodlama CodedCharSetId	
MQGET MQGMO seçenekleri	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF QUIESCING MQGMO_BROWSE_FIRST ^g MQGMO_BROWSE_NEXT ^g MQGMO_BROWSE_MESSAGE _UNDER_CURSOR ^d	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_VAR MQGMO_ALL_SEGMENTS_VAR MQGMO_MARK_BROWSE_HANDLE 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 _KAPATILMA MQGMO_MSG_ALTINDA _CURSOR ^e MQGMO_KILIK MQGMO_UNLOCK
MQGMO değerleri		MsgHandle	

- a. MQGET çağrıları arasında bu seçenekler değiştirilirse, bir MQRC_OPTIONS_CHANGED neden kodu döndürülür.
- b. Bu seçenekler ilk MQGET çağrısında belirtilirse, önden okuma geçersiz kılınır. Sonraki bir MQGET çağrısında bu seçenekler belirtilirse, MQRC_OPTIONS_ERROR neden kodu döndürülür.

- c. İstemci uygulamalarının, MsgId ve CorrelId değerleri önceki değerlerle MQGET çağrılarını iletileri arasında değiştirilirse, istemciye önceden gönderilmiş olabileceğini ve tüketilmeye (ya da otomatik olarak temizleninceye) kadar istemcide önden okuma arabelleğinde kalacağını bilmeleri gerekir.
 - d. İlk MQGET çağrısı, önden okuma etkinleştirildiğinde iletilere göz atılacağını ya da kuyruktan ileti alınacağını belirler. Uygulama göz atma ve alma birleşimi kullanmayı denerse, MQRC_OPTIONS_CHANGED neden kodu döndürülür.
 - e. MQGMO_MSG_UNDER_CURSOR, önden okuma ile olanaklı değil. Önden okuma etkinleştirildiğinde iletilere göz atılabilir ya da iletiler alınabilir, ancak her ikisinin birleşimi olamaz.
13. Uygulamalar, yalnızca bu iletiler get ile aynı yerel iş birimine konursa, kesinleştirilmemiş iletileri yıkıcı bir şekilde alabilir. Uygulamalar, kesinleştirilmemiş iletileri yıkıcı olmadan alamaz.
14. Göz atma imleci altındaki iletiler bir iş biriminde alınabilir. Bu şekilde kesinleştirilmemiş bir ileti alınamaz.

C çağrısı

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

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağrısı

```
CALL 'MQGET' USING HCONN, HOBJ, MSGDESC, GETMSGOPTS, BUFFERLENGTH,
  BUFFER, DATALENGTH, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağırısı

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

Değiştirgeleri aşağıdaki gibi bildirin:

```
dcl Hconn          fixed bin(31); /* Connection handle */
dcl Hobj           fixed bin(31); /* Object handle */
dcl MsgDesc        like MQMD;    /* Message descriptor */
dcl GetMsgOpts     like MQGMO;   /* Options that control the action of
                                MQGET */
dcl BufferLength    fixed bin(31); /* Length in bytes of the Buffer
                                area */
dcl Buffer          char(n);      /* Area to contain the message data */
dcl DataLength     fixed bin(31); /* Length of the message */
dcl CompCode       fixed bin(31); /* Completion code */
dcl Reason         fixed bin(31); /* Reason code qualifying CompCode */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQGET, (HCONN, HOBJ, MSGDESC, GETMSGOPTS, BUFFERLENGTH,
            BUFFER, DATALENGTH, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

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

Visual Basic çağırısı

```
MQGET Hconn, Hobj, MsgDesc, GetMsgOpts, BufferLength, Buffer,
DataLength, CompCode, Reason
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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-Nesne özniteliklerini sorgu

MQINQ çağırısı, bir nesnenin özniteliklerini içeren bir dizi tamsayı ve karakter dizisi döndürür.

Aşağıdaki nesne tipleri geçerlidir:

- Kuyruk yöneticisi
- Kuyruk
- Ad listesi

- Süreç tanımlaması

Sözdizimi

MQINQ (*Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs, CharAttrLength, CharAttrs, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN -giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri, önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve *Hconn*'in aşağıdaki değer belirtilir:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

Hobj.

Tip: MQHOBJ -giriş

Bu tanıtıcı, gerekli öznitelikleri olan nesneyi (herhangi bir tipte) gösterir. Tanıtıcı değerinin, MQOO_INQUIRE seçeneğini belirten önceki bir MQOPEN çağrısı tarafından döndürülmesi gerekir.

SelectorCount

Tip: MQLONG -giriş

Bu, *Selectors* dizisinde sağlanan seçicilerin sayısıdır. Döndürülecek özniteliklerin sayısıdır. Sıfır geçerli bir değerdir. İzin verilen üst sınır 256 'dir.

Seçiciler

Tip: MQLONG x *SelectorCount* -giriş

Bu bir **SelectorCount** öznitelik seçicileri dizisidir; her seçici, gerekli bir değere sahip bir özniteliği (tamsayı ya da karakter) tanımlar.

Her seçici, *Hobj* ' in temsil ettiği nesne tipi için geçerli olmalıdır; tersi durumda, çağrı tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_SELECTOR_ERROR ile başarısız olur.

Kuyrukların özel durumunda:

- Seçici herhangi bir tipteki kuyruklar için geçerli değilse, çağrı tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_SELECTOR_ERROR ile başarısız olur.
- Seçici yalnızca nesne tipi dışındaki tiplerdeki kuyruklar için geçerliyse, çağrı tamamlanma kodu MQCC_WARNING ve neden kodu MQRC_SELECTOR_NOT_FOR_TYPE ile başarılı olur.
- Sorulmakta olan kuyruk bir küme kuyruğuysa, geçerli seçiciler kuyruğun nasıl çözümlendiğine bağlıdır; daha fazla ayrıntı için bkz. "[Kullanım notları](#)" sayfa 712 .

Seçicileri herhangi bir sırada belirtebilirsiniz. Tamsayı öznitelik seçicilere (MQIA_* seçiciler) karşılık gelen öznitelik değerleri, *IntAttrs* içinde bu seçicilerin *Selectors* içinde gerçekleştiği sırayla döndürülür. Karakter özniteliği seçicilere (MQCA_* seçicileri) karşılık gelen öznitelik değerleri, *CharAttrs* içinde bu seçicilerin olduğu sırada döndürülür. MQIA_* seçicileri MQCA_* seçicileriyle serpiştirilebilirler; yalnızca her tip içindeki görelî sıra önemlidir.

Not:

1. Tamsayı ve karakter özniteliği seçiciler iki farklı aralık içinde ayrılır; MQIA_* seçiciler MQIA_FIRST - MQIA_LAST aralığında ve MQCA_* seçiciler MQCA_FIRST - MQCA_LAST aralığında bulunur.

Her aralık için, MQIA_LAST_USED ve MQCA_LAST_USED değişmezleri, kuyruk yöneticisinin kabul ettiği en yüksek değeri tanımlar.

2. Tüm MQIA_* seçicileri önce gerçekleşirse, *Selectors* ve *IntAttrs* dizilerinde karşılık gelen öğeleri adreslemek için aynı öğe numaraları kullanılabilir.

3. **SelectorCount** parametresi sıfırsa, *Selectors* parametresine başvuruda bulunulmaz. Bu durumda, C ya da S/390 çevirici dilinde yazılan programlar tarafından geçirilen parametre adresi boş olabilir.

Sorulabilecek öznitelikler aşağıdaki çizelgelerde listelenir. MQCA_* seçicileri için, *CharAttr*s içinde elde edilen dizginin uzunluğunu bayt cinsinden tanımlayan değişmez parantez içinde sağlanır.

Aşağıdaki çizelgeler, seçicileri nesneye göre alfabetik sırayla aşağıdaki gibi listeler:

- Kuyruklar için Çizelge 549 sayfa 698 MQINQ öznitelik seçicileri
- Ad listesi için Çizelge 550 sayfa 700 MQINQ öznitelik seçicileri
- Süreç tanımlamaları için Çizelge 551 sayfa 701 MQINQ öznitelik seçicileri
- Kuyruk yöneticisine ilişkin Çizelge 552 sayfa 701 MQINQ öznitelik seçicileri

Not sütununda belirtilenler dışında, tüm seçiciler tüm IBM MQ platformlarında desteklenir:

DEĞİLz/OS

Tüm platformlarda desteklenir **hariç** z/OS

z/OS

z/OS üzerinde **yalnızca** desteklenir

Çizelge 549. Kuyruklar için MQINQ öznitelik seçicileri			
Seçici	Alanın uzunluğu	Açıklama	Not
MQCA_ALTERATION_DATE	MQ_DATE_LENGTH	En son değişiklik tarihi	
MQCA_ALTERATION_TIME	MQ_TIME_LENGTH	En son değişiklik saati	
MQCA_BACKOUT_REQ_Q_NAME	MQ_Q_NAME_LENGTH	Aşırı geriletme yeniden kuyruğa alma adı	
MQCA_BASE_Q_NAME	MQ_Q_NAME_LENGTH	Diğer adın çözümleneceği kuyruğun adı	
MQCA_CF_STRUC_NAME	MQ_CF_STRUC_NAME_LENGTH	Bağlaşım olanağı yapısı adı	z/OS
MQCA_CLUS_CHL_NAME	MQ_CHANNEL_NAME_LENGTH	Bu kuyruğu bir iletim kuyruğu olarak kullanan küme gönderen kanalın adı.	
MQCA_CLUSTER_NAME	MQ_CLUSTER_NAME_LENGTH	Küme adı	
MQCA_CLUSTER_NAMELIST	MQ_NAMELIST_NAME_LENGTH	Küme ad listesi	
MQCA_CREATION_DATE	MQ_CREATION_DATE_LENGTH	Kuyruk oluşturma tarihi	
MQCA_CREATION_TIME	MQ_CREATION_TIME_LENGTH	Kuyruk oluşturma zamanı	
MQCA_CUSTOM	MQ_CUSTOM_LENGTH	Yeni özellikler için özel öznitelik	
MQCA_INITIATION_Q_NAME	MQ_Q_NAME_LENGTH	Başlatma kuyruğu adı	
MQCA_PROCESS_NAME	MQ_PROCESS_NAME_LENGTH	Süreç tanımlamasının adı	
MQCA_Q_DESC	MQ_Q_DESC_LENGTH	Kuyruk tanımlaması	
MQCA_Q_NAME	MQ_Q_NAME_LENGTH	Kuyruk adı	
MQCA_REMOTE_Q_MGR_NAME	MQ_Q_MGR_NAME_LENGTH	Uzak kuyruk yöneticisinin adı	

Çizelge 549. Kuyruklar için MQINQ öznelik seçicileri (devamı var)

Seçici	Alanın uzunluğu	Açıklama	Not
MQCA_REMOTE_Q_NAME	MQ_Q_NAME_LENGTH	Uzak kuyruk yöneticisinde bilinen uzak kuyruğun adı	
MQCA_STORAGE_CLASS	MQ_STORAGE_CLASS_LENGTH	Depolama sınıfının adı	z/OS
MQCA_TRIGGER_DATA	MQ_TRIGGER_DATA_LENGTH	Tetikleyici verileri	
MQCA_XMIT_Q_NAME	MQ_Q_NAME_LENGTH	İletim kuyruğu adı	
MQIA_ACCOUNTING_Q	MQLONG	Kuyruğa ilişkin hesap verilerinin toplanmasını denetler	DEĞİL z/OS
MQIA_BACKOUT_THRESHOLD	MQLONG	Geriletme eşiği	
MQIA_CLWL_Q_PRIORITY	MQLONG	Kuyruğun önceliği	
MQIA_CLWL_Q_RANK	MQLONG	Kuyruk sırası	
MQIA_CLWL_USEQ	MQLONG	Uzak kuyrukları kullan	
MQIA_CURRENT_Q_DEPTH	MQLONG	Kuyruktaki ileti sayısı	
MQIA_DEF_BIND	MQLONG	Varsayılan bağ tanımı	
MQIA_DEF_INPUT_OPEN_OPTION	MQLONG	Varsayılan giriş için aç seçeneği	
MQIA_DEF_PERSISTENCE	MQLONG	Varsayılan ileti kalıcılığı	
MQIA_DEF_PRIORITY	MQLONG	Varsayılan ileti önceliği	
MQIA_DEFINITION_TYPE	MQLONG	Kuyruk tanımlaması tipi	
MQIA_DIST_LISTS	MQLONG	Dağıtım listesi desteği	DEĞİL z/OS
MQIA_HARDEN_GET_BACKOUT	MQLONG	Geriletme sayısının sertleşip güçlendirilmeyeceği	
MQIA_INDEX_TYPE	MQLONG	Kuyruk için korunan dizin tipi	z/OS
MQIA_INHIBIT_GET	MQLONG	Alma işlemlerine izin verilip verilmediğini belirler	
MQIA_INHIBIT_PUT	MQLONG	Koyma işlemlerine izin verilip verilmediğini belirler	
MQIA_MAX_MSG_LENGTH	MQLONG	İleti uzunluğu üst sınırı	
MQIA_MAX_Q_DEPTH	MQLONG	Kuyrukta izin verilen ileti sayısı üst sınırı	
MQIA_MSG_DELIVERY_SEQUENCE	MQLONG	İleti önceliğinin ilgili olup olmadığı	
MQIA_NPM_CLASS	MQLONG	Kalıcı olmayan iletiler için güvenilirlik düzeyi	
MQIA_OPEN_INPUT_COUNT	MQLONG	Giriş için kuyruğu açık olan MQOPEN çağrılarının sayısı	
MQIA_OPEN_OUTPUT_COUNT	MQLONG	Kuyruğun çıkış için açık olduğu MQOPEN çağrılarının sayısı	


<i>Çizelge 549. Kuyruklar için MQINQ öznitelik seçicileri (devamı var)</i>			
Seçici	Alanın uzunluğu	Açıklama	Not
MQIA_PROPERTY_CONTROL	MQLONG	Özellik denetimi özniteliği	
MQIA_Q_DEPTH_HIGH_EVENT	MQLONG	Kuyruk derinliği yüksek olayları için denetim özniteliği	DEĞİL z/OS
MQIA_Q_DEPTH_HIGH_LIMIT	MQLONG	Kuyruk derinliği için üst sınır	DEĞİL z/OS
MQIA_Q_DEPTH_LOW_EVENT	MQLONG	Kuyruk derinliği düşük olayları için denetim özniteliği	DEĞİL z/OS
MQIA_Q_DEPTH_LOW_LIMIT	MQLONG	Kuyruk derinliği için alt sınır	DEĞİL z/OS
MQIA_Q_DEPTH_MAX_EVENT	MQLONG	Kuyruk derinliği olay sayısı üst sınırı için denetim özniteliği	DEĞİL z/OS
MQIA_Q_SERVICE_INTERVAL	MQLONG	Kuyruk hizmeti aralığı sınırı	DEĞİL z/OS
MQIA_Q_SERVICE_INTERVAL_EVENT	MQLONG	Kuyruk hizmeti aralığı olayları için denetim özniteliği	DEĞİL z/OS
MQIA_Q_TYPE	MQLONG	Kuyruk tipi	
MQIA_QSG_DISP	MQLONG	Kuyruk paylaşım grubu yok etme	z/OS
MQIA_RETENTION_INTERVAL	MQLONG	Kuyruk alıkoyma aralığı	
MQIA_SCOPE	MQLONG	Kuyruk tanımlaması kapsamı	DEĞİL z/OS
MQIA_SHAREABILITY	MQLONG	Kuyruğun giriş için paylaşılıp paylaşılamayacağını belirler	
MQIA_STATISTICS_Q	MQLONG	Kuyruğa ilişkin istatistik verilerinin toplanmasını denetler	DEĞİL z/OS
MQIA_TRIGGER_CONTROL	MQLONG	Tetikleme denetimi	
MQIA_TRIGGER_DEPTH	MQLONG	Tetikleme derinliği	
MQIA_TRIGGER_MSG_PRIORITY	MQLONG	Tetikleyiciler için eşik iletisi önceliği	
MQIA_TRIGGER_TYPE	MQLONG	Tetikleme Tipi	
MQIA_USAGE	MQLONG	Kullanım	

<i>Çizelge 550. Ad listesi için MQINQ öznitelik seçicileri</i>			
Seçici	Alanın uzunluğu	Açıklama	Not
MQCA_ALTERATION_DATE	MQ_DATE_LENGTH	En son değişiklik tarihi	
MQCA_ALTERATION_TIME	MQ_TIME_LENGTH	En son değişiklik zamanı	
MQCA_NAMELIST_DESC	MQ_NAMELIST_DESC_LENGTH	Ad listesi açıklaması	
MQCA_NAMELIST_NAME	MQ_NAMELIST_NAME_LENGTH	Ad listesi nesnesinin adı	

<i>Çizelge 550. Ad listesi için MQINQ öznitelik seçicileri (devamı var)</i>			
Seçici	Alanın uzunluğu	Açıklama	Not
MQIA_NAMELIST_TYPE	MQLONG	Ad listesi tipi	z/OS
MQCA_NAMES	MQ_Q_NAME_LENGTH x Number of names in the list	Ad listesindeki adlar	
MQIA_NAME_COUNT	MQLONG	Ad listesindeki ad sayısı	
MQIA_QSG_DISP	MQLONG	Kuyruk paylaşım grubu yok etme	z/OS


<i>Çizelge 551. Süreç tanımlamaları için MQINQ öznitelik seçicileri</i>			
Seçici	Alanın uzunluğu	Açıklama	Not
MQCA_ALTERATION_DATE	MQ_DATE_LENGTH	En son değişiklik tarihi	
MQCA_ALTERATION_TIME	MQ_TIME_LENGTH	En son değişiklik zamanı	
MQCA_APPL_ID	MQ_PROCESS_APPL_ID_LENGTH	Uygulama tanıtıcısı	
MQCA_ENV_DATA	MQ_PROCESS_ENV_DATA_LENGTH	Ortam verileri	
MQCA_PROCESS_DESC	MQ_PROCESS_DESC_LENGTH	Süreç tanımlamasının tanımı	
MQCA_PROCESS_NAME	MQ_PROCESS_NAME_LENGTH	Süreç tanımlamasının adı	
MQCA_USER_DATA	MQ_PROCESS_USER_DATA_LENGTH	Kullanıcı verileri	
MQIA_APPL_TYPE	MQLONG	Uygulama tipi	
MQIA_QSG_DISP	MQLONG	Kuyruk paylaşım grubu yok etme	z/OS

<i>Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznitelik seçicileri</i>			
Seçici	Alanın uzunluğu	Açıklama	Not
MQCA_ALTERATION_DATE	MQ_DATE_LENGTH	En son değişiklik tarihi	
MQCA_ALTERATION_TIME	MQ_TIME_LENGTH	En son değişiklik zamanı	
MQCA_CHANNEL_AUTO_DEF_EXIT	MQ_EXIT_NAME_LENGTH	Otomatik kanal tanımlaması çıkış adı	
MQCA_CHINIT_SERVICE_PARM		IBM tarafından kullanılmak üzere ayrılmıştır	
MQCA_CLUSTER_WORKLOAD_DATA	MQ_EXIT_DATA_LENGTH	Küme iş yükü çıkışına geçirilen veriler	
MQCA_CLUSTER_WORKLOAD_EXIT	MQ_EXIT_NAME_LENGTH	Küme iş yükü çıkışının adı	
MQCA_COMMAND_INPUT_Q_NAME	MQ_Q_NAME_LENGTH	Sistem komutu giriş kuyruğu adı	
MQCA_CUSTOM	MQ_CUSTOM_LENGTH	Yeni özellikler için özel öznitelik	

Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznelik seçicileri (devamı var)			
Seçici	Alanın uzunluğu	Açıklama	Not
MQCA_DEAD_LETTER_Q_NAME	MQ_Q_NAME_LENGTH	Gönderilmeyen iletiler kuyruğunun adı	
MQCA_DEF_XMIT_Q_NAME	MQ_Q_NAME_LENGTH	Varsayılan iletim kuyruğu adı	
MQCA_DNS_GROUP	MQ_DNS_GROUP_NAME_LENGTH	Katılacak kuyruk paylaşım grubuna ilişkin gelen iletimleri işleyen TCP dinleyicisine ilişkin grubun adı. Ad, İş Yükü Yöneticisi Dinamik Etki Alanı Adı Hizmetleri kullanılırken geçerlidir.	z/OS
MQCA_IGQ_USER_ID	MQ_USER_ID_LENGTH	Grup içi kuyruğa alma kullanıcı kimliği	z/OS
 MQCA_INITIAL_KEY	MQ_INITIAL_KEY_LENGTH	Parola Koruma Sistemi için ilk anahtar	Boş değilse *** ** değeri ya da varsayılan ilk anahtar kullanılmıyorsa boşluk değeri döndürür.
MQCA_INSTALLATION_DESC	MQ_INSTALLATION_DESC_LENGTH	İlişkili kuruluşun açıklaması	z/OSd eğil. DEĞİ LIBM i
MQCA_INSTALLATION_NAME	MQ_INSTALLATION_NAME_LENGTH	Kuyruk yöneticisiyle ilişkili kuruluşun adı	z/OSd eğil. DEĞİ LIBM i
MQCA_INSTALLATION_PATH	MQ_INSTALLATION_PATH_LENGTH	İlişkili IBM MQ ' in kurulu olduğu yol	z/OSd eğil. DEĞİ LIBM i

Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznelik seçicileri (devamı var)			
Seçici	Alanın uzunluğu	Açıklama	Not
MQCA_LU_GROUP_NAME	MQ_LU_NAME_LENGTH	Kuyruk paylaşım grubunun kullanacağı gelen iletimleri işleyen LU 6.2 dinleyicisine ilişkin soysal LU adı	z/OS
MQCA_LU_NAME	MQ_LU_NAME_LENGTH	Giden LU 6.2 iletimleri için kullanılacak LU ' nun adı. Bu adı, dinleyicinin gelen iletimler için kullandığı aynı LU ' ya ayarlayın	z/OS
MQCA_LU62_ARM_SUFFIX	MQ_ARM_SUFFIX_LENGTH	Bu kanal başlatıcısı için LUADD ' yi gösteren SYS1 . PARMLIB üyenin APPCPM <i>xx</i> soneki	z/OS
MQCA_PARENT	MQ_Q_MGR_NAME_LENGTH	Bu kuyruk yöneticisinin üst ögesi olarak aday gösterilen, sıradüzensel olarak bağlı bir kuyruk yöneticisinin adı	
MQCA_Q_MGR_DESC	MQ_Q_MGR_DESC_LENGTH	Kuyruk yöneticisi açıklaması	
MQCA_Q_MGR_IDENTIFIER	MQ_Q_MGR_IDENTIFIER_LENGTH	Kuyruk yöneticisi tanıtıcısı (H)	
MQCA_Q_MGR_NAME	MQ_Q_MGR_NAME_LENGTH	Yerel kuyruk yöneticisinin adı	
MQCA_QSG_NAME	MQ_QSG_NAME_LENGTH	Kuyruk paylaşım grubu adı	z/OS
MQCA_REPOSITORY_NAME	MQ_CLUSTER_NAME_LENGTH	Kuyruk yöneticisinin havuz hizmetleri sağladığı kümenin adı	
MQCA_REPOSITORY_NAMELIST	MQ_NAMELIST_NAME_LENGTH	Kuyruk yöneticisinin havuz hizmetleri sağladığı kümelerin adlarını içeren ad listesi nesnesinin adı	


Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznelik seçicileri (devamı var)

Seçici	Alanın uzunluğu	Açıklama	Not
 MQCA_SSL_KEY_REPO_PASSWORD	MQ_SSL_ENCRYPT_KEY_REPO_PWD_LEN	Anahtar havuzu parolası	Boş değilse *** ** değeri ya da ayarlanmamışsa boş değeri döndürür. Depo lama dan önce ayarlandı ğında şifrelenir.
MQCA_TCP_NAME	MQ_TCP_NAME_LENGTH	Kullanmakta olduğunuz TCP/IP sisteminin adı	z/OS
MQIA_ACCOUNTING_CONN_OVERRIDE	MQLONG	Muhasebe ayarlarını geçersiz kıl	DEĞİ Lz/OS
MQIA_ACCOUNTING_INTERVAL	MQLONG	Ara muhasebe kayıtlarının ne sıklıkta yazılacağı	DEĞİ Lz/OS
MQIA_ACCOUNTING_MQI	MQLONG	MQI verileri için muhasebe bilgileri toplamasını denetler	DEĞİ Lz/OS
MQIA_ACCOUNTING_Q	MQLONG	Kuyruklara ilişkin hesap bilgilerinin toplanmasını denetler	DEĞİ Lz/OS
MQIA_ACTIVE_CHANNELS	MQLONG	Herhangi bir zamanda etkin olabilecek kanal sayısı üst sınırı	z/OS

Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznelik seçicileri (devamı var)			
Seçici	Alanın uzunluğu	Açıklama	Not
MQIA_ADOPTNEWMCA_CHECK	MQLONG	Bir MCA 'nın benimsenip benimsenmeyeceğini belirlemek için denetlenen öğeler. Denetim, zaten etkin olan bir MCA ile aynı ada sahip yeni bir gelen kanal saptandığında gerçekleştirilir.	z/OS
MQIA_ADOPTNEWMCA_INTERVAL	MQLONG	Yeni kanalın artık kanalın sona ermesini bekleyeceği süre (saniye)	DEĞİ Lz/O S
MQIA_ADOPTNEWMCA_TYPE	MQLONG	AdoptNewMCACheck parametreleriyle eşleşen yeni bir gelen kanal isteği saptandığında, belirli bir kanal tipindeki bir MCA 'nın artık kalmış bir örneğinin otomatik olarak yeniden başlatılıp başlatılmayacağı	z/OS
MQIA_AUTHORITY_EVENT	MQLONG	Yetki olayları için denetim özneliği	DEĞİ Lz/O S
MQIA_BRIDGE_EVENT	MQLONG	IMS köprü olayları için denetim özneliği	z/OS
MQIA_CHANNEL_AUTO_DEF	MQLONG	Otomatik kanal tanımlaması için denetim özneliği	DEĞİ Lz/O S
MQIA_CHANNEL_AUTO_DEF_EVENT	MQLONG	Otomatik kanal tanımlaması olayları için denetim özneliği	DEĞİ Lz/O S
MQIA_CHANNEL_EVENT	MQLONG	Kanal olayları için denetim özneliği	
MQIA_CHINIT_ADAPTERS	MQLONG	IBM MQ çağrılarını işlemek için kullanılacak bağdaştırıcı alt görevlerinin sayısı	z/OS
MQIA_CHINIT_DISPATCHERS	MQLONG	Kanal başlatıcısı için kullanılacak dağıtıcı sayısı	z/OS
MQIA_CHINIT_TRACE_AUTO_START	MQLONG	Kanal başlatıcı izlemesinin otomatik olarak başlatılıp başlatılmayacağını belirler	z/OS
MQIA_CHINIT_TRACE_TABLE_SIZE	MQLONG	Kanal başlatıcısının izleme verileri alanının büyüklüğü (MB)	z/OS
MQIA_CLUSTER_WORKLOAD_LENGTH	MQLONG	Küme iş yükü uzunluğu.	
MQIA_CLWL_MRU_CHANNELS	MQLONG	Küme iş yükü dengelemesi için en son kullanılan kanalların sayısı	
MQIA_CLWL_USEQ	MQLONG	Uzak kuyrukları kullan	
MQIA_CODED_CHAR_SET_ID	MQLONG	Kodlanmış karakter takımı tanıtıcısı	
MQIA_COMMAND_EVENT	MQLONG	Komut olayları için denetim özneliği	

Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznelik seçicileri (devamı var)			
Seçici	Alanın uzunluğu	Açıklama	Not
MQIA_COMMAND_LEVEL	MQLONG	Kuyruk yöneticisi tarafından desteklenen komut düzeyi	
MQIA_CONFIGURATION_EVENT	MQLONG	Yapılandırma olayları için denetim özneliği	DEĞİ Lz/O S
MQIA_DEF_CLUSTER_XMIT_Q_TYPE	MQLONG	Küme gönderen kanallar için kullanılacak varsayılan iletim kuyruğu tipi.	
MQIA_DIST_LISTS	MQLONG	Dağıtım listesi desteği	DEĞİ Lz/O S
MQIA_DNS_WLM	MQLONG	Kuyruk paylaşım grubu için gelen iletimleri işleyen TCP dinleyicisinin, Dinamik Etki Alanı Adı Hizmetleri için İş Yüğü Yöneticisi 'ne kaydedilip kaydedilmeyeceğini belirler	z/OS
MQIA_EXPIRY_INTERVAL	MQLONG	Süresi dolan iletilere ilişkin taramalar arasındaki aralık	z/OS
MQIA_GROUP_UR	MQLONG	Bu kuyruk yöneticisi için GROUP kurtarma birimlerinin etkinleştirilip etkinleştirilmediğine ilişkin denetim özneliği. GROUP kurtarma birimi yalnızca, kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiye kullanılabilir.	z/OS
MQIA_IGQ_PUT_AUTHORITY	MQLONG	Grup içi kuyruğa alma koyma yetkisi	z/OS
MQIA_INHIBIT_EVENT	MQLONG	Olayları engelleme için denetim özneliği	DEĞİ Lz/O S
MQIA_INTRA_GROUP_queuing	MQLONG	Grup içi kuyruğa alma desteği	z/OS
MQIA_LISTENER_TIMER	MQLONG	IBM MQ , APPC ya da TCP/IP başarısız olursa dinleyiciyi yeniden başlatmayı dener.	z/OS
MQIA_LOCAL_EVENT	MQLONG	Yerel olaylar için denetim özneliği	DEĞİ Lz/O S
MQIA_LOGGER_EVENT	MQLONG	Olayları engelleme için denetim özneliği	DEĞİ Lz/O S
MQIA_LU62_CHANNELS	MQLONG	LU 6.2 iletim protokolünü kullanarak, yürürlükteki kanal ya da bağlanabilen istemci sayısı üst sınırı	z/OS

Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznelik seçicileri (devamı var)

Seçici	Alanın uzunluğu	Açıklama	Not
MQIA_MSG_MARK_BROWSE_INTERVAL	MQLONG	Kuyruk yöneticisinin göz atma iletilerinden bir işareti otomatik olarak kaldırabileceği zaman aralığı (milisaniye).  Uyarı: Bu değeri varsayılan 5000 değerinin altına ayarlamamalısınız.	
MQIA_MAX_CHANNELS	MQLONG	Geçerli olabilecek kanal sayısı üst sınırı (bağlı istemcilerle sunucu bağlantısı kanalları dahil)	z/OS
MQIA_MAX_HANDLES	MQLONG	Tanıtıcı sayısı üst sınırı	
MQIA_MAX_MSG_LENGTH	MQLONG	İleti uzunluğu üst sınırı	
MQIA_MAX_PRIORITY	MQLONG	Öncelik üst sınırı	
MQIA_MAX_UNCOMMITTED_MESSAGES	MQLONG	Bir iş birimindeki kesinleştirilmemiş ileti sayısı üst sınırı	
MQIA_OUTBOUND_PORT_MAX	MQLONG	MQIA_OUTBOUND_PORT_MIN ile, giden kanallar için bağ tanımlanırken kullanılacak kapı numaraları aralığını tanımlar	z/OS
MQIA_OUTBOUND_PORT_MIN	MQLONG	MQIA_OUTBOUND_PORT_MAX ile, giden kanallar için bağ tanımlanırken kullanılacak kapı numaraları aralığını tanımlar	z/OS
MQIA_PERFORMANCE_EVENT	MQLONG	Performans olayları için denetim özneliği	DEĞİLz/OS
MQIA_PLATFORM	MQLONG	Kuyruk yöneticisinin bulunduğu platform	
MQIA_PROT_POLICY_CAPABILITY	MQLONG	Advanced Message Security güvenlik yeteneklerinin bir kuyruk yöneticisi için kullanılabilir olup olmadığını belirtir.	
MQIA_PUBSUB_MAXMSG_RETRY_COUNT	MQLONG	Başarısız olan bir komut iletisini eşitleme noktası altında yeniden işleme girişimi sayısı	

Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznelik seçicileri (devamı var)

Seçici	Alanın uzunluğu	Açıklama	Not
MQIA_PUBSUB_MODE	MQLONG	Yayınlama/abone olma motorunun ve kuyruğa alınan yayınlama/abone olma arabiriminin çalışıp çalışmadığını belirler. Uygulama programlama arabirimi kullanılarak yayınlanacak ya da abone olunacak uygulamalar, yayınlama/abone olma motorunu gerektirir. Kuyruğa alınan yayınlama/abone olma arabirimi tarafından izlenen kuyruklar, kuyruğa alınan yayınlama/abone olma arabiriminin çalışıyor olmasını gerektirir.	
MQIA_PUBSUB_NP_MSG	MQLONG	Teslim edilmemiş bir giriş iletisinin atılıp atılmayacağı (ya da alınıp tutulmayacağı)	
MQIA_PUBSUB_NP_RESP	MQLONG	Teslim edilmemiş yanıt iletilerinin davranışını denetler	
MQIA_PUBSUB_SYNC_PT	MQLONG	Yalnızca kalıcı (ya da tüm) iletilerin eşitleme noktası altında işlenip işlenmeyeceğini belirler	
MQIA_QMGR_CFCONLOS	MQLONG	Kuyruk yöneticisi yönetim yapısına ya da CFCONLOS ayarı ASQMGR olan herhangi bir CF yapısına bağlılığı kaybettiğinde yapılacak işlemi belirtir.	z/OS
MQIA_RECEIVE_TIMEOUT	MQLONG	Bir TCP/IP kanalının, etkin olmayan duruma dönmeden önce ortağından sağlıklı işletim bildirimleri de içinde olmak üzere veri almak için ne kadar süre beklediği. Değer, MQIA_RECEIVE_TIMEOUT_TYPE ile nitelenmiş sayısal bir değerdir.	z/OS
MQIA_RECEIVE_TIMEOUT_MIN	MQLONG	Bir TCP/IP kanalının, etkin olmayan duruma dönmeden önce ortağından sağlıklı işletim bildirimleri de içinde olmak üzere veri almak için bekleyeceği süre alt sınırı	z/OS
MQIA_RECEIVE_TIMEOUT_TYPE	MQLONG	Bir TCP/IP kanalının, etkin olmayan duruma dönmeden önce ortağından sağlıklı işletim bildirimleri de içinde olmak üzere veri almak için ne kadar süre beklediği. MQIA_RECEIVE_TIMEOUT_TYPE, MQIA_RECEIVE_TIMEOUT' a uygulanan niteleyicidir.	z/OS
MQIA_REMOTE_EVENT	MQLONG	Uzak olaylar için denetim özneliği	DEĞİL z/OS
MQIA_SECURITY_CASE	MQLONG	Güvenlik profilleri vakası	z/OS

Çizelge 552. Kuyruk yöneticisine ilişkin MQINQ öznitelik seçicileri (devamı var)			
Seçici	Alanın uzunluğu	Açıklama	Not
MQIA_SSL_EVENT	MQLONG	Kanal olayları için denetim özniteliği	
MQIA_SSL_FIPS_REQUIRED	MQLONG	Şifreleme için yalnızca FIPS sertifikalı algoritmaları kullan	
MQIA_SSL_RESET_COUNT	MQLONG	TLS anahtar sıfırlama sayısı	
MQIA_START_STOP_EVENT	MQLONG	Durdurma başlatma olayları için denetim özniteliği	DEĞİ Lz/O S
MQIA_STATISTICS_AUTO_CLUSTER	MQLONG	Küme gönderen kanalları için istatistik izleme bilgilerinin toplanmasını denetler	
MQIA_STATISTICS_CHANNEL	MQLONG	Kanallar için istatistik verilerinin toplanmasını denetler	
MQIA_STATISTICS_INTERVAL	MQLONG	İstatistik izleme verilerinin yazma sıklığı	DEĞİ Lz/O S
MQIA_STATISTICS_MQI	MQLONG	Kuyruk yöneticisine ilişkin istatistik izleme bilgilerinin toplanmasını denetler	DEĞİ Lz/O S
MQIA_STATISTICS_Q	MQLONG	Kuyruklara ilişkin istatistik verilerinin toplanmasını denetler	DEĞİ Lz/O S
MQIA_SYNCPOINT	MQLONG	eşitleme noktası kullanılabilirliği	
MQIA_TCP_CHANNELS	MQLONG	TCP/IP iletim protokolünü kullanarak, yürürlükteki kanal ya da bağlanabilen istemci sayısı üst sınırı	z/OS
MQIA_TCP_KEEP_ALIVE	MQLONG	Bağlantının diğer ucunun hala kullanılabilir olup olmadığını denetlemek için TCP KEEPALIVE olanağının kullanılıp kullanılmayacağını belirler	z/OS
MQIA_TCP_STACK_TYPE	MQLONG	Kanal başlatıcının yalnızca TCPNAME içinde belirtilen TCP/IP adres alanını mı kullanabileceğini, yoksa isteğe bağlı olarak seçilen herhangi bir TCP/IP adresine mi bağlanabileceğini belirler.	z/OS
MQIA_TRACE_ROUTE_RECORDING	MQLONG	İzleme rotası bilgilerinin kaydedilmesini denetler	z/OS
MQIA_TREE_LIFE_TIME	MQLONG	Kullanılmayan yönetimle ilgili olmayan konuların geçerlik süresi	
MQIA_TRIGGER_INTERVAL	MQLONG	Tetikleme aralığı	

IntAttrSayısı

Tip: MQLONG -giriş

Bu, *IntAttrs* dizisindeki öğelerin sayısıdır. Sıfır geçerli bir değerdir.

`IntAttrSayı` en az **Selectors** parametresindeki MQIA_* seçicilerinin sayısıysa, istenen tüm tamsayı öznitelikleri döndürülür.

IntAttrs

Tip: MQLONG x *IntAttrCount* -output

Bu, *IntAttrCount* tamsayı öznitelik değerlerinin bir dizisidir.

Tamsayı öznitelik değerleri, **Selectors** parametresindeki MQIA_* seçicileri ile aynı sırayla döndürülür. Dizi, MQIA_* seçicilerinin sayısından daha fazla öğe içeriyorsa, fazla öğeler değişmez.

Hobj bir kuyruğu gösteriyorsa, ancak öznitelik seçici o kuyruk tipi için geçerli değilse, MQIAV_NOT_APPLICABLE değeri döndürülür. *IntAttrs* dizisindeki ilgili öğe için döndürülür.

IntAttrCount ya da **SelectorCount** parametresi sıfırsa, *IntAttrs* değerine başvurmaz. Bu durumda, C ya da S/390 çevirici dilinde yazılan programlar tarafından geçirilen parametre adresi boş olabilir.

CharAttrUzunluğu

Tip: MQLONG -giriş

Bu, **CharAttrs** parametresinin bayt cinsinden uzunluğudur.

`CharAttrUzunluk`, istenen karakter özniteliklerinin uzunluklarının en az toplamı olmalıdır (bkz. Seçiciler). Sıfır geçerli bir değerdir.

CharAttrs

Tip: MQCHAR x *CharAttrLength* -output

Bu, karakter özniteliklerinin birlikte döndürüldüğü ve birleştirildiği arabellektir. Arabelleğin uzunluğu **CharAttrLength** değışırtirgesiyle verilir.

Karakter öznitelikleri, **Selectors** parametresindeki MQCA_* seçicileri ile aynı sırayla döndürülür. Her öznitelik dizgisinin uzunluğu, her öznitelik için sabitlenir (bkz. Seçiciler) ve içindeki değer, gerekirse boşluklarla sağa doğru doldurulur. İstenen tüm karakter özniteliklerini ve dolguyu içermek için gerekenden daha büyük bir arabellek ekleyebilirsiniz. Döndürülen son öznitelik değerinin ötesindeki baytlar değişmez.

Hobj bir kuyruğu gösteriyorsa, ancak öznitelik seçici bu kuyruk tipi için geçerli değilse, tamamen yıldız işaretlerinden (*) oluşan bir karakter dizgisi döndürülür. Yıldız işareti, *CharAttrs* içinde bu özniteliğin değeri olarak döndürülür.

CharAttrLength ya da **SelectorCount** parametresi sıfırsa, *CharAttrs* değerine başvurmaz. Bu durumda, C ya da S/390 çevirici dilinde yazılan programlar tarafından geçirilen parametre adresi boş olabilir.

CompCode

Tip: MQLONG -output

Tamamlanma kodu:

MQCC_OK

Başarıyla tamamlandı.

MQCC_WARNING

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRD_CHAR_ATTRS_TOO_SHORT
(2008, X'7D8') Karakter öznitelikleri için yeterli yer yok.

MQRD_INT_ATTR_COUNT_TOO_SMALL
(2022, X'7E6') Tamsayı öznitelikleri için yeterli yer yok.

MQRD_SELECTOR_NOT_FOR_TYPE
(2068, X'814') Seçici kuyruk tipi için geçerli değil.

CompCode MQRD_FAILEDise:

MQRD_ADAPTER_NOT_AVAILABLE
(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRD_ADAPTER_SERV_LOAD_ERROR
(2130, X'852') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRD_API_EXIT_ERROR
(2374, X'946') API çıkışı başarısız oldu.

MQRD_API_EXIT_LOAD_ERROR
(2183, X'887') API çıkışı yüklenemiyor.

MQRD_ASID_MISMATCH
(2157, X'86D') Birincil ve ana ASID ' ler farklıdır.

MQRD_CALL_IN_PROGRESS
(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRD_CF_STRUC_FAILED
(2373, X'945') Coupling-facility yapısı başarısız oldu.

MQRD_CF_STRUC_IN_USE
(2346, X'92A') Kullanımda olan Coupling-facility yapısı.

MQRD_CHAR_ATTR_LENGTH_ERROR
(2006, X'7D6') Karakter özniteliklerinin uzunluğu geçersiz.

MQRD_CHAR_ATTRS_ERROR
(2007, X'7D7') Karakter öznitelikleri dizgisi geçersiz.

MQRD_CICS_WAIT_FAILED
(2140, X'85C') Bekleme isteği CICS tarafından reddedildi.

MQRD_CONNECTION_BROKEN
(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRD_CONNECTION_NOT_AUTHORIZED
(2217, X'8A9') Bağlantı için yetkili değil.

MQRD_CONNECTION_STOPPING
(2203, X'89B') Bağlantı kapatılıyor.

MQRD_HCONN_ERROR
(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRD_HOBJ_ERROR
(2019, X'7E3') Nesne tanıtıcısı geçersiz.

MQRD_INT_ATTR_COUNT_ERROR
(2021, X'7E5') Tamsayı öznitelikleri sayısı geçerli değil.

MQRD_INT_ATTRS_ARRAY_ERROR
(2023, X'7E7') Tamsayı öznitelikleri dizisi geçerli değil.

MQRD_NOT_OPEN_FOR_INQUIRE
(2038, X'7F6') Kuyruk, sorma için açık değil.

MQRD_OBJECT_CHANGED
(2041, X'7F9') Açıldığında nesne tanımlaması değiştirildi.

MQRD_OBJECT_DAMAGED
(2101, X'835') Nesne zarar gördü.

MQRC_PAGESET_ERROR

(2193, X' 891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRC_Q_DELETED

(2052, X' 804 ') Kuyruk silindi.

MQRC_Q_MGR_NAME_ERROR

(2058, X' 80A ') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRC_Q_MGR_NOT_AVAILABLE

(2059, X' 80B ') Bağlantı için kuyruk yöneticisi kullanılamıyor.

MQRC_Q_MGR_STOPPING

(2162, X' 872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM

(2102, X' 836 ') Kullanılabilir sistem kaynakları yetersiz.

MQRC_SELECTOR_COUNT_ERROR

(2065, X' 811 ') Seçici sayısı geçerli değil.

MQRC_SELECTOR_ERROR

(2067, X' 813 ') Öznitelik seçici geçerli değil.

MQRC_SELECTOR_LIMIT_EXCEEDED

(2066, X' 812 ') Seçicilerin sayısı çok büyük.

MQRC_STORAGE_NOT_AVAILABLE

(2071, X' 817 ') Kullanılabilir bellek yetersiz.

MQRC_SUPPRESSED_BY_EXIT

(2109, X' 83D ') Çağrı çıkış programı tarafından engellendi.

MQRC_UNEXPECTED_ERROR

(2195, X' 893 ') Beklenmeyen bir hata oluştu.

Bu kodlara ilişkin ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#)

Kullanım notları

1. Döndürülen değerler, seçilen özniteliklerin anlık görüntüsüdür. Uygulamanın döndürülen değerlere göre hareket edebilmesi için özniteliklerin aynı kalacağı garanti edilmez.
2. Bir model kuyruğunu açtığınızda, dinamik bir yerel kuyruk yaratılır. Dinamik bir yerel kuyruk, öznitelikleri hakkında bilgi almak üzere model kuyruğunu açsanız bile yaratılır.

Dinamik kuyruğun öznitelikleri, dinamik kuyruk yaratıldığı sırada model kuyruğunun öznitelikleriyle büyük ölçüde aynıdır. Daha sonra bu kuyrukte MQINQ çağrısı kullanırsanız, kuyruk yöneticisi model kuyruğunun özniteliklerini değil, dinamik kuyruğun özniteliklerini döndürür. Dinamik kuyruk tarafından devralınan model kuyruğunun özniteliklerine ilişkin ayrıntılar için bkz. [Çizelge 561 sayfa 828](#).
3. Sorulmakta olan nesne bir diğer ad kuyruğuysa, MQINQ çağrısının döndürdüğü öznitelik değerleri, diğer ad kuyruğunun öznitelikleridir. Bu öznitelikler, diğer adın çözüldüğü temel kuyruğun ya da konunun öznitelikleri değildir.
4. Sorulacak nesne bir küme kuyruğuysa, sorulabilecek öznitelikler kuyruğun nasıl açılmasına bağlıdır:
 - Sorgu için bir ya da daha fazla giriş, göz atma ya da ayarlama işlemi için bir küme kuyruğu açabilirsiniz. Bunu yapmak için, açmanın başarılı olması için küme kuyruğunun yerel bir eşgörünümü olmalıdır. Bu durumda, sorulabilecek öznitelikler, yerel kuyruklar için geçerli olan özniteliklerdir.

Küme kuyruğu giriş, göz atma ya da küme belirtilmeden sorgu için açıksa, küme kuyrukları için değil, yalnızca yerel kuyruklar için geçerli olan öznitelikleri sormaya çalışırsanız, çağrı tamamlanma kodu MQCC_WARNING ve neden kodu MQRC_SELECTOR_NOT_FOR_TYPE (2068) döndürülür.
 - Bağlı kuyruk yöneticisinin temel kuyruk yöneticisi adını geçirirken, sorgu için bir küme kuyruğu açabilirsiniz.

Bunu yapmak için, açmanın başarılı olması için küme kuyruğunun yerel bir eşgörünümü olmalıdır. Temel kuyruk yöneticisi geçirilmezse, küme kuyrukları için değil, yalnızca yerel kuyruklar için geçerli

olan öznitelikleri sormaya çalışırsanız, çağrı tamamlama kodu MQCC_WARNING ve neden kodu MQRC_SELECTOR_NOT_FOR_TYPE (2068) döndürür.

- Küme kuyruğu tek başına sorgu ya da sorgu ve çıkış için açılırsa, yalnızca listelenen öznitelikler sorgulanabilir. **QType** özniteliği bu durumda MQQT_CLUSTER değerine sahiptir:

- MQCA_Q_DESC
- MQCA_Q_NAME
- MQIA_DEF_BIND
- MQIA_DEF_PERSISTENCE
- MQIA_DEF_PRIORITY
- MQIA_INHIBIT_PUT
- MQIA_Q_TYPE

Küme kuyruğunu sabit bağ tanımlı olmadan açabilirsiniz. Bunu, MQOPEN çağrısında belirtilen MQOO_BIND_NOT_FIXED ile açabilirsiniz. Diğer bir seçenek olarak, MQOO_BIND_AS_Q_DEFdeğerini belirtin ve kuyruğun **DefBind** özniteliğini MQBND_BIND_NOT_FIXEDolarak ayarlayın. Değişmez bağ tanımlı olmayan bir küme kuyruğunu açarsanız, kuyruk için ardışık MQINQ çağrıları küme kuyruğunun farklı eşgörünümlerini sorgulayabilir. Ancak, tüm eşgörünümlerin aynı öznitelik değerlerine sahip olması tipik bir durumdur.

- Bir küme için diğer ad kuyruğu nesnesi tanımlanabilir. TARGTYPE ve TARGET küme öznitelikleri olmadığından, diğer ad kuyruğunda MQOPEN işlemini gerçekleştiren işlem, diğer adın çözümleneceği nesneyi bilmez.

İlk MQOPENsırasında, diğer ad kuyruğu bir kuyruk yöneticisine ve kümedeki bir kuyruğa çözülür. Ad çözme işlemi uzak kuyruk yöneticisinde yeniden gerçekleşir ve diğer ad kuyruğunun TARGTYPE ' nin çözülmesi burada gerçekleşir.

Diğer ad kuyruğu bir konu diğer adına çözülürse, diğer ad kuyruğuna konan iletilerin yayınlanması bu uzak kuyruk yöneticisinde gerçekleşir.

Bkz. [Küme kuyrukları](#)

5. Bir dizi özniteliği sorgulamak ve daha sonra, MQSET aramasını kullanarak bunların bazılarını ayarlamak isteyebilirsiniz. Sorgulamayı programlamak ve verimli bir şekilde ayarlamak için, seçici dizilerinin başında ayarlanacak öznitelikleri konumlandırın. Bunu yaparsanız, MQSETiçin azaltılmış sayılarla aynı diziler kullanılabilir.
6. Birden çok uyarı durumu ortaya çıkarsa (**CompCode** değiştirgesine bakın), döndürülen neden kodu, aşağıdaki listede geçerli olan ilk durumdur:
 - a. MQRC_SELECTOR_NOT_FOR_TYPE
 - b. MQRC_INT_ATTR_COUNT_TOO_SMALL
 - c. MQRC_CHAR_ATTRS_TOO_SHORT

7. Aşağıdaki konu, nesne öznitelikleriyle ilgili bilgi içerir:

- [“Kuyruklara ilişkin öznitelikler” sayfa 826](#)
- [“Ad listesi öznitelikleri” sayfa 858](#)
- [“Süreç tanımlamalarına ilişkin öznitelikler” sayfa 860](#)
- [“Kuyruk yöneticisine ilişkin öznitelikler” sayfa 789](#)

C çağrısı

```
MQINQ (Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs,  
CharAttrLength, CharAttrs, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

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 */

```

COBOL çağırısı

```

CALL 'MQINQ' USING HCONN, HOBJ, SELECTORCOUNT, SELECTORS-TABLE,
                  INTATTRCOUNT, INTATTRS-TABLE, CHARATTRLENGTH,
                  CHARATTRS, COMPCODE, REASON.

```

Değiştirgeleri aşağıdaki gibi bildirin:

```

** 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.

```

PL/I çağırısı

```

call MQINQ (Hconn, Hobj, SelectorCount, Selectors, IntAttrCount,
            IntAttrs, CharAttrLength, CharAttrs, CompCode, Reason);

```

Değiştirgeleri aşağıdaki gibi bildirin:

```

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 */

```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL MQINQ, (HCONN, HOBJ, SELECTORCOUNT, SELECTORS, INTATTRCOUNT, X  
INTATTRS, CHARATTRLENGTH, CHARATTRS, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

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

Visual Basic çağrısı

```
MQINQ Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs,  
CharAttrLength, CharAttrs, CompCode, Reason
```

Değiştirgeleri aşağıdaki gibi bildirin:

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-İleti sorgusuna ilişkin özellik

MQINQMP çağrısı, bir iletinin özelliğinin değerini döndürür.

Sözdizimi

MQINQMP (*Hconn, Hmsg, InqPropOpts, Name, PropDesc, Type, ValueLength, Value, DataLength, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıttıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri, **Hmsg** değiştirgesinde belirtilen ileti tanıttıcısını yaratmak için kullanılan bağlantı tanıttıcısıyla eşleşmelidir.

İleti tanıttıcısı MQHC_UNASSOCIATED_HCONN kullanılarak yaratıldıysa, ileti tanıttıcısının bir özelliğini soran iş parçacığında geçerli bir bağlantı kurulmalıdır; tersi durumda, çağrı MQRC_CONNECTION_BROKEN ile başarısız olur.

İtt

Tip: MQHMSG-giriş

Bu, sorulacak ileti tanıttıcısıdır. Değer, önceki bir **MQCRTMH** çağrısı tarafından döndürüldü.

InqPropOpts (InqProp)

Tip: MQIMPO-giriş/çıkış

Ayrıntılar için [MQIMPO](#) veri tipine bakın.

Ad

Tip: MQCHARV-giriş/çıkış

Sorulacak özelliğin adı.

Bu ada sahip bir özellik bulunamazsa, çağrı MQRC_PROPERTY_NOT_ALLOWED nedeniyle başarısız olur.

Özellik adının sonunda genel arama karakteri yüzde işaretini (%) kullanabilirsiniz. Joker karakter, nokta (.) karakteri de içinde olmak üzere sıfır ya da daha fazla karakterle eşleşir. Bu, bir uygulamanın birçok özelliğin değerini sorgulamasına olanak sağlar. MQINQMP ' yi MQIMPO_INQ_FIRST seçeneğiyle, ilk eşleşen özelliği almak için ve sonraki eşleşen özelliği almak için MQIMPO_INQ_NEXT seçeneğiyle yeniden çağırın. Eşleşen başka özellik yoksa, çağrı MQRC_PROPERTY_NOT_ALLOWED ile başarısız olur. InqPropOpts yapısının *ReturnedName* alanı, özelliğin döndürülen adı için bir adres ya da görelî konumla kullanıma hazırlandıysa, bu, eşleştirilen özelliğin adıyla MQINQMP ' den döndürüldükten sonra tamamlanır. InqPropOpts yapısındaki *ReturnedName* alanının *VSBufSize* alanı döndürülen özellik adının uzunluğundan azsa, tamamlama kodu MQCC_FAILED olarak ayarlanır. Neden MQRC_NAME_TOO_BIG?

Bilinen eşanlamlıları olan özellikler aşağıdaki gibi döndürülür:

1. "mqps" öneğine sahip özellikler. IBM MQ özellik adı olarak döndürülür. Örneğin, "MQTopicString", "mqps.Top" yerine döndürülen addır.
2. "jms" öneğine sahip özellikler. Ya da "McD." JMS üstbilgi alanı adı olarak döndürülür; örneğin, "JMSExpiration", "jms.Exp" yerine döndürülen addır.
3. "usr." öneğine sahip özellikler bu önek olmadan döndürülür; örneğin, "usr.Color" yerine "Color" döndürülür.

Eşanlamlıları olan özellikler yalnızca bir kez döndürülür.

C programlama dilinde, tüm özellikler ve "usr" ile başlayan tüm özellikler için aşağıdaki makro değişkenleri tanımlanır:

MQPROP_INQUIRE_ALL

İletinin tüm özelliklerini sorgulamanızı sağlar.

MQPROP_INQUIRE_ALL aşağıdaki şekilde kullanılabilir:

```
MQCHARV Name = {MQPROP_INQUIRE_ALL};
```

MQPROP_INQUIRE_ALL_USR

"usr." ile başlayan iletinin tüm özelliklerini sorgulamanızı sağlar. Döndürülen ad "usr" olmadan döndürülür. önek.

MQIMP_INQ_NEXT belirtilirse, ancak Ad önceki çağrıdan sonra değiştiyse ya da bu ilk çağrıysa, MQIMPO_INQ_FIRST örtük olarak belirtilir.

Özellik adlarının kullanımıyla ilgili ek bilgi için [Özellik adları](#) ve [Özellik adı kısıtlamaları](#) konusuna bakın.

PropDesc

Tip: MQPD-çıkış

Bu yapı, özellik desteklenmiyorsa ne olacağı, özelliğin hangi ileti bağlamına ait olduğu ve özelliğin hangi iletilere kopyalanması gerektiği de içinde olmak üzere, bir özelliğin özniteliklerini tanımlamak için kullanılır. Bu yapının ayrıntıları için bkz. [MQPD](#).

Tip

Tip: MQLONG-giriş/çıkış

MQINQMP çağrısından dönüş için bu parametre *Değer* veri tipine ayarlanır. Veri tipi aşağıdakilerden biri olabilir:

MQTYPE_BOOLEAN

Bir boole.

MQTYPE_BYTE_STRING

bir bayt dizilimi.

MQTYPE_INT8

8 bitlik işaretli bir tamsayı.

MQTYPE_INT16

16 bit işaretli tamsayı.

MQTYPE_INT32

32 bit işaretli tamsayı.

MQTYPE_INT64

64 bitlik imzalı bir tamsayı.

MQTYPE_FLOAT32

32 bit kayan noktalı sayı.

MQTYPE_FLOAT64

64 bit kayan noktalı sayı.

MQTYPE_STRING

Bir karakter dizisi.

MQTYPE_NULL (boş değerli)

Özellik var, ancak boş değerli.

Özellik değerinin veri tipi tanınmazsa, MQTYPE_STRING döndürülür ve değer dizgi gösterimi *Değer* alanına yerleştirilir. Veri tipinin dizgi gösterimi, *InqPropOpts* parametresinin *TypeString* alanında bulunabilir. MQRC_PROP_TYPE_NOT_SUPPORTED nedeniyle bir uyarı tamamlanma kodu döndürüldü.

Ayrıca, MQIMPO_CONVERT_TYPE seçeneği belirtilirse, özellik değerinin dönüştürülmesi isteniyor. Özelliğin döndürülmesini istediğiniz veri tipini belirtmek için giriş olarak *Tip* ögesini kullanın. Veri tipi dönüştürmeye ilişkin ayrıntılar için [MQIMPO](#) yapısının [MQIMPO_CONVERT_TYPE](#) seçeneğine bakın.

Tip dönüştürme isteğinde bulunmazsanız, girişte aşağıdaki değeri kullanabilirsiniz:

MQTYPE_AS_SET

Özelliğin değeri, veri tipi dönüştürülmeden döndürülür.

ValueLength

Tip: MQLONG-input

Değer alanının bayt cinsinden uzunluğu. Değer döndürülmesini gerektirmediğiniz özellikler için sıfır belirtin. Bunlar, bir uygulama tarafından boş değere ya da boş bir dizgiye sahip olacak şekilde tasarlanmış özellikler olabilir. [MQIMPO_QUERY_LENGTH](#) seçeneği belirtildiyse sıfır değerini de belirtin; bu durumda değer döndürülmez.

Değer

Tip: MQBYTEEx *ValueLength* -çıkış

Bu, sorulmuş özellik değerini içerecek alandır. Arabellek, döndürülmekte olan değer için uygun bir sınıra hizalanmalıdır. Bunun yapılmaması, değere daha sonra erişildiğinde bir hatayla sonuçlanabilir.

ValueLength özellik değerinin uzunluğundan azsa, özellik değerinin mümkün olduğu kadar çoğu *Değer* ' e taşınır ve çağrı MQCC_FAILED tamamlama koduyla başarısız olur ve neden MQRC_19TVALUE_TOO_BIG.

Değer içindeki verilerin karakter takımı, InqPropOpts parametresindeki ReturnedCCSID alanı tarafından verilir. *Değer* içindeki verilerin kodlaması, InqPropOpts parametresindeki ReturnedEncoding alanı tarafından verilir.

C programlama dilinde, parametre bir gösterge-to-void olarak bildirilir; herhangi bir veri tipinin adresi parametre olarak belirtilebilir.

ValueLength parametresi sıfırsa, *Değer* parametresine başvuruda bulunulmaz ve değeri C ya da System/390 derleyicisine yazılan programlar tarafından iletilir.

DataLength

Tip: MQLONG-output

Bu, *Değer* alanında döndürülen gerçek özellik değerinin bayt cinsinden uzunluğudur.

DataLength özellik değeri uzunluğundan azsa, MQINQMP çağrısından dönüşte *DataLength* yine de doldurulur. Bu, uygulamanın özellik değerini karşılamak için gereken arabellek büyüklüğünü belirlemesini sağlar ve daha sonra, çağrıyı uygun büyüklükte bir arabellekle yeniden yürütün.

Aşağıdaki değerler de döndürülebilir.

Tip değiştirgesi MQTYPE_STRING ya da MQTYPE_BYTE_STRING olarak ayarlandıysa:

MQVL_EMPTY_STRING

Özellik var, ancak karakter ya da byte içermiyor.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_PROP_NAME_NOT_CONVERTED

(2492, X'09BC') Döndürülen özellik adı dönüştürülmedi.

MQRC_PROP_VALUE_NOT_CONVERTED

(2466, X'09A2') Özellik değeri dönüştürülmedi.

MQRC_PROP_TYPE_NOT_SUPPORTED

(2467, X'09A3') Özellik veri tipi desteklenmiyor.

MQRC_RFH_FORMAT_HATA

(2421, X'0975 ') Özellikleri içeren bir MQRFH2 klasörü ayrıştırılmadı.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'089C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'0852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'086D') Birincil ve ana ASID' ler farklıdır.

MQRC_BUFFER_ERROR

(2004, X'07D4') Değer parametresi geçersiz.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'07D5') Değer uzunluğu parametresi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CONNECTION_BROKEN

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_DATA_LENGTH_HATA

(2010, X'07DA') Veri uzunluğu parametresi geçersiz.

MQRC_IMPO_HATA

(2464, X'09A0') İleti sorgusuna ilişkin özellik seçenekleri yapısı geçersiz.

MQRC_HMSG_HATA

(2460, X'099C') İleti tanıtıcısı geçersiz.

MQRC_MSG_HANDLE_IN_USE

(2499, X'09C3') İleti tanıtıcısı kullanımda.

MQRC_OPTIONS_HATA

(2046, X'07F8') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_PD_ERROR (QRC_PD_HATA)

(2482, X'09B2') Özellik tanımlayıcı yapısı geçersiz.

MQRC_PROP_CONV_NOT_SUPPORTED

(2470, X'09A6') Gerçek veri tipinden istenen veri tipine dönüştürme desteklenmiyor.

MQRC_NAMETY_NAME_ERROR

(2442, X'098A') Özellik adı geçersiz.

MQRC_COUNTY_NAME_TOO_BIG

(2465, X'09A1') Özellik adı, döndürülen ad arabelleği için çok büyük.

MQRC_PROPERTY_NOT_VAR

(2471, X'09A7') Özellik yok.

MQRC_COUNTY_VALUE_TOO_BIG

(2469, X'09A5') Özellik değeri, Değer alanı için çok büyük.

MQRC_PROP_NUMBER_FORMAT_HATA

(2472, X'09A8') Değer verilerinde sayı biçimi hatası saptandı.

MQRC_19TY_TYPE_HATA

(2473, X'09A9') İstenen özellik tipi geçersiz.

MQRC_SOURCE_CCSID_HATA

(2111, X'083F') Özellik adı kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'0871 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'0893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

C çağrısı

```
MQINQMP (Hconn, Hmsg, &InqPropOpts, &Name, &PropDesc, &Type,
ValueLength, Value, &DataLength, &CompCode, &Reason);
```

Değiştirelileri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağırısı

```
CALL 'MQINQMP' USING HCONN, HMSG, INQMSGOPTS, NAME, PROPDESC, TYPE,
VALUELENGTH, VALUE, DATALENGTH, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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 CMQCHRUV.
** 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.
```

PL/I çağırısı

```
call MQINQMP (Hconn, Hmsg, InqPropOpts, Name, PropDesc, Type,
ValueLength, Value, DataLength, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
dcl Hconn fixed bin(31); /* Connection handle */
dcl Hmsg fixed bin(63); /* Message handle */
dcl InqPropOpts like MQIMPO; /* Options that control the action of MQINQMP */
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 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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQINQMP, (HCONN, HMSG, INQMSGOPTS, NAME, PROPDESC, TYPE,
VALUELENGTH, VALUE, DATALENGTH, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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-İleti tanıtıcısını arabelleğe dönüştür

MQMHBUF çağrısı, bir ileti tanıtıcısını arabelleğe dönüştürür ve MQBUFMH çağrısının tersidir.

Sözdizimi

MQMHBUF (*Hconn, Hmsg, MsgHBufOpts, Name, MsgDesc, BufferLength, Buffer, DataLength, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri, **Hmsg** değiştirilmesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eşleşmelidir.

İleti tanıtıcısı MQHC_UNASSOCIATED_HCONN kullanılarak yaratıldıysa, ileti tanıtıcısını silen iş parçacığında geçerli bir bağlantı kurulmalıdır. Geçerli bir bağlantı kurulmazsa, çağrı MQRC_CONNECTION_BROKEN ile başarısız olur.

İt

Tip: MQHMSG-giriş

Bu, arabelleğin gerekli olduğu ileti tanıtıcısıdır. Değer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

MsgHBufSeçmeler

Tip: MQMHBO-giriş

MQMHBO yapısı, uygulamaların, arabelleklerin ileti tanıtıcılarından nasıl üretildiğini denetleyen seçenekleri belirtmesine olanak sağlar.

Ayrıntılar için bkz. [“MQMHBO-Arabellek seçeneklerine ileti tanıtıcısı” sayfa 474.](#)

Ad

Tip: MQCHARV-giriş

Arabelleğe konacak özelliğin ya da özelliklerin adı.

Adla eşleşen bir özellik bulunamazsa, çağrı MQRC_PROPERTY_NOT_ALLOWED ile başarısız olur.

Arabelleğe birden çok özellik koymak için joker karakter kullanabilirsiniz. Bunu yapmak için özellik adının sonundaki '%' joker karakterini kullanın. Bu joker karakter, '!' de dahil olmak üzere sıfır ya da daha fazla karakterle eşleşir karakter.

C programlama dilinde, 'usr' ile başlayan tüm özelliklerde ve tüm özelliklerde sorma için aşağıdaki makro değişkenleri tanımlanır:

MQPROP_INQUIRE_ALL

İletinin tüm özelliklerini arabelleğe koy

MQPROP_INQUIRE_ALL_USR

İletinin 'usr.' karakterleriyle başlayan tüm özelliklerini girin. arabelleğe.

Özellik adlarının kullanımıyla ilgili ek bilgi için [Özellik adları](#) ve [Özellik adı kısıtlamaları](#) konusuna bakın.

MsgDesc

Tip: MQMD-giriş/çıkış

MsgDesc yapısı, arabellek alanının içeriğini açıklar.

Çıkışta, *Encoding*, *CodedCharSetId* ve *Format* alanları, çağrı tarafından yazıldığı gibi arabellek alanındaki verilerin kodlamasını, karakter kümesi tanıtıcısını ve biçimini doğru şekilde tanımlamak için ayarlanır.

Bu yapıdaki veriler, uygulamanın karakter kümesinde ve kodlamasında yer almaktadır.

BufferLength

Tip: MQLONG-input

BufferLength, Arabellek alanının bayt cinsinden uzunluğudur.

Arabellek

Tip: MQBYTEExBufferUzunluk-çıkış

Buffer, ileti özelliklerini içerecek alanı tanımlar. Arabelleği 4 baytlık bir sınırdaki hizalayın.

BufferLength, özellikleri *Buffer* içinde depolamak için gerekli uzunluktan azsa, MQMHBUF, MQRC_19TY_VALUE_TOO_BIG ile başarısız olur.

Arama başarısız olsa da arabelleğin içeriği değişebilir.

DataLength

Tip: MQLONG-output

DataLength, arabellekte döndürülen özelliklerin bayt cinsinden uzunluğudur. Değer sıfırsa, *Name* içinde belirtilen değerle hiçbir özellik eşleşmez ve çağrı MQRC_PROPERTY_NOT_ALLOWED neden koduyla başarısız olur.

BufferLength, özellikleri arabellekte saklamak için gereken uzunluktan azsa, MQMHBUF çağrısı MQRC_COUNTY_VALUE_TOO_BIG ile başarısız olur, ancak *DataLength*' a yine de bir değer girilir. Bu, uygulamanın özellikleri yerleştirmek için gereken arabellek büyüklüğünü saptamasını sağlar ve daha sonra, gerekli *BufferLength* ile çağrışı yeniden yürütün.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'089C') Bağdaştırıcı kullanılamıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_MHBO_HATA

(2501, X'095C') Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısı geçersiz.

MQRC_BUFFER_ERROR

(2004, X'07D4') Arabellek değiştirgesi geçersiz.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'07D5') Arabellek uzunluğu değiştirgesi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağırısı.

MQRC_CONNECTION_BROKEN

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_DATA_LENGTH_HATA

(2010, X'07DA') Veri uzunluğu parametresi geçersiz.

MQRC_HMSG_HATA

(2460, X'099C') İleti tanıtıcısı geçersiz.

MQRC_MD_ERROR

(2026, X'07EA') İleti tanımlayıcısı geçersiz.

MQRC_MSG_HANDLE_IN_USE

(2499, X'09C3') İleti tanıtıcısı kullanımda.

MQRC_OPTIONS_HATA

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_NAMETY_NAME_ERROR

(2442, X'098A') Özellik adı geçerli değil.

MQRC_PROPERTY_NOT_VAR

(2471, X'09A7') Özellik kullanılamıyor.

MQRC_COUNTY_VALUE_TOO_BIG

(2469, X'09A5') BufferLength değeri, belirtilen özellikleri içermeyecek kadar küçük.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

C çağırısı

```
MQMHBUF (Hconn, Hmsg, &MsgHBufOpts, &Name, &MsgDesc, BufferLength, Buffer,  
&DataLength, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

Kullanım notları

MQMHBUF, ileti tanıtıcısını arabelleğe dönüştürür.

Bunu, ileti özelliği API ' lerini kullanarak belirli özelliklere erişmek için bir MQGET API çıkışıyla kullanabilir ve daha sonra bunları, ileti tanıtıcıları yerine MQRFH2 üstbilgilerini kullanmak üzere tasarlanmış bir uygulamaya geri aktarabilirsiniz.

Bu çağrı, ileti özelliklerini arabellekten ileti tanıtıcısına ayırtırmak için kullanabileceğiniz MQBUFMH çağrısının tersidir.

COBOL çağrısı

```
CALL 'MQMHBUF' USING HCONN, HMSG, MSGHBUFOPTS, NAME, MSGDESC,  
BUFFERLENGTH, BUFFER, DATALENGTH, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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 CMQCHRUV.  
** 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.
```

PL/I çağrısı

```
call MQMHBUF (Hconn, Hmsg, MsgHBufOpts, Name, MsgDesc, BufferLength, Buffer,  
DataLength, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL MQMHBUF, (HCONN, HMSG, MSGHBUFOPTS, NAME, MSGDESC, BUFFERLENGTH,  
BUFFER, DATALENGTH, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

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-Nesneyi aç

MQOPEN çağrısı bir nesneye erişim sağlar.

Aşağıdaki nesne tipleri geçerlidir:

- Kuyruk (dağıtım listeleri dahil)
- Ad listesi
- Süreç tanımlaması
- Kuyruk yöneticisi
- Konu

Sözdizimi

MQOPEN (*Hconn, ObjDesc, Seçenekler, Hobj, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıttıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. Hconn değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

 z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve Hconn için belirtilen şu değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıttıcısı.

ObjDesc

Tip: MQOD-giriş/çıkış

Bu, açılacak nesneyi tanımlayan bir yapıdır; ayrıntılar için bkz. [“MQOD-Nesne tanımlayıcı” sayfa 476](#) .

ObjDesc parametresindeki ObjectName alanı bir model kuyruğunun adıysa, dinamik bir yerel kuyruk model kuyruğunun öznitelikleriyle yaratılır; bu, **Options** değiştirgesinde belirttiğiniz seçeneklerle gerçekleşir. MQOPEN çağrısıyla döndürülen Hobj kullanılarak yapılan sonraki işlemler, model kuyruğunda değil, yeni dinamik kuyrukta gerçekleştirilir. Bu, MQINQ ve MQSET çağrıları için bile geçerlidir. **ObjDesc** değiştirgesindeki model kuyruğunun adı, yaratılan dinamik kuyruğun adıyla değiştirilir. Dinamik kuyruğun tipi, model kuyruğunun **DefinitionType** özniteliğinin değerine göre belirlenir (bkz. [“Kuyruklara ilişkin öznitelikler” sayfa 826](#)). Dinamik kuyruklar için geçerli kapatma seçenekleriyle ilgili bilgi için MQCLOSE çağrısının tanımına bakın.

Seçenekler

Tip: MQLONG-input

Aşağıdaki seçeneklerden en az birini belirtmelisiniz:

- MQOO_GÖZ AT
- MQOO_INPUT_ * (bunlardan yalnızca biri)

- MQOO_INQUIRE
- MQOO_ÇIKIŞI
- MQOO_SET
- MQOO_BIND_* (bunlardan yalnızca biri)

Bu seçeneklerin ayrıntıları için aşağıdaki çizelgeye bakın; diğer seçenekler gerektiği gibi belirtilebilir. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyorsa). Geçerli olmayan birleşimler not edildi; diğer tüm birleşimler geçerlidir. Yalnızca ObjDesc tarafından belirtilen nesne tipi için geçerli olan seçeneklere izin verilir.

Çizelge 553. Kuyruklar ve konular için geçerli MQOPEN seçenekleri						
Seçenek	Diğer ad ¹	Yerel ve Model	Uzak	Yerel Olmayan Küme	Dağıtım listesi	Konu
<u>MQOO_INPUT_AS_Q_DEF</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_INPUT_SHARED</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_INPUT_EXCLUSIVE</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_OUTPUT</u>	Evet	Evet	Evet	Evet	Evet	Evet
<u>MQOO_BROWSE</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_CO_OP</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_INQUIRE</u>	Evet	Evet	<u>2</u>	Evet	Hayır	Hayır
<u>MQOO_SET</u>	Evet	Evet	<u>2</u>	Hayır	Hayır	Hayır
<u>MQOO_BIND_ON_OPEN</u> ³	Evet	Evet	Evet	Evet	Evet	Hayır
<u>MQOO_BIND_NOT_FIXED</u> ³	Evet	Evet	Evet	Evet	Evet	Hayır
<u>MQOO_BIND_ON_GROUP</u> ³	Evet	Evet	Evet	Evet	Evet	Hayır
<u>MQOO_BIND_AS_Q_DEF</u> ³	Evet	Evet	Evet	Evet	Evet	Hayır
<u>MQOO_SAVE_ALL_CONTEXT</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_PASS_IDENTITY_CONTEXT</u>	Evet	Evet	Evet	Evet	Evet	<u>4</u>
<u>MQOO_PASS_ALL_CONTEXT</u>	Evet	Evet	Evet	Evet	Evet	Evet
<u>MQOO_SET_IDENTITY_CONTEXT</u>	Evet	Evet	Evet	Evet	Evet	<u>4</u>
<u>MQOO_SET_ALL_CONTEXT</u>	Evet	Evet	Evet	Evet	Evet	Evet
<u>MQOO_NO_READ_ÖNDEN</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_READ_ÖNDEN</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_READ_AHEAD_AS_Q_DEF</u>	Evet	Evet	Hayır	Hayır	Hayır	Hayır
<u>MQOO_ALTERNATE_USER_AUTHORITY</u>	Evet	Evet	Evet	Evet	Evet	Evet
<u>MQOO_FAIL_IF QUIESCING</u>	Evet	Evet	Evet	Evet	Evet	Evet
<u>MQOO_RESOLVE_LOCAL_Q</u>	Evet	Evet	Evet	Evet	Hayır	Hayır
<u>MQOO_RESOLVE_LOCAL_TOPIC</u>	Hayır	Hayır	Hayır	Hayır	Hayır	Evet
<u>MQOO_NO_MULTICAST</u>	Hayır	Hayır	Hayır	Hayır	Hayır	Evet

Notlar:

1. Diğer adlara ilişkin seçeneklerin geçerliliği, diğer adın çözümleneceği kuyruğa ilişkin seçeneğin geçerliliğine bağlıdır.
2. Bu seçenek yalnızca uzak kuyruğun yerel tanımlaması için geçerlidir.

3. Bu seçenek herhangi bir kuyruk tipi için belirlenebilir, ancak kuyruk bir küme kuyruğu değilse yoksayılr. Ancak **DefBind** kuyruk özniteliği, diğer ad kuyruğu bir kümede olmasa da temel kuyruğu geçersiz kılar.
4. Bu öznitelikler bir konuyla birlikte kullanılabilir, ancak herhangi bir aboneye gönderilen bağlam alanlarını değil, yalnızca alıkonan ileti için ayarlanan bağlamı etkiler.

Erişim seçenekleri: Aşağıdaki seçenekler, nesne üzerinde gerçekleştirilebilecek işlemlerin tipini denetler:

MQOO_INPUT_AS_Q_DEF

Kuyruk tanımlı varsayılan değeri kullanarak ileti almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Erişim tipi, **DefInputOpenOption** kuyruk özniteliğinin değerine bağlı olarak paylaşılır ya da dışlayıcıdır; ayrıntılar için bkz. [“Kuyruklara ilişkin öznitelikler”](#) sayfa 826 .

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir.

MQOO_INPUT_SHARED

Paylaşılan erişimi olan iletileri almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Kuyruk şu anda MQOO_INPUT_SHARED olan bu uygulama ya da başka bir uygulama tarafından açıksa, ancak kuyruk şu anda MQOO_INPUT_EXCLUSIVE ile açıksa, MQRC_OBJECT_IN_USE neden koduyla başarısız olursa çağrı başarılı olabilir.

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir.

MQOO_INPUT_EXCLUSIVE

Dışlayıcı erişimi olan iletileri almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Kuyruk şu anda herhangi bir tip giriş için bu ya da başka bir uygulama tarafından açıksa (MQOO_INPUT_SHARED ya da MQOO_INPUT_EXCLUSIVE), çağrı MQRC_OBJECT_IN_USE neden koduyla başarısız olur.

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir.

MQOO_ÇIKIŞI

İletileri koymak için kuyruğu açın ya da iletileri yayınlamak için bir konu ya da konu dizesini açın.

Kuyruk ya da konu, sonraki MQPUT çağrılarıyla kullanılmak üzere açılır.

Bu seçenekle bir MQOPEN çağrısı, **InhibitPut** kuyruk özniteliği MQQA_PUT_INÇEKINGEN olarak ayarlanmış olsa da (öznitelik bu değere ayarlandığında sonraki MQPUT çağrıları başarısız olsa da) başarılı olabilir.

Bu seçenek, dağıtım listeleri ve konular da içinde olmak üzere tüm kuyruk tipleri için geçerlidir.

Aşağıdaki notlar bu seçenekler için geçerlidir:

- Bu seçeneklerden yalnızca biri belirtilebilir.
- Bu seçeneklerden birine sahip bir MQOPEN çağrısı, **InhibitGet** kuyruk özniteliği MQQA_GET_INÇEKINGEN olarak ayarlanırsa da (öznitelik bu değere ayarlandığında sonraki MQGET çağrıları başarısız olsa da) başarılı olabilir.
- Kuyruk paylaşılabilir olarak tanımlanmazsa (yani, **Shareability** kuyruk özniteliği MQQA_NOT_SHAREABLE değerine sahipse), kuyruğu paylaşılabilir erişim için açma girişimleri, kuyruğu dışlayıcı erişimle açma girişimleri olarak kabul edilir.
- Bu seçeneklerden biriyle bir diğer ad kuyruğu açılırsa, dışlayıcı kullanım sınaması (ya da başka bir uygulamanın özel kullanımı olup olmadığı), diğer adın çözümleneceği temel kuyruğa karşı olur.

- **ObjectQMgrName** bir kuyruk yöneticisi diğer adının adıysa bu seçenekler geçerli değildir; kuyruk yöneticisi diğer adı için kullanılan uzak kuyruğun yerel tanımında **RemoteQMgrName** özniteliğinin değeri yerel kuyruk yöneticisinin adı olsa bile bu geçerlidir.

MQOO_GÖZ AT

İletilere göz atmak için kuyruğu açın.

Kuyruk, aşağıdaki seçeneklerden biriyle sonraki MQGET çağrılarıyla kullanılmak üzere açılır:

- MQGMO_BROWSE_FIRST
- MQGMO_BROWSE_NEXT
- MQGMO_BROWSE_MSG_UNDER_CURSOR

Kuyruk şu anda MQOO_INPUT_EXCLUSIVE için açık olsa da buna izin verilir. MQOO_BROWSE seçeneğiyle bir MQOPEN çağrısı bir göz atma imleci oluşturur ve mantıksal olarak kuyruktaki ilk iletinin önüne konumlandırır; ek bilgi için [MQGMO-Options alanı](#) konusuna bakın.

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir. ObjectQMgrName bir kuyruk yöneticisi diğer adının adıysa da geçerli değildir; uzak kuyruk yöneticisi diğer adı için kullanılan uzak kuyruk yöneticisinin yerel tanımındaki **RemoteQMgrName** özniteliğinin değeri yerel kuyruk yöneticisinin adı olsa da bu geçerlidir.

MQOO_CO_OP

Tutamaçlar kümesinin işbirliği üyesi olarak açın.

Bu seçenek yalnızca MQOO_BROWSE seçeneğiyle geçerlidir. MQOO_BROWSE, MQOPEN olmadan belirtilirse, MQRC_OPTIONS_ERROR ile döndürülür.

Döndürülen tanıtıcı, aşağıdaki seçeneklerden biriyle sonraki MQGET çağrılarını için işbirliği içeren tanıtıcı kümesinin üyesi olarak kabul edilir:

- MQGMO_MARK_BROWSE_CO_OP
- MQGMO_UNMARKED_BROWSE_MSG
- MQGMO_UNMARK_BROWSE_CO_OP

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir.

MQOO_INQUIRE

Öznitelikleri sorgulamak için nesneyi açın.

Kuyruk, ad listesi, süreç tanımlaması ya da kuyruk yöneticisi sonraki MQINQ çağrılarıyla kullanılmak üzere açılır.

Bu seçenek, dağıtım listeleri dışındaki tüm nesne tipleri için geçerlidir. ObjectQMgrName bir kuyruk yöneticisi diğer adının adıysa geçerli değildir; kuyruk yöneticisi için kullanılan uzak kuyruğun yerel tanımında **RemoteQMgrName** özniteliğinin değeri yerel kuyruk yöneticisinin adı olsa da bu geçerlidir.

MQOO_SET

Öznitelikleri ayarlamak için kuyruğu açın.

Kuyruk, sonraki MQSET çağrılarıyla kullanılmak üzere açılır.

Bu seçenek, dağıtım listeleri dışındaki tüm kuyruk tipleri için geçerlidir. ObjectQMgrName uzak bir kuyruğun yerel tanımının adıdır; kuyruk yöneticisi için kullanılan uzak kuyruğun yerel tanımında **RemoteQMgrName** özniteliğinin değeri yerel kuyruk yöneticisinin adı olsa bile bu geçerlidir.

Bağ tanımlama seçenekleri: Açılmakta olan nesne bir küme kuyruğuyken aşağıdaki seçenekler geçerlidir; bu seçenekler, kuyruk tanıtıcısının küme kuyruğunun bir eşgörünümüne bağlanmasını denetler:

MQOO_BIND_ON_OPEN

Yerel kuyruk yöneticisi, kuyruk açıldığında kuyruk tanıtıcısını hedef kuyruğun bir eşgörünümüne bağlar. Sonuç olarak, bu tanıtıcı kullanılarak konan tüm iletiler hedef kuyruğun aynı eşgörünümüne ve aynı rotaya gönderilir.

Bu seçenek yalnızca kuyruklar için geçerlidir ve yalnızca küme kuyruklarını etkiler. Küme kuyruğu olmayan bir kuyruk için belirtildiyse, seçenek yoksayılr.

MQOO_BIND_NOT_FIXED

Bu, yerel kuyruk yöneticisinin kuyruk tanıtıcısını hedef kuyruğun bir eşgörünümüne bağlamasını durdurur. Sonuç olarak, bu tanıtıcıyı kullanan ardışık MQPUT çağrıları, iletileri hedef kuyruğun farklı eşgörünümüne ya da aynı örneğe, ancak farklı rotalarla gönderir. Ayrıca, seçilen yönetim ortamının daha sonra yerel kuyruk yöneticisi, uzak kuyruk yöneticisi ya da bir ileti kanalı aracısı (MCA) tarafından ağ koşullarına göre değiştirilmesini sağlar.

Not: Bir hareketi tamamlamak için bir dizi iletiyi değiş tokuş etmesi gereken istemci ve sunucu uygulamaları, DefBind değeri MQBND_BIND_NOT_FIXED olduğunda MQOO_BIND_NOT_FIXED (ya da MQOO_BIND_AS_Q_DEF) kullanmamalıdır; serideki ardışık iletiler sunucu uygulamasının farklı yönetim ortamlarına gönderilebilir.

Bir küme kuyruğu için MQOO_BROWSE ya da MQOO_INPUT_* seçeneklerinden biri belirtilirse, kuyruk yöneticisi küme kuyruğunun yerel eşgörünümünü seçmeye zorlanır. Sonuç olarak, MQOO_BIND_NOT_FIXED belirtilse bile, kuyruk tanıtıcısının bağ tanımı düzeltilmiştir.

MQOO_BIND_NOT_FIXED ile MQOO_INQUIRE belirtilirse, bu tanıtıcıyı kullanan ardışık MQINQ çağrıları küme kuyruğunun farklı eşgörünümünü sorgulayabilir; ancak, genellikle tüm eşgörünümlerin öznitelik değerleri aynıdır.

MQOO_BIND_NOT_FIXED yalnızca kuyruklar için geçerlidir ve yalnızca küme kuyruklarını etkiler. Küme kuyruğu olmayan bir kuyruk için belirtildiyse, seçenek yoksayılr.

MQOO_BIND_ON_GROUP

Bir uygulamanın, bir grup iletinin aynı hedef örneğe ayrılmasını istemesini sağlar.

Bu seçenek yalnızca kuyruklar için geçerlidir ve yalnızca küme kuyruklarını etkiler. Küme kuyruğu olmayan bir kuyruk için belirtildiyse, seçenek yoksayılr.

MQOO_BIND_AS_Q_DEF

Yerel kuyruk yöneticisi, kuyruk tanıtıcısını **DefBind** kuyruk özniteliğinin tanımladığı şekilde bağlar. Bu özniteliğin değeri MQBND_BIND_ON_OPEN, MQBND_BIND_NOT_FIXED ya da MQBND_BIND_ON_GROUP.

MQOO_BIND_AS_Q_DEF, MQOO_BIND_ON_OPEN, MQOO_BIND_NOT_FIXED ya da MQOO_BIND_ON_GROUP belirtilmediğinde varsayılan değerdir.

MQOO_BIND_AS_Q_DEF program belgeleri. Bu seçeneğin diğer iki bağ tanımlama seçeneğinden biriyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bağlam seçenekleri: Aşağıdaki seçenekler ileti bağlamının işlenmesini denetler:

MQOO_SAVE_ALL_CONTEXT

Bağlam bilgileri bu kuyruk tanıtıcısıyla ilişkilendirilir. Bu bilgi, bu tanıtıcı kullanılarak alınan tüm iletilerin bağlamından ayarlanır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

Bu bağlam bilgileri, MQPUT ya da MQPUT1 çağrıları kullanılarak kuyruğa konan bir iletiye iletilebilir. "MQPMO-İleti koyma seçenekleri" sayfa 496 içinde açıklanan MQPMO_PASS_IDENTITY_CONTEXT ve MQPMO_PASS_ALL_CONTEXT seçeneklerine bakın.

Bir ileti başarıyla alınıncaya kadar, bağlam kuyruğa konan bir iletiye geçirilemez.

MQGMO_BROWSE_* göz atma seçeneklerinden biri kullanılarak alınan bir iletinin bağlam bilgileri saklanmaz (ancak, **MsgDesc** değiştirgesindeki bağlam alanları bir göz atma işleminden sonra ayarlanır).

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir. MQOO_INPUT_* seçeneklerinden biri belirtilmelidir.

MQOO_PASS_IDENTITY_CONTEXT

Bu, kuyruğa bir ileti konduğunda **PutMsgOpts** değiştirgesinde MQPMO_PASS_IDENTITY_CONTEXT seçeneğinin belirtilmesini sağlar; bu, iletiye MQOO_SAVE_ALL_CONTEXT seçeneğiyle açılan bir giriş kuyruğundaki kimlik bağlamı bilgilerini verir. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

MQOO_OUTPUT seçeneği belirtilmelidir.

Bu seçenek, dağıtım listeleri de içinde olmak üzere, tüm kuyruk tipleri için geçerlidir.

MQOO_PASS_ALL_CONTEXT

Bu, kuyruğa bir ileti konduğunda **PutMsgOpts** değiştirgesinde MQPMO_PASS_ALL_CONTEXT seçeneğinin belirtilmesini sağlar; bu, iletiye, MQOO_SAVE_ALL_CONTEXT seçeneğiyle açılan bir giriş kuyruğundaki tanıtıcı ve kaynak bağlam bilgilerini verir. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

Bu seçenek, belirtilmesi gerekmeyen MQOO_PASS_IDENTITY_CONTEXT değerini belirtir. MQOO_OUTPUT seçeneği belirtilmelidir.

Bu seçenek, dağıtım listeleri de içinde olmak üzere, tüm kuyruk tipleri için geçerlidir.

MQOO_SET_IDENTITY_CONTEXT

Bu, kuyruğa bir ileti konduğunda **PutMsgOpts** değiştirgesinde MQPMO_SET_IDENTITY_CONTEXT seçeneğinin belirtilmesini sağlar; bu, iletiye MQPUT ya da MQPUT1 çağrısında belirtilen **MsgDesc** değiştirgesinde bulunan kimlik bağlamı bilgilerini verir. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

Bu seçenek, belirtilmesi gerekmeyen MQOO_PASS_IDENTITY_CONTEXT değerini belirtir. MQOO_OUTPUT seçeneği belirtilmelidir.

Bu seçenek, dağıtım listeleri de içinde olmak üzere, tüm kuyruk tipleri için geçerlidir.

MQOO_SET_ALL_CONTEXT

Bu, kuyruğa bir ileti konduğunda **PutMsgOpts** değiştirgesinde MQPMO_SET_ALL_CONTEXT seçeneğinin belirtilmesini sağlar; bu, iletiye MQPUT ya da MQPUT1 çağrısında belirtilen **MsgDesc** değiştirgesinde belirtilen tanıtıcı ve kaynak bağlam bilgilerini verir. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

Bu seçenek, belirtilmesi gerekmeyen aşağıdaki seçenekleri belirtir:

- MQOO_PASS_IDENTITY_CONTEXT
- MQOO_PASS_ALL_CONTEXT
- MQOO_SET_IDENTITY_CONTEXT

MQOO_OUTPUT seçeneği belirtilmelidir.

Bu seçenek, dağıtım listeleri de içinde olmak üzere, tüm kuyruk tipleri için geçerlidir.

Önden okuma seçenekleri:

MQOO_READ_ÖNCEDEN ile MQOPEN ' i çağırdığınızda, IBM MQ istemcisi belirli koşullar karşılandığında yalnızca önden okuma seçeneğini etkinleştirir. Bu koşullar şunlardır:

- İstemci uygulaması derlenmeli ve iş parçacıklı IBM MQ MQI istemcisi kitaplıklarına bağlanmalıdır.
- İstemci kanalı TCP/IP protokolünü kullanıyor olmalıdır
- Kanal, istemci ve sunucu kanalı tanımlarında sıfır dışında bir SharingConversations (SHARECNV) ayarı içermelidir.

Aşağıdaki seçenekler, kalıcı olmayan iletilerin bir uygulama tarafından istenmeden önce istemciye gönderilip gönderilmeyeceğini denetler. Aşağıdaki notlar, önden okuma seçenekleri için geçerlidir:

- Bu seçeneklerden yalnızca biri belirtilebilir.
- Bu seçenekler yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir. Bunlar uzak kuyruklar, dağıtım listeleri, konular ya da kuyruk yöneticileri için geçerli değildir.
- Bu seçenekler yalnızca, MQOO_INQUIRE ya da MQOO_SET ile bu seçeneklerin belirtilmesi bir hata olmasa da, MQOO_BROWSE, MQOO_INPUT_SHARED ve MQOO_INPUT_EXCLUSIVE seçeneklerinden biri de belirtilirse geçerlidir.
- Uygulama IBM MQ istemcisi olarak çalışmıyorsa, bu seçenekler yoksayılr.

MQOO_NO_READ_ONDEN

Kalıcı olmayan iletiler, bir uygulama istemeden önce istemciye gönderilmez.

MQOO_READ_ONDEN

Kalıcı olmayan iletiler, bir uygulama istemeden önce istemciye gönderilir.

MQOO_READ_AHEAD_AS_Q_DEF

Önden okuma davranışı, açılmakta olan kuyruğun varsayılan önden okuma özneliğine göre belirlenir. Bu varsayılan değerdir.

Diğer seçenekler: Aşağıdaki seçenekler yetki denetimini, kuyruk yöneticisi susturulduğunda ne olacağını, yerel kuyruk adının çözümlenip çözümlenmeyeceğini ve çok hedefli olup olmayacağını denetler:


MQOO_ALTERNATE_USER_AUTHORITY

ObjDesc deęiştirgesindeki *AlternateUserId* alanı, bu MQOPEN çağrısının geçerliliğini denetlemek için kullanılacak bir kullanıcı kimliği içerir. Çaęrı, yalnızca bu *AlternateUserId* ' in belirtilen erişim seçenekleriyle nesneyi açma yetkisi varsa, uygulamanın çalıştığı kullanıcı kimliğinin bunu yapma yetkisi olup olmadığına bakılmaksızın başarılı olabilir. Bu, belirtilen bağlam seçenekleri için geçerli değildir; ancak, her zaman uygulamanın çalıştığı kullanıcı kimliğiyle karşılaştırılarak denetlenir.

Bu seçenek, tüm nesne tipleri için geçerlidir.

MQOO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturma durumundaysa MQOPEN çağrısı başarısız olur.

 z/OS sistemlerinde, bir CICS ya da IMS uygulaması için bu seçenek, bağlantı susturma durumundaysa MQOPEN çağrısına da başarısız olmasına neden olur.

Bu seçenek, tüm nesne tipleri için geçerlidir.

Müşteri kanalları hakkında bilgi için bkz. [IBM MQ MQI clients](#).

MQOO_RESOLVE_LOCAL_Q

MQOD yapısındaki ResolvedQName öęesini, açılan yerel kuyruęun adıyla doldurun. Benzer şekilde, ResolvedQMgrAdı, yerel kuyruęu barındıran yerel kuyruk yöneticisinin adıyla doldurulur. MQOD yapısı Sürüm 3 'ten küçükse, MQOO_RESOLVE_LOCAL_Q yoksayılr ve hata döndürölmez.

Yerel bir kuyruk, diğer ad ya da model kuyruęu açıldığında yerel kuyruk her zaman döndürölür; ancak, örneęin, uzak kuyruk ya da yerel olmayan bir küme kuyruęu MQOO_RESOLVE_LOCAL_Q seçeneęi olmadan açıldığında; ResolvedQName ve ResolvedQMgrAdı RemoteQName ve RemoteQMgradı uzak kuyruk tanımlamasında ya da benzer şekilde seçilen uzak küme kuyruęuyla birlikte bulunduęunda bu durum olmaz.

Açılırken MQOO_RESOLVE_LOCAL_Q belirtirseniz (örneęin, uzak bir kuyruk), iletilerin konduęu iletim kuyruęudur (ResolvedQName). ResolvedQMgrAdı, iletim kuyruęunu barındıran yerel kuyruk yöneticisinin adıyla doldurulur.

Bir kuyruкта göz atma, giriş ya da çıkış için yetkiniz varsa, MQOPEN çağrısında bu işareti belirtmek için gereken yetkiye sahipsiniz. Özel bir yetkiye gerek yoktur.

Bu seçenek yalnızca kuyruklar ve kuyruk yöneticileri için geçerlidir.

MQOO_RESOLVE_LOCAL_TOPIC

MQOD yapısındaki ResolvedQName ögesini, açılan denetim konusunun adıyla doldurun.

MQOO_NO_MULTICAST

Yayın iletileri çoklu yayın kullanılarak gönderilmez.

Bu seçenek yalnızca MQOO_OUTPUT seçeneğiyle geçerlidir. MQOO_OUTPUT olmadan belirtilirse, MQOPEN MQRC_OPTIONS_ERROR ile döndürülür.

Bu seçenek yalnızca bir konu için geçerlidir.

Hobj.

Tip: MQHOBJ-output

Bu tanıtıcı, nesne için oluşturulan erişimi gösterir. Nesne üzerinde çalışan sonraki IBM MQ çağrılarında belirtilmelidir. MQCLOSE çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde geçerli olmaktan çıkar.

Döndürülen nesne tanıtıcısının kapsamı, çağrıda belirlenen bağlantı tanıtıcısının kapsamıyla aynı. Tanıtıcı kapsamıyla ilgili bilgi için [MQCONN-Hconn değiştirgesi](#) konusuna bakın.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_MULTIPLE_REASONS

(2136, X'858 ') Birden çok neden kodu döndürüldü.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ALIAS_BASE_Q_TYPE_HATA

(2001, X'7D1') Diğer ad temel kuyruğu geçerli bir tip değil.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

(2183, X'887 ') API çıkışı yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CF_NOT_YOK
(2345, X' 929 ') Baęlama olanaęı kullanılmıyor.

MQRC_CF_STRUC_AUTH_FAILED
(2348, X'92C') Coupling-facility yapısı yetkilendirme denetimi başarısız oldu.

MQRC_CF_STRUC_HATA
(2349, X'92D') Baęlama olanaęı yapısı geęerli deęil.

MQRC_CF_STRUC_FAILED
(2373, X' 945 ') Baęlama tesisi yapısı başarısız oldu.

MQRC_CF_STRUC_IN_USE
(2346, X'92A') Coupling-facility yapısı kullanımda.

MQRC_CF_STRUC_LIST_HDR_IN_USE
(2347, X'92B') Coupling-facility yapısı liste üstbilgisi kullanımda.

MQRC_CICS_WAIT_FAILED
(2140, X'85C') Bekleme isteęi CICStarafından reddedildi.

MQRC_CLUSTER_EXIT_ERROR (Çıkış)
(2266, X'8DA') Küme iş yükü çıkışı başarısız oldu.

MQRC_CLUSTER_PUT_INENGELLEME
(2268, X'8DC') Kümedeki tüm kuyruklar için çağrılarını engelleme.

MQRC_CLUSTER_RESOLUTION_HATA
(2189, X'88D') Küme adı çözümlemesi başarısız oldu.

MQRC_CLUSTER_RESOURCE_ERROR
(2269, X'8DD') Küme kaynaęı hatası.

MQRC_CONNECTION_BROKEN
(2009, X'7D9') Kuyruk yöneticisine baęlantı kaybedildi.

MQRC_CONNECTION_NOT_YETKILI
(2217, X'8A9') Baęlantı için yetkili deęil.

MQRC_CONNECTION QUIESCING
(2202, X'89A') Baęlantı susturuyor.

MQRC_CONNECTION_DURDURULUYOR
(2203, X'89B') Baęlantı kapatılıyor.

MQRC_DB2_NOT_AVAILABLE
(2342, X' 926 ') Db2 altsistemi kullanılmıyor.

MQRC_DEF_XMIT_Q_TYPE_HATA
(2198, X'896 ') Varsayılan iletim kuyruęu yerel deęil.

MQRC_DEF_XMIT_Q_USAGE_HATA
(2199, X'897 ') Varsayılan iletim kuyruęu kullanım hatası.

MQRC_DYNAMIC_Q_AD_HATASI
(2011, X'7DB') Dinamik kuyruęun adı geęersiz.

MQRC_HANDLE_NOT_YOK
(2017, X'7E1') Baęka tanıtıcı yok.

MQRC_HCONN_HATA
(2018, X'7E2') Baęlantı tanıtıcısı geęersiz.

MQRC_HOBJ_HATA
(2019, X'7E3') Nesne tanıtıcısı geęerli deęil.

MQRC_MULTIPLE_REASONS
(2136, X'858 ') Birden çok neden kodu döndürüldü.

MQRC_NAME_IN_USE
(2201, X'899 ') Ad kullanımda.

MQRC_NAME_NOT_VALID_FOR_TYPE
(2194, X'892 ') Nesne adı nesne tipi için geęerli deęil.

MQRN_NOT_YETKILI
(2035, X'7F3') Eriřim için yetkili deęil.

MQRN_OBJECT_ALREADY_EXISTS
(2100, X'834 ') Nesne var.

MQRN_OBJECT_HASARLI
(2101, X'835 ') Nesne zarar grd.

MQRN_OBJECT_IN_USE
(2042, X'7FA') Nesne zaten akıřan seeneklerle aık.

MQRN_OBJECT_LEVEL_UYUMSUZ
(2360, X' 938 ') Nesne dzeyi uyumlu deęil.

MQRN_OBJECT_NAME_ERROR
(2152, X'868 ') Nesne adı geersiz.

MQRN_OBJECT_NOT_UNIQUE
(2343, X' 927 ') Nesne benzersiz deęil.

MQRN_OBJECT_Q_MGR_NAME_ERROR
(2153, X'869 ') Nesne kuyruęu yneticisi adı geersiz.

MQRN_OBJECT_RECORDS_ERROR (RECORDS_HATASI)
(2155, X'86B') Nesne kayıtları geerli deęil.

MQRN_OBJECT_STRING_HATA
(2441, X'0989 ') Nesne dizgisi alanı geersiz

MQRN_OBJECT_TYPE_ERROR
(2043, X'7FB') Nesne tipi geersiz.

MQRN_OD_HATA
(2044, X'7FC') Nesne tanımlayıcı yapısı geerli deęil.

MQRN_OPTION_NOT_VALID_FOR_TYPE
(2045, X'7FD') Seenek, nesne tipi için geerli deęil.

MQRN_OPTIONS_HATA
(2046, X'7FE') Seenekler geerli deęil ya da tutarlı deęil.

MQRN_PAGESET_HATA
(2193, X'891 ') Sayfa kmesi veri kmesine eriřilirken hata oluřtu.

MQRN_PAGESET_FULL
(2192, X'890 ') Dıř depolama ortamı dolu.

MQRN_Q_SILINDI
(2052, X'804 ') Kuyruk silindi.

MQRN_Q_MGR_AD_HATASI
(2058, X'80A') Kuyruk yneticisi adı geerli deęil ya da bilinmiyor.

MQRN_Q_MGR_YOK
(2059, X'80B') Kuyruk yneticisi baęlantı için kullanılmıyor.

MQRN_Q_MGR_QUIESCING
(2161, X'871 ') Kuyruk yneticisi susturuyor.

MQRN_Q_MGR_DURDURULUYOR
(2162, X'872 ') Kuyruk yneticisi kapatılıyor.

MQRN_Q_TYPE_HATA
(2057, X'809 ') Kuyruk tipi geersiz.

MQRN_RECS_PRESENT_ERROR
(2154, X'86A') Mevcut kayıt sayısı geerli deęil.

MQRN_REMOTE_Q_NAME_ERROR
(2184, X'888 ') Uzak kuyruk adı geersiz.

MQRN_RESOURCE_PROBLEM
(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_RESPONSE_RECORDS_ERROR

(2156, X'86C') Yanıt kayıtları geçerli değil.

MQRC_SECURITY_HATA

(2063, X'80F') Güvenlik hatası oluştu.

MQRC_SELECTOR_SYNTAX_ERROR

2459 (X'099B') Bir MQOPEN, MQPUT1 ya da MQSUB çağrısı yayınlandı, ancak sözdizimi hatası içeren bir seçim dizgisi belirtildi.

MQRC_STOPPED_BY_CLUSTER_EXIT

(2188, X'88C') Çağrı küme iş yükü çıkışı tarafından reddedildi.

MQRC_STORAGE_MEDIUM_FULL

(2192, X'890 ') Dış depolama ortamı dolu.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_SUPPRESSED_BY_EXIT

(2109, X'83D') Çağrı çıkış programı tarafından engellendi.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

MQRC_UNKNOWN_ALIAS_BASE_Q

(2082, X'822 ') Bilinmeyen diğer ad temel kuyruğu.

MQRC_UNKNOWN_DEF_XMIT_Q

(2197, X'895 ') Bilinmeyen varsayılan iletim kuyruğu.

MQRC_UNKNOWN_OBJECT_NAME

(2085, X'825 ') Bilinmeyen nesne adı.

MQRC_UNKNOWN_OBJECT_Q_MGR

(2086, X'826 ') Bilinmeyen nesne kuyruk yöneticisi.

MQRC_UNKNOWN_REMOTE_Q_MGR

(2087, X'827 ') Bilinmeyen uzak kuyruk yöneticisi.

MQRC_UNKNOWN_XMIT_Q

(2196, X'894 ') Bilinmeyen iletim kuyruğu.

MQRC_YANLIŞ_CF_DÜZEYI

(2366, X'93E') Coupling-facility yapısı yanlış düzeyde.

MQRC_XMIT_Q_TYPE_HATA

(2091, X'82B') İletim kuyruğu yerel değil.

MQRC_XMIT_Q_USAGE_HATASI

(2092, X'82C') İletim kuyruğu yanlış kullanımla.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Genel kullanım notları


1. Açılan nesne aşağıdakilerden biridir:

- Bir kuyruk:
 - İletileri al ya da göz at (MQGET çağrısı kullanarak)
 - Koyma iletileri (MQPUT çağrılarını kullanarak)
 - Kuyruğun öznitelikleri hakkında bilgi alma (MQINQ çağrısının kullanılması)
 - Kuyruğun özniteliklerinin ayarlanması (MQSET çağrısının kullanılması)

Adı belirtilen kuyruk bir model kuyruğuysa, dinamik bir yerel kuyruk yaratılır. "[MQOPEN-Nesneyi aç](#)" sayfa 725içinde açıklanan **ObjDesc** parametresine bakın.

Dağıtım listesi, kuyruk listesini içeren özel bir kuyruk nesnesi tipidir. İletileri koymak için açılabilir, ancak iletileri almak ya da göz atmak ya da öznitelikleri sorgulamak ya da ayarlamak için açılmaz. Daha fazla ayrıntı için kullanım notu 8 'e bakın.

QSGDISP (GROUP) içeren bir kuyruk, MQOPEN ya da MQPUT1 çağrılarıyla kullanılmayacak özel bir kuyruk tanımlaması tipidir.

- Listedeki kuyrukların adlarını sorgulamak için bir ad listesi (MQINQ çağrısının kullanılması).
 - Süreç öznitelikleri (MQINQ çağrısı kullanılarak) hakkında bilgi almak için bir süreç tanımlaması.
 - Yerel kuyruk yöneticisinin öznitelikleri hakkında bilgi almak için kuyruk yöneticisi (MQINQ çağrısının kullanılması).
 - İleti yayınlamak için konu (MQPUT çağrısı kullanarak)
2. Bir uygulama aynı nesneyi birden çok kez açabilir. Her açık için farklı bir nesne tanıtıcısı döndürülür. Döndürülen her tanıtıcı, ilgili açma işlemi gerçekleştirildiği işlevler için kullanılabilir.
 3. Açılmakta olan nesne bir küme kuyruğundan başka bir kuyruksa, MQOPEN çağrısı sırasında yerel kuyruk yöneticisi içindeki tüm ad çözümlemesi gerçekleşir. Bu, aşağıdakileri içerebilir:
 - Uzak kuyruk yöneticisinin adına uzak kuyruk tanımının yerel tanımının çözülmesi ve kuyruğun uzak kuyruk yöneticisinde tanındığı ad
 - Uzak kuyruk yöneticisi adının yerel iletim kuyruğu adına çözülmesi
 -  Yalnızca z/OS üzerinde, uzak kuyruk yöneticisi adının IGQ aracı tarafından kullanılan paylaşılan iletim kuyruğunun adına çözülmesi (yalnızca yerel ve uzak kuyruk yöneticileri aynı kuyruk paylaşım grubuna aitse geçerlidir)
 - Temel kuyruğun ya da konu nesnesinin adına ilişkin diğer ad çözümlemesi.

Ancak, tanıtıcı için sonraki MQINQ ya da MQSET çağrılarının yalnızca açılmış olan adla ilişkili olduğunu ve ad çözüldükten sonra sonuçlanan nesneyle değil. Örneğin, açılan nesne bir diğer adsa, MQINQ çağrısının döndürdüğü öznitelikler, temel kuyruğun öznitelikleri ya da diğer adın çözümleneceği bir konu nesnesi değil, diğer adın öznitelikleridir.

Açılmakta olan nesne bir küme kuyruğuyorsa, MQOPEN çağrısı sırasında ad çözme gerçekleşebilir ya da daha sonraya ertelenebilir. Çözümün olduğu nokta, MQOPEN çağrısında belirtilen MQOO_BIND_* seçenekleriyle denetlenir:

- MQOO_BIND_ON_OPEN
- MQOO_BIND_NOT_FIXED
- MQOO_BIND_AS_Q_DEF
- MQOO_BIND_ON_GROUP

Küme kuyruklarına ilişkin ad çözümlemesi hakkında ek bilgi için [Ad çözünürlüğü](#) başlıklı konuya bakın.

4. MQOO_BROWSE seçeneğiyle bir MQOPEN çağrısı, nesne tanıtıcısını ve göz atma seçeneklerinden birini belirten MQGET çağrılarını kullanılmak üzere bir göz atma geçici çizelgesi oluşturur. Bu, kuyruğun içeriğini değiştirmeden taranmasını sağlar. Göz atarak bulunan bir ileti, MQGMO_MSG_UNDER_CURSOR seçeneği kullanılarak kuyruktan kaldırılabilir. Aynı kuyruk için birden çok MQOPEN isteği yayınlanarak, tek bir uygulama için birden çok göz atma imleçleri etkin olabilir.
5. Bir tetikleyici izleme programı tarafından başlatılan uygulamalara, uygulama başlatıldığında uygulamayla ilişkili kuyruğun adı geçirilir. Bu kuyruk adı, kuyruğu açmak için **ObjDesc** parametresinde belirtilebilir. Daha fazla ayrıntı için bkz. ["MQTMC2 -Tetikleyici ileti 2 \(karakter biçimi\)" sayfa 600](#) .

Önden okuma seçenekleri

MQOO_READ_ÖNCEDEN ile MQOPEN ' i çağırdığınızda, IBM MQ istemcisi belirli koşullar karşılandığında yalnızca önden okuma seçeneğini etkinleştirir. Bu koşullar şunlardır:

- İstemci uygulaması derlenmeli ve iş parçacıklı IBM MQ MQI istemcisi kitaplıklarına bağlanmalıdır.

- İstemci kanalı TCP/IP protokolünü kullanıyor olmalıdır
- Kanal, istemci ve sunucu kanalı tanımlarında sıfır dışında bir SharingConversations (SHARECNV) ayarı içermelidir.

Aşağıdaki notlar, önden okuma seçeneklerinin kullanımı için geçerlidir.

1. Önden okuma seçenekleri yalnızca, MQOO_BROWSE, MQOO_INPUT_SHARED ve MQOO_INPUT_EXCLUSIVE seçeneklerinden biri ve yalnızca biri belirtildiğinde geçerlidir. MQOO_INQUIRE ya da MQOO_SET seçenekleriyle önden okuma seçenekleri belirtilirse bir hata oluşmaz.
2. İlk MQGET çağrısında kullanılan seçenekler, önden okuma ile kullanım için desteklenmiyorsa, önden okuma istendiğinde etkinleştirilmez. Ayrıca, istemci önden okumayı desteklemeyen bir kuyruk yöneticisine bağlanırken önden okuma devre dışı bırakılır.
3. Uygulama IBM MQ istemcisi olarak çalışmıyorsa, önden okuma seçenekleri yoksayılr.

Küme kuyrukları

Aşağıdaki notlar küme kuyruklarının kullanımı için geçerlidir.

1. Bir küme kuyruğu ilk kez açıldığında ve yerel kuyruk yöneticisi tam havuz kuyruğu yöneticisi olmadığına, yerel kuyruk yöneticisi küme kuyruğuna ilişkin bilgileri tam havuz kuyruk yöneticisinden alır. Ağ meşgul olduğunda, yerel kuyruk yöneticisinin havuz kuyruk yöneticisinden gereken bilgileri alması birkaç saniye sürebilir. Sonuç olarak, MQOPEN çağrısı yayınlayan uygulamanın, denetim MQOPEN çağrısından dönmeden önce en çok 10 saniye beklemesi gerekebilir. Yerel kuyruk yöneticisi bu süre içinde küme kuyruğuna ilişkin gerekli bilgileri almazsa, çağrı MQRC_CLUSTER_RESOLUTION_ERROR neden koduyla başarısız olur.
2. Bir küme kuyruğu açıldığında ve kümede kuyruğun birden çok eşgörünümü olduğunda, açılan eşgörünüm MQOPEN çağrısında belirtilen seçeneklere bağlıdır:

- Belirtilen seçenekler aşağıdakilerden herhangi birini içerirse:
 - MQOO_GÖZ AT
 - MQOO_INPUT_AS_Q_DEF
 - MQOO_INPUT_EXCLUSIVE
 - MQOO_INPUT_SHARED
 - MQOO_SET

Açılan küme kuyruğunun yönetim ortamı yerel yönetim ortamı olmalıdır. Kuyruğun yerel eşgörünümü yoksa, MQOPEN çağrısı başarısız olur.

- Belirtilen seçenekler önceden açıklanan seçeneklerden hiçbirini içermezse, ancak aşağıdakilerden birini ya da her ikisini de içerirse:
 - MQOO_INQUIRE
 - MQOO_ÇIKIŞI

Açılan yönetim ortamı yerel yönetim ortamı (varsa) ve uzak yönetim ortamı (CLWLUSEQ varsayılanları kullanılıyorsa). Ancak, kuyruk yöneticisi tarafından seçilen yönetim ortamı bir küme iş yükü çıkışıyla (varsa) değiştirilebilir.


3. Kuyruk için bir abonelik varsa, ancak tam havuz tarafından onaylanmamışsa, nesne kümede yoktur ve çağrı MQRC_OBJECT_NAME neden koduyla başarısız olur.

Küme kuyruklarıyla ilgili ek bilgi için [Küme kuyrukları](#) başlıklı konuya bakın.

Dağıtım listeleri

Aşağıdaki notlar dağıtım listelerinin kullanımı için geçerlidir.

Dağıtım listeleri aşağıdaki ortamlarda desteklenir:

-  AIX

-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağı IBM MQ MQI clients için.

1. Bir dağıtım listesi açılırken MQOD yapısındaki alanlar aşağıdaki gibi ayarlanmalıdır:

- Version , MQOD_VERSION_2 ya da daha büyük olmalıdır.
- ObjectType , MQOT_Q olmalıdır.
- ObjectName boş değerli dizgi ya da boş değerli dizgi olmalıdır.
- ObjectQMgrName boş değerli dizgi ya da boş değerli dizgi olmalıdır.
- RecsPresent sıfırdan büyük olmalıdır.
- ObjectRecOffset ve ObjectRecPtr değerlerinden biri sıfır, diğeri sıfır olmayan olmalıdır.
- ResponseRecOffset ve ResponseRecPtr değerlerinden en fazla biri sıfır dışında olabilir.
- ObjectRecOffset ya da ObjectRecPtr tarafından adreslenen RecsPresent nesne kayıtları olmalıdır. Nesne kayıtları, açılacak hedef kuyrukların adlarına ayarlanmalıdır.
- ResponseRecOffset ve ResponseRecPtr değerlerinden biri sıfır değilse, RecsPresent yanıt kaydı olmalıdır. Çağrı MQRC_MULTIPLE_REASON ile tamamlanırsa kuyruk yöneticisi tarafından ayarlanır.

version-2 MQOD, RecsPresent ' in sıfır olduğundan emin olarak, dağıtım listesinde olmayan tek bir kuyruğu açmak için de kullanılabilir.

2. **Options** değıştirgesinde yalnızca aşağıdaki açma seçenekleri geçerlidir:

- MQOO_ÇIKIŞI
- MQOO_PAROLASI_ * _BAĞLAM
- MQOO_AYARI_ * _BAĞLAM
- MQOO_ALTERNATE_USER_AUTHORITY
- MQOO_FAIL_IF_QUIESCING

3. Dağıtım listesindeki hedef kuyruklar yerel, diğere ad ya da uzak kuyruklar olabilir, ancak bunlar model kuyrukları olamaz. Bir model kuyruğu belirtilirse, bu kuyruk açılmaz; neden kodu MQRC_Q_TYPE_ERROR. Ancak bu, listedeki diğere kuyrukların başarıyla açılmasını önlemez.

4. Tamamlanma kodu ve neden kodu değıştirgeleri aşağıdaki gibi ayarlanır:

- Dağıtım listesindeki kuyruklara ilişkin açma işlemleri aynı şekilde başarılı ya da başarısız olursa, tamamlanma kodu ve neden kodu parametreleri ortak sonucu tanımlayacak şekilde ayarlanır. Bu durumda MQRR yanıt kayıtları (uygulama tarafından sağlandıysa) ayarlanmaz.

Örneğin, her açma başarılı olursa, tamamlama kodu MQCC_OK olarak ayarlanır ve neden kodu MQRC_NONE olarak ayarlanır; kuyrukların hiçbiri var olmadığı için her açma başarısız olursa, parametreler MQCC_FAILED ve MQRC_UNKNOWN_OBJECT_NAME olarak ayarlanır.

- Dağıtım listesindeki kuyruklara ilişkin açma işlemleri aynı şekilde başarılı olmazsa ya da başarısız olursa:
 - En az bir açma başarılı olursa, tamamlama kodu parametresi MQCC_WARNING olarak ve tümü başarısız olursa MQCC_FAILED olarak ayarlanır.
 - Neden kodu parametresi MQRC_MULTIPLE_REASON olarak ayarlandı.
 - Yanıt kayıtları (uygulama tarafından sağlandıysa), dağıtım listesindeki kuyruklara ilişkin tek tek tamamlanma kodlarına ve neden kodlarına ayarlanır.

5. Bir dağıtım listesi başarıyla açıldığında, çağrı tarafından döndürülen Hobj tanıtıcısı, dağıtım listesindeki kuyruklara ileti koymak için sonraki MQPUT çağrılarında ve dağıtım listesine erişimi bırakmaya ilişkin bir MQCLOSE çağrısında kullanılabilir. Dağıtım listesi için tek geçerli kapanış seçeneğı MQCO_NONE.

MQPUT1 çağrısı bir iletiyi dağıtım listesine koymak için de kullanılabilir; listedeki kuyrukları tanımlayan MQOD yapısı, o çağrıda bir değiştirge olarak belirtilir.

6. Dağıtım listesinde başarıyla açılan her hedef, uygulamanın izin verilen tanıtıcı sayısı üst sınırını aşmışa denetlenirken ayrı bir tanıtıcı olarak sayılır (**MaxHandles** kuyruk yöneticisi özneliğine bakın). Dağıtım listesindeki iki ya da daha çok hedef aynı fiziksel kuyruğa çözüldüğünde bile bu geçerlidir. Bir dağıtım listesine ilişkin MQOPEN ya da MQPUT1 çağrısı, uygulamanın kullandığı tanıtıcı sayısının MaxHandlesdeğerini aşmasına neden olursa, çağrı başarısız olur; neden kodu MQRC_HANDLE_NOT_ALLOWED.
7. Başarıyla açılan her hedefin **OpenOutputCount** özneliğinin değeri bir artırıldı. Dağıtım listesindeki hedeflerden ikisi ya da daha fazlası aynı fiziksel kuyruğa çözümlenirse, o kuyruğun **OpenOutputCount** özneliği, dağıtım listesindeki o kuyruğa çözülecek hedeflerin sayısıyla artırılır.
8. Kuyruk tanımlamalarında, kuyruklar ayrı ayrı açıldığında (örneğin, çözüm yolundaki bir değişiklik), tanıtıcı geçersiz duruma gelmesine neden olacak herhangi bir değişiklik, dağıtım listesi tanıtıcılarının geçersiz olmasına neden olmaz. Ancak, dağıtım listesi tanıtıcısı sonraki bir MQPUT çağrısında kullanıldığında, o kuyruk için bir hata oluşmasına neden olur.
9. Dağıtım listesi tek bir hedef içerebilir.

Uzak kuyruklar

Aşağıdaki notlar uzak kuyrukların kullanımı için geçerlidir.

Uzak kuyruk, bu çağrıya ilişkin **ObjDesc** değiştirgesinde iki yoldan biriyle belirtilebilir.

- **ObjectName** için uzak kuyruğun yerel tanımının adını belirleyerek. Bu durumda **ObjectQMGrName** , yerel kuyruk yöneticisini ifade eder ve boşluk ya da (C programlama dilinde) boş bir dizgi olarak belirtilebilir.

Yerel kuyruk yöneticisi tarafından gerçekleştirilen güvenlik doğrulaması, kullanıcının uzak kuyruğun yerel tanımını açma yetkisinin olduğunu doğrular.

- Uzak kuyruk yöneticisi tarafından bilinen uzak kuyruğun adını **ObjectName** olarak belirterek. Bu durumda, **ObjectQMGrName** uzak kuyruk yöneticisinin adıdır.

Yerel kuyruk yöneticisi tarafından gerçekleştirilen güvenlik doğrulaması, kullanıcının ad çözme işleminden kaynaklanan iletim kuyruğuna ileti gönderme yetkisi olduğunu doğrular.

Her iki durumda da:

- Kullanıcının kuyruğa ileti koyma yetkisi olup olmadığını denetlemek için yerel kuyruk yöneticisi tarafından uzak kuyruk yöneticisine ileti gönderilmez.
- Uzak kuyruk yöneticisine bir ileti geldiğinde, uzak kuyruk yöneticisi iletiyi oluşturan kullanıcının yetkisi olmadığı için iletiyi reddedebilir.

Daha fazla bilgi için "[MQOD-Nesne tanımlayıcı](#)" sayfa 476 içinde açıklanan **ObjectName** ve **ObjectQMGrName** alanlarına bakın.

Nesneler

Durumu

Aşağıdaki notlar, MQOPEN kullanımının güvenlik yönleriyle ilgilidir.

Kuyruk yöneticisi, bir MQOPEN çağrısı yayınlandığında, uygulamanın çalıştırıldığı kullanıcı kimliğinin erişime izin verilmeden önce uygun yetki düzeyine sahip olduğunu doğrulamak için güvenlik denetimleri gerçekleştirir. Yetki denetimi, bir ad çözüldükten sonra ortaya çıkan, adı ya da adları değil, açılmakta olan nesnenin adı üzerinde yapılır.

Açılmakta olan nesne, bir konu nesnesini gösteren bir diğer ad kuyruğuysa, kuyruk yöneticisi, konu nesnesi doğrudan kullanılmış gibi bir güvenlik denetimi gerçekleştirmeden önce diğer ad kuyruğu adı üzerinde bir güvenlik denetimi gerçekleştirir.

Açılmakta olan nesne bir konu nesnesiyse, `ObjectName` ile ya da `ObjectString` kullanılarak (temel öge `ObjectName` ile ya da temel nesne olmadan), kuyruk yöneticisi güvenlik denetimini `ObjectName` içinde belirtilen konu nesnesinden alınan sonuçtaki konu dizgisini kullanarak ve gerekiyorsa, `ObjectString` içinde sağlanan nesne ile birleştirerek ve daha sonra, güvenlik denetimini gerçekleştirmek için konu ağacında bu noktada ya da üstünde en yakın konu nesnesini bularak gerçekleştirir. Bu, `ObjectName` içinde belirtilen konu nesnesiyle aynı olmayabilir.

Açılmakta olan nesne bir model kuyruğuysa, kuyruk yöneticisi hem model kuyruğunun adı, hem de yaratılan dinamik kuyruğun adı için tam güvenlik denetimi gerçekleştirir. Sonuçta ortaya çıkan dinamik kuyruk belirttik olarak açılırsa, dinamik kuyruğun adına göre başka bir kaynak güvenliği denetimi gerçekleştirilir.

z/OS z/OS' da, kuyruk yöneticisi güvenlik denetimlerini yalnızca güvenlik etkinleştirildiyse gerçekleştirir. Güvenlik denetimiyle ilgili daha fazla bilgi için [z/OS üzerinde güvenliği ayarlama](#) başlıklı konuya bakın.

Öznitelikler

Aşağıdaki notlar özniteliklerle ilgilidir.

Bir uygulamanın nesnesi açıkken nesnenin öznitelikleri değişebilir. Birçok durumda, uygulama bunu fark etmez, ancak belirli öznitelikler için kuyruk yöneticisi tanıtıcıyı artık geçerli değil olarak işaretler. Bu öznitelikler şunlardır:

- Nesnenin ad çözünürlüğünü etkileyen herhangi bir öznitelik. Bu, kullanılan açık seçeneklerden bağımsız olarak geçerlidir ve aşağıdakileri içerir:
 - Açık olan bir diğer ad kuyruğunun **BaseQName** özniteliğinde yapılan bir değişiklik.
 - Açık olan bir diğer ad kuyruğunun **TargetType** özniteliğinde yapılan bir değişiklik.
 - **RemoteQName** ya da **RemoteQMGrName** kuyruk özniteliklerinde, bu kuyruk için açık olan herhangi bir tanıtıcıya ya da kuyruk yöneticisi diğer adı olarak bu tanımlamayla çözülen bir kuyruğa ilişkin değişiklik.
 - Uzak kuyruğa ilişkin açık bir tanıtıcı değişikliğinin farklı bir iletim kuyruğuna çözümlenmesine ya da tek bir iletim kuyruğuna çözülememesine neden olan herhangi bir değişiklik. Örneğin, bu şunları içerebilir:
 - Uzak bir kuyruğun yerel tanımlamasının **XmitQName** özniteliğinde, tanımlamanın bir kuyruk için mi, yoksa kuyruk yöneticisi diğer adı için mi kullanıldığına ilişkin bir değişiklik.
 - **z/OS** Yalnızca z/OS sistemlerinde, **IntraGroupqueuing** kuyruk yöneticisi özniteliğinde yapılan bir değişiklik ya da paylaşılan iletim kuyruğunun (SYSTEM.QSG.TRANSMIT.QUEUE) IGQ aracı tarafından kullanılır.

Bunun tek bir istisnası vardır: yeni bir iletim kuyruğu yaratılması. Tanıtıcı açıldığında bu kuyruğa çözülmüş olan, ancak varsayılan iletim kuyruğuna çözülmüş olan bir tanıtıcı geçersiz kılmaz.

- **DefXmitQName** kuyruk yöneticisi özniteliğinde değişiklik. Bu durumda, önceden adlandırılan kuyruğa çözülen (yalnızca varsayılan iletim kuyruğu olduğu için çözülen) tüm açık tanıtıcıları geçersiz olarak işaretlenir. Başka nedenlerle bu kuyruğa çözülen tanıtıcıları etkilenmez.
- Bu kuyruk için ya da bu kuyruğa çözülen bir kuyruk için `MQOO_INPUT_SHARED` erişimi sağlayan iki ya da daha fazla tanıtıcı varsa, **Shareability** kuyruk özniteliği. Bu durumda, bu kuyruk için ya da bu kuyruğa çözülen bir kuyruk için açık olan tüm çekme noktaları, açma seçeneklerinden bağımsız olarak geçersiz olarak işaretlenir.

z/OS z/OS üzerinde, bir ya da daha çok tanıtıcı kuyruğa `MQOO_INPUT_SHARED` ya da `MQOO_INPUT_EXCLUSIVE` erişimi sağlıyorsa, daha önce açıklanan tanıtıcıları geçersiz olarak işaretlenir.

- Açık seçeneklerden bağımsız olarak, bu kuyruk için ya da bu kuyruğa çözülen bir kuyruk için açık olan tüm tanıtıcıları içeren **Usage** kuyruk özniteliği.

Bir tanıtıcı geçersiz olarak işaretlendiğinde, bu tanıtıcıyı kullanan sonraki tüm çağrılar (`MQCLOSE` dışında) `MQRC_OBJECT_CHANGED` neden koduyla başarısız olur. Uygulama bir `MQCLOSE` çağrısı yayınlamalı

(özgün tanıtıcıyı kullanarak) ve kuyruğu yeniden açmalıdır. Önceki başarılı çağrıların eski tanıtıcısıyla ilgili kesinleştirilmemiş güncellemeler, uygulama mantığının gerektirdiği şekilde, yine de kesinleştirilebilir ya da geriletilebilir.

Bir özniteliğin değiştirilmesi bunun gerçekleşmesine neden olursa, çağrım için özel bir zorlamalı sürüm kullanın.

C çağrısı

```
MQOPEN (Hconn, &ObjDesc, Options, &Hobj, &CompCode,  
        &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağrısı

```
CALL 'MQOPEN' USING HCONN, OBJDESC, OPTIONS, HOBJ, COMPCODE, REASON
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağrısı

```
call MQOPEN (Hconn, ObjDesc, Options, Hobj, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL MQOPEN, (HCONN, OBJDESC, OPTIONS, HOBJ, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

HCONN	DS	F	Connection handle
OBJDESC	CMQODA	,	Object descriptor
OPTIONS	DS	F	Options that control the action of MQOPEN
HOBJ	DS	F	Object handle
COMP CODE	DS	F	Completion code
REASON	DS	F	Reason code qualifying COMP CODE

Visual Basic çağrısı

Windows

MQOPEN Hconn, ObjDesc, Options, Hobj, CompCode, Reason

Değiştirgeleri aşağıdaki gibi bildirin:

```
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-Put iletisi

MQPUT çağrısı bir iletiyi bir kuyruğa, dağıtım listesine ya da bir konuya koyar. Kuyruk, dağıtım listesi ya da konu açık olmalıdır.

Sözdizimi


MQPUT (*Hconn, Hobj, MsgDesc, PutMsgOpts, BufferLength, Buffer, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıttıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. Hconn değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

 z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve Hconn için belirtilen şu değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıttıcısı.

Hobj.

Tip: MQHOBJ-giriş

Bu tanıttıcı, iletinin eklendiği kuyruğu ya da iletinin yayınlandığı konuyu gösterir. Hobj değeri, MQOO_OUTPUT seçeneğini belirten önceki bir MQOPEN çağrısı tarafından döndürüldü.

MsgDesc

Tip: MQMD-giriş/çıkış

Bu yapı, gönderilmekte olan iletinin özniteliklerini tanımlar ve koyma isteği tamamlandıktan sonra iletiyle ilgili bilgileri alır. Ayrıntılar için bkz. [“MQMD-İleti tanımlayıcı” sayfa 420.](#)

Uygulama bir version-1 MQMD 'si sağlarsa, ileti verilerinin başına, version-1 değil, version-2 MQMD ' de var olan alanlara ilişkin değerleri belirtmek için bir MQMDE yapısı konabilir. MQMD 'deki *Biçim* alanı, MQMDE ' nin var olduğunu belirtmek için MQFMT_MD_EXTENSION olarak ayarlanmalıdır. Daha fazla ayrıntı için bkz. [“MQMDE-İleti tanımlayıcı uzantısı” sayfa 468 .](#)

MQPMO yapısının OriginalMsgHandle ya da NewMsgHandle alanlarında geçerli bir ileti tanıtıcısı sağlanırsa, uygulamanın MQMD yapısı sağlaması gerekmez. Bu alanlardan birinde hiçbir şey sağlanmazsa, ileti tanıtıcılarıyla ilişkili tanımlayıcıdan iletinin tanımlayıcısı alınır.

API çıkışlarını kullanır ya da kullanmayı planlıyorsanız, belirtik olarak bir MQMD yapısı sağlamanızı ve ileti tanıtıcılarıyla ilişkili ileti tanımlayıcılarını kullanmamanızı öneririz. Bunun nedeni, MQPUT ya da MQPUT1 çağrısıyla ilişkilendirilen API çıkışının, kuyruk yöneticisi tarafından MQPUT ya da MQPUT1 isteğini tamamlamak için hangi MQMD değerlerinin kullanıldığını belirleyemiyor olmasıdır.

PutMsgSeçmeler

Tip: MQPMO-giriş/çıkış

Ayrıntılar için bkz. [“MQPMO-İleti koyma seçenekleri” sayfa 496.](#)

BufferLength

Tip: MQLONG-input

Bufferiçindeki iletinin uzunluğu. Sıfır geçerlidir ve iletinin uygulama verisi içermediğini gösterir. BufferLength için üst sınır çeşitli etmenlere bağlıdır:

- Hedef yerel bir kuyruksa ya da yerel bir kuyruğa çözümlenirse, üst sınır aşağıdakilerden biri olup olmadığına bağlıdır:
 - Yerel kuyruk yöneticisi bölümlenmeyi destekler.
 - Gönderen uygulama, kuyruk yöneticisinin iletiyi kesmesini sağlayan işareti belirtir. Bu işaret MQMF_SEGMENTATION_ALLOWED işaretidir ve version-2 MQMD 'de ya da version-1 MQMD ile kullanılan bir MQMDE' de belirtilebilir.

Bu koşulların her ikisi de karşılandıysa, BufferLength , MQMD 'deki Offset alanının değerinden 999 999 999 999 değerini aşamaz. Bu nedenle, konabilecek en uzun mantıksal ileti 999 999 999 999 bayttır (Offset sıfır olduğunda). Ancak, uygulamanın çalıştığı işletim sistemi ya da ortamın gerektirdiği kaynak kısıtlamaları daha düşük bir sınırla sonuçlanabilir.

Önceki koşullardan biri ya da her ikisi de karşılanmazsa, BufferLength , kuyruğun MaxMsgLength özniteliği ve kuyruk yöneticisinin MaxMsgLength özniteliğinden daha küçük olanı aşamaz.

- Hedef uzak bir kuyruksa ya da uzak bir kuyruğa çözüldüyse, yerel kuyruklara ilişkin koşullar geçerli olur; ancak, hedef kuyruğa ulaşmak için iletinin iletilmesi gereken her kuyruk yöneticisinde; Özellikle:
 1. İletiyi geçici olarak yerel kuyruk yöneticisinde saklamak için kullanılan yerel iletim kuyruğu
 2. İletiyi yerel ve hedef kuyruk yöneticileri arasındaki rotada kuyruk yöneticilerine saklamak için kullanılan ara iletim kuyrukları (varsa)
 3. Hedef kuyruk yöneticisindeki hedef kuyruk

Bu nedenle, konabilecek en uzun ileti, bu kuyrukların ve kuyruk yöneticilerinin en kısıtlayıcı olanına göre yönetilir.

Bir ileti iletim kuyruğundayken, ek bilgiler ileti verileriyle birlikte bulunur ve bu, taşınabilecek uygulama verileri miktarını azaltır. Bu durumda, BufferLengthsınırı saptanırken iletim kuyruklarının MaxMsgLength değerlerinden MQ_MSG_HEADER_LENGTH byte 'ları çıkarın.

Not: İleti konduğunda, yalnızca 1 numaralı koşula uyulmaması (MQRC_MSG_TOO_BIG_FOR_Q ya da MQRC_MSG_TOO_BIG_FOR_Q_MGR neden koduyla) zamanuyumlu olarak tanımlanabilir. 2 ya da 3 numaralı koşullar karşılanmazsa, ileti, bir ara kuyruk yöneticisinde ya da hedef kuyruk yöneticisinde bir teslim edilmeyen ileti kuyruğuna yeniden yönlendirilir. Bu durumda, gönderen tarafından istendiyse bir rapor iletisi oluşturulur.

Arabellek

Tip: MQBYTEExBufferUzunluk-giriş

Bu, gönderilecek uygulama verilerini içeren bir arabellektir. Arabellek, iletideki verilerin niteliğine uygun bir sınırdaki hizalanmalıdır. 4 baytlık hizalama çoğu ileti için uygundur (IBM MQ üstbilgi yapılarını içeren iletiler de içinde olmak üzere), ancak bazı iletiler daha sıkı hizalama gerektirebilir. Örneğin, 64 bitlik ikili tamsayı içeren bir ileti 8 baytlık hizalama gerektirebilir.

Buffer karakter ya da sayısal veri içeriyorsa, **MsgDesc** değiştirgesindeki CodedCharSetId ve Encoding alanlarını veriler için uygun değerlere ayarlayın; bu, iletinin alıcısının verileri (gerekliyse) alıcı tarafından kullanılan karakter kümesine ve kodlamaya dönüştürmesini sağlar.

Not: MQPUT çağrısındaki diğer tüm değiştirgeler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında (**CodedCharSetId** kuyruk yöneticisi özniteliği ve MQENC_NATIVE ile verilir) olmalıdır.

C programlama dilinde, parametre bir gösterge-to-void olarak bildirilir; herhangi bir veri tipinin adresi parametre olarak belirtilebilir.

BufferLength parametresi sıfırsa, Buffer değiştirgesine başvurulamaz; bu durumda, C ya da System/390 çevirici dilinde yazılmış programlar tarafından geçirilen parametre adresi boş olabilir.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici CompCode.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_INCOMPLETE_GROUP

(2241, X'8C1') İleti grubu tamamlanmadı.

MQRC_INCOMPLETE_MSG

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

MQRC_INPERSIST_PERSISTENCE

(2185, X'889 ') Tutarsız kalıcılık belirtimi.

MQRC_TUTARSIZ_UOW

(2245, X'8C5') İş birimi belirtimi tutarsız.

MQRC_MULTIPLE_REASONS

(2136, X'858 ') Birden çok neden kodu döndürüldü.

MQRC_PRIORITY_MAXIMUM değerini aşıyor

(2049, X'801 ') İleti Önceliği, desteklenen değer üst sınırını aşıyor.

MQRC_UNKNOWN_REPORT_OPTION

(2104, X'838 ') İleti tanımlayıcıdaki Rapor seçenek (ler) i tanınmıyor.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ALIAS_TARGTYPE_CHANGED

(2480, X'09B0') Abonelik hedefi tipi kuyruktan konu nesnesine değiştirildi.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X'946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

(2183, X'887 ') API çıkışı yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BACKED_OUT

(2003, X'7D3') Çalışma birimi geri çekildi.

MQRC_BUFFER_ERROR

(2004, X'7D4') Arabellek değıştirgesi geçersiz.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CALL_INTERRUPTED

(2549, X'9F5') MQPUT ya da MQCMIT kesildi ve yeniden bağlanma işlemi kesin bir sonucu yeniden oluşturamıyor.

MQRC_CF_NOT_YOK

(2345, X'929 ') Bağlama olanağı kullanılamıyor.

MQRC_CF_STRUC_FAILED

(2373, X'945 ') Bağlama tesisi yapısı başarısız oldu.

MQRC_CF_STRUC_IN_USE

(2346, X'92A') Coupling-facility yapısı kullanımda.

MQRC_CFGR_HATA

(2416, X'970 ') İleti verilerindeki PCF grup yapısı MQCFGR geçersiz.

MQRC_CFH_HATA

(2235, X'8BB') PCF üstbilgi yapısı geçersiz.

MQRC_CFIF_HATA

(2414, X'96E') İleti verilerindeki PCF tamsayı süzgeci değıştirgesi yapısı geçersiz.

MQRC_CFIL_HATA

(2236, X'8BC') PCF tamsayı listesi parametre yapısı ya da PCIF*64 tamsayı listesi parametre yapısı geçersiz.

MQRC_CFIN_HATA

(2237, X'8BD') PCF tamsayı değıştirgesi yapısı ya da PCIF*64 tamsayı değıştirgesi yapısı geçersiz.

MQRC_CFSF_HATA

(2415, X'96F') İleti verilerindeki PCF dizgi süzgeci değıştirgesi yapısı geçersiz.

MQRC_CFSL_HATA

(2238, X'8BE') PCF dizgi listesi değıştirge yapısı geçersiz.

MQRC_CFST_HATA

(2239, X'8BF') PCF dizgi değıştirgesi yapısı geçersiz.

MQRC_CICS_WAIT_FAILED

(2140, X'85C') Bekleme isteği CICStarafından reddedildi.

MQRC_CLUSTER_EXIT_ERROR (Çıkış)

(2266, X'8DA') Küme iş yükü çıkışı başarısız oldu.

MQRC_CLUSTER_RESOLUTION_HATA

(2189, X'88D') Küme adı çözümlemesi başarısız oldu.

MQRC_CLUSTER_RESOURCE_ERROR

(2269, X'8DD') Küme kaynağı hatası.

MQRC_COD_NOT_VALID_FOR_XCF_Q

(2106, X'83A') COD rapor seçeneği XCF kuyruğu için geçerli değil.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_CONNECTION_NOT_YETKILI

(2217, X'8A9') Bağlantı için yetkili değil.

MQRC_CONNECTION_QUIESCING

(2202, X'89A') Bağlantı susturuyor.

MQRC_CONNECTION_DURDURULUYOR

(2203, X'89B') Bağlantı kapatılıyor.

MQRC_CONTENT_HATA

2554 (X'09FA') İletinin genişletilmiş bir ileti seçici ile bir aboneye teslim edilip edilmeyeceğini belirlemek için ileti içeriği ayrıştırılmadı.

MQRC_CONTEXT_HANDLE_HATA

(2097, X'831 ') Bu gönderide bulunan kuyruk tanıtıcısı bağlamı saklamıyor.

MQRC_CONTEXT_NOT_VAR

(2098, X'832 ') Bu bağlam, gönderme yapılan kuyruk tanıtıcısı için kullanılamaz.

MQRC_DATA_LENGTH_HATA

(2010, X'7DA') Veri uzunluğu parametresi geçersiz.

MQRC_DH_ERROR

(2135, X'857 ') Dağıtım üstbilgisi yapısı geçersiz.

MQRC_DLH_HATA

(2141, X'85D') Ölü harf üstbilgisi yapısı geçerli değil.

MQRC_EPH_HATA

(2420, X' 974 ') Yerleşik PCF yapısı geçersiz.

MQRC_EXPIRY_HATA

(2013, X'7DD') Sona erme süresi geçerli değil.

MQRC_FEEDBACK_HATA

(2014, X'7DE') Geribildirim kodu geçersiz.

MQRC_GLOBAL_UOW_CONFLICT

(2351, X'92F') Genel iş birimleri çakışması.

MQRC_GROUP_ID_HATA

(2258, X'8D2') Grup tanıtıcısı geçersiz.

MQRC_HANDLE_IN_USE_FOR_UOW

(2353, X' 931 ') Genel iş birimi için kullanımda.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_HEADER_HATA

(2142, X'85E') MQ üstbilgi yapısı geçersiz.

MQRC_HOBJ_HATA

(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

MQRC_IIH_HATA

(2148, X'864 ') IMS bilgi üstbilgisi yapısı geçersiz.

MQRC_INCOMPLETE_GROUP

(2241, X'8C1') İleti grubu tamamlanmadı.

MQRC_INCOMPLETE_MSG

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

MQRC_INPERSIST_PERSISTENCE

(2185, X'889 ') Tutarsız kalıcılık belirtimi.

MQRC_TUTARSIZ_UOW

(2245, X'8C5') İş birimi belirtimi tutarsız.

MQRC_LOCAL_UOW_CONFLICT

(2352, X'930 ') Genel iş birimi yerel iş birimiyle çakışıyor.

MQRC_MD_ERROR

(2026, X'7EA') İleti tanımlayıcısı geçersiz.

MQRC_MDE_ERROR

(2248, X'8C8') İleti tanımlayıcı uzantısı geçersiz.

MQRC_MISSING_REPLY_TO_Q

(2027, X'7EB') Yanıt kuyruğu eksik ya da MQPMO_SUPPRESS_REPLYTO kullanıldı

MQRC_MISSING_WIH

(2332, X'91C') İleti verileri MQWIH ile başlamıyor.

MQRC_MSG_FLAGS_HATA

(2249, X'8C9') İleti işaretleri geçerli değil.

MQRC_MSG_SEQ_NUMBER_ERROR (QRC_MSG_SEQ_NUMBER_ERROR)

(2250, X'8CA') İleti sıra numarası geçersiz.

MQRC_MSG_TOO_BIG_FOR_Q

(2030, X'7EE') İleti uzunluğu kuyruk için üst sınırdan fazla.

MQRC_MSG_TOO_BIG_FOR_Q_MGR

(2031, X'7EF') Kuyruk yöneticisi için ileti uzunluğu üst sınırından fazla.

MQRC_MSG_TYPE_HATA

(2029, X'7ED') İleti tanımlayıcıdaki ileti tipi geçersiz.

MQRC_MULTIPLE_REASONS

(2136, X'858 ') Birden çok neden kodu döndürüldü.

MQRC_NO_DESTINATIONS_VAR

(2270, X'8DE') Kullanılabilir hedef kuyruk yok.

MQRC_NOT_OPEN_FOR_OUTPUT

(2039, X'7F7') Kuyruk çıkış için açık değil.

MQRC_NOT_OPEN_FOR_PASS_Tümü

(2093, X'82D') Kuyruk tüm bağlamı geçirmek için açık değil.

MQRC_NOT_OPEN_FOR_PASS_IDENT

(2094, X'82E') Kuyruk kimlik bağlamı için açık değil.

MQRC_NOT_OPEN_FOR_SET_ALL

(2095, X'82F') Kuyruk tüm bağlamı ayarlamak için açık değil.

MQRC_NOT_OPEN_FOR_SET_IDENT

(2096, X'830 ') Kimlik bağlamını ayarlamak için kuyruk açık değil.

MQRC_OBJECT_CHANGED

(2041, X'7F9') Nesne tanımı açıldıktan sonra değiştirildi.

MQRC_OBJECT_HASARLI

(2101, X'835 ') Nesne zarar gördü.

MQRC_OFFSET_HATA

(2251, X'8CB') İleti kesimi görelî konumu geçersiz.

MQRC_OPEN_FAILED

(2137, X'859 ') Nesne başarıyla açılmadı.

MQRC_OPTIONS_HATA

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_ORIGINAL_LENGTH_ERROR

(2252, X'8CC') Özgün uzunluk geçersiz.

MQRC_PAGESET_HATA

(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRC_PAGESET_FULL

(2192, X'890 ') Dış depolama ortamı dolu.

MQRCPCF_HATA
(2149, X'865 ') PCF yapıları geçersiz.

MQRCPERSISTENCE_HATA
(2047, X'7FF') Kalıcılık geçersiz.

MQRCPERSISTENT_NOT_ALLOWED
(2048, X'800 ') Kuyruk kalıcı iletileri desteklemiyor.

MQRCPMO_HATA
(2173, X'87D') Put-message seçenek yapısı geçerli değil.

MQRCPMO_RECORD_FLAGS_HATA
(2158, X'86E') Put ileti kaydı işaretleri geçerli değil.

MQRCPRIORITY_HATA
(2050, X'802 ') İleti önceliği geçersiz.

MQRCPUBLICATION_FAILURE
(2502, X'9C6') Yayın hiçbir aboneye teslim edilmedi.

MQRCPUT_INENGELLEME
(2051, X'803 ') Kuyruk, bu kuyruğun çözüldüğü kuyruk ya da konu için engellenen çağrılarını girin.

MQRCPUT_MSG_RECORDS_HATA
(2159, X'86F') Put ileti kayıtları geçerli değil.

MQRCPUT_NOT_KORUNDU
(2479, X'09AF') Yayın alıkonamadı

MQRQ_SILINDI
(2052, X'804 ') Kuyruk silindi.

MQRQ_FULL
(2053, X'805 ') Kuyruk zaten ileti sayısı üst sınırını içeriyor.

MQRQ_MGR_AD_HATASI
(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRQ_MGR_YOK
(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.

MQRQ_MGR_QUIESCING
(2161, X'871 ') Kuyruk yöneticisi susturuyor.

MQRQ_MGR_DURDURULUYOR
(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRQ_SPACE_NOT_KULLANILAMIYOR
(2056, X'808 ') Diskte kuyruk için kullanılabilir alan yok.

MQRQ_RECONNECT_FAILED
(2548, X'9F4') Yeniden bağlandıktan sonra, yeniden bağlanabilir bir bağlantıya ilişkin tutamaçlar yeniden alınırken bir hata oluştu.

MQRQ_RECS_PRESENT_ERROR
(2154, X'86A') Mevcut kayıt sayısı geçerli değil.

MQRQ_REPORT_OPTIONS_HATA
(2061, X'80D') İleti tanımlayıcıdaki rapor seçenekleri geçersiz.

MQRQ_RESOURCE_PROBLEM
(2102, X'836 ') Sistem kaynakları yetersiz.

MQRQ_RESPONSE_RECORDS_ERROR
(2156, X'86C') Yanıt kayıtları geçerli değil.

MQRQ_RFH_ERROR
(2334, X'91E') MQRFH ya da MQRFH2 yapısı geçersiz.

MQRQ_RMH_ERROR (QRC_RMH_HATA)
(2220, X'8AC') Başvuru iletilisi üstbilgi yapısı geçersiz.

MQRC_SEGMENT_LENGTH_ZERO

(2253, X'8CD') İleti bölümündeki verilerin uzunluğu sıfır.

MQRC_SEGMENTS_NOT_DESTEKLENMIYOR

(2365, X'93D') Bölümler desteklenmiyor.

MQRC_SELECTION_NOT_KULLANILAMIYOR

2551 (X'09F7') Yayının olası bir abonesi var, ancak kuyruk yöneticisi yayını aboneye gönderip göndermeyeceğini denetleyemiyor.

MQRC_STOPPED_BY_CLUSTER_EXIT

(2188, X'88C') Çağrı küme iş yükü çıkışı tarafından reddedildi.

MQRC_STORAGE_CLASS_HATA

(2105, X'839 ') Depolama sınıfı hatası.

MQRC_STORAGE_MEDIUM_FULL

(2192, X'890 ') Dış depolama ortamı dolu.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_SUPPRESSED_BY_EXIT

(2109, X'83D') Çağrı çıkış programı tarafından engellendi.

MQRC_SYNCPOINT_LIMIT_ULAŞILDI

(2024, X'7E8') Yürürlükteki iş birimi içinde başka ileti işlenmez.

MQRC_SYNCPOINT_NOT_YOK

(2072, X'818 ') Syncpoint desteği kullanılmıyor.

MQRC_TM_ERROR

(2265, X'8D9') Tetikleyici ileti yapısı geçersiz.

MQRC_TMC_ERROR

(2191, X'88F') Karakter tetikleyicisi ileti yapısı geçersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

MQRC_UOW_ENLISTMENT_HATA

(2354, X' 932 ') Genel iş birimine kayıt başarısız oldu.

MQRC_UOW_MIX_NOT_DESTEKLENMIYOR

(2355, X' 933 ') İş birimi çağrılarının karışımı desteklenmez.

MQRC_UOW_YOK

(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.

MQRC_WIH_ERROR

(2333, X'91D') MQWIH yapısı geçersiz.

MQRC_ERROR_md_version

(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

MQRC_XQH_HATA

(2260, X'8D4') İletim kuyruğu üstbilgi yapısı geçersiz.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Konu kullanım notları

1. Aşağıdaki notlar konuların kullanımı için geçerlidir:

- a. Bir konuda ileti yayınlamak için MQPUT kullanılırken, abone kuyruğuyla ilgili bir sorun (örneğin, dolu) nedeniyle o konuya ilişkin bir ya da daha çok aboneye yayın verilemediğinde, MQPUT çağrısına döndürülen neden kodu ve teslim davranışı, TOPIC üzerindeki PMSGDLV ya da NPMMSGDLV özniteliklerinin ayarına bağlıdır. MQRO_DEAD_LETTER_Q belirtildiğinde ya da MQRO_DISCARD_MSG belirtildiğinde iletinin atılması, iletinin başarılı bir teslimi olarak kabul edilir. Yayınların hiçbiri teslim edilmezse, MQPUT, MQRC_PUBLICATION_FAILURE ile geri döner. Bu, aşağıdaki durumlarda gerçekleşebilir:

- Bir ileti, PMSGDLV ya da NPMSGDLV (iletinin kalıcı olmasına bağlı olarak) ALL değerine ayarlanmış olarak bir TOPIC içinde yayınlanır ve herhangi bir aboneliğin (sürekli ya da değil) yayını alamayacağı bir kuyruğu vardır.
- Bir ileti, ALLDUR olarak ayarlanmış PMSGDLV ya da NPMSGDLV (iletinin kalıcı olmasına bağlı olarak) ile bir TOPIC ' e yayınlanır ve sürekli abonelik, yayını alamayan bir kuyruğa sahiptir.

Yayınlar aşağıdaki durumlarda bazı abonelere teslim edilemese de MQPUT MQRC_NONE ile dönebilir:

- Bir ileti, PMSGDLV ya da NPMSGDLV (iletinin kalıcı olmasına bağlı olarak) AllaVAIL olarak ayarlanmış bir TOPIC ' e yayınlanır ve kalıcı ya da kalıcı olmayan herhangi bir abonelik, yayını alamayan bir kuyruğa sahiptir.
- Bir ileti, ALLDUR olarak ayarlanmış PMSGDLV ya da NPMSGDLV ile bir TOPIC içinde yayınlanır ve kalıcı olmayan bir abonelik, yayını alamayan bir kuyruğa sahiptir.

Yayınlama iletileri doğru abone kuyruğuna teslim edilemediğinde, teslim edilmeyen iletiler kuyruğunun kullanılıp kullanılmayacağını belirlemek için USEDLO konu özniteliğini kullanabilirsiniz. USEDLO kullanımı hakkında daha fazla bilgi için bkz. [DEFINE TOPIC](#).

- b. Kullanılmakta olan konuya abone yoksa, yayınlanan ileti herhangi bir kuyruğa gönderilmez ve atılır. İletinin kalıcı ya da kalıcı olmaması ya da sınırsız süre bitimi olması ya da süresi dolmuş olması fark etmez, abone yoksa yine de atılır. Bunun istisnası, ileti alıkonacaksa, bu durumda ileti herhangi bir abonenin kuyruğuna gönderilmese de, yeni aboneliklere teslim edilecek konuya ya da MQSUBRQ kullanılarak alıkonan yayınları soran abonelere karşı depolanır.

MQPUT ve MQPUT1

İletileri kuyruğa koymak için hem MQPUT, hem de MQPUT1 çağrılarını kullanabilirsiniz; bu çağrı duruma bağlıdır

- Aynı kuyruğa birden çok ileti yerleştirmek için MQPUT çağrısı kullanın.

Önce MQOO_OUTPUT seçeneğini belirten bir MQOPEN çağrısı, ardından kuyruğa ileti eklemek için bir ya da daha fazla MQPUT isteği gönderilir; son olarak kuyruk bir MQCLOSE çağrısıyla kapatılır. Bu, MQPUT1 çağrısının yinelenen kullanımından daha iyi performans sağlar.

- Kuyruğa tek bir ileti koymak için MQPUT1 çağrısı kullanın.

Bu çağrı, MQOPEN, MQPUT ve MQCLOSE çağrılarını tek bir çağrıya sararak, verilmesi gereken çağrı sayısını en aza indirir.

Hedef Kuyruklar

Aşağıdaki notlar hedef kuyrukların kullanımı için geçerlidir:

1. Bir uygulama ileti gruplarını kullanmadan aynı kuyruğa ileti dizisi koyarsa, Ayrıntılı koşullar yerine getirilirse, bu iletilerin sırası korunur. Bazı koşullar hem yerel, hem de uzak hedef kuyruklar için geçerlidir; diğer koşullar yalnızca uzak hedef kuyruklar için geçerlidir.


Yerel ve uzak hedef kuyruklarına uygulanan koşullar

- Tüm MQPUT çağrıları aynı iş birimi içinde ya da hiçbiri bir iş birimi içinde değil.



İletiler tek bir iş birimi içinde belirli bir kuyruğa yerleştirildiğinde, diğer uygulamalardan gelen iletilerin kuyruktaki ileti sırasıyla birlikte serpiştirilebileceğini unutmayın.

- Tüm MQPUT çağrıları aynı nesne tanıtıcısı *Hobjk* kullanılarak yapılır.

Bazı ortamlarda, çağrılar aynı uygulamadan yapılırsa, farklı nesne tanıtıcıları kullanıldığında da ileti sırası korunur. *Aynı uygulamanın* anlamı ortam tarafından belirlenir:

–  z/OS üzerinde uygulama:

- CICS için CICS görevi
- IMS için, görev

- z/OS toplu iş için görev
-  IBM i' da, iş uygulama.
-  AIX, Linux, and Windows üzerinde uygulama iş parçacığıdır.
- Tüm iletilerin önceliği aynı.
- DefBind kuyruk özniteliği MQBND_BIND_NOT_FIXED değerine sahip olduğunda, iletiler MQOO_BIND_NOT_FIXED ile (ya da MQOO_BIND_AS_Q_DEF ile) bir küme kuyruğuna konmaz.

Uzak hedef kuyrukları için geçerli olan ek koşullar

- Gönderen kuyruk yöneticisinden hedef kuyruk yöneticisine tek bir yol vardır.
Sıradaki bazı iletiler farklı bir yola (örneğin, yeniden yapılandırma, trafik dengeleme ya da ileti büyüklüğüne dayalı yol seçimi nedeniyle) devam edebilirse, hedef kuyruk yöneticisindeki iletilerin sırası garanti edilemez.
- İletiler, gönderen, ara ya da hedef kuyruk yöneticilerindeki gitmeyen iletiler kuyruklarına geçici olarak yerleştirilmez.
İletilerden biri ya da birkaçı geçici olarak gitmeyen iletiler kuyruğuna konursa (örneğin, bir iletim kuyruğu ya da hedef kuyruk geçici olarak dolu olduğu için), iletiler hedef kuyruğa sıra dışı olarak gelebilir.
- İletilerin tümü kalıcı ya da kalıcı değil.

Gönderen ve hedef kuyruk yöneticileri arasındaki rotadaki bir kanalda **NonPersistentMsgSpeed** özniteliği MQNPMS_FAST olarak ayarlanmışsa, kalıcı olmayan iletiler kalıcı iletilerin önüne geçebilir ve kalıcı olmayan iletilerin korunmamasına ilişkin kalıcı iletilerin sırası ile sonuçlanabilir. Ancak, birbiriyle görelili kalıcı iletilerin ve birbirleriyle görelili kalıcı olmayan iletilerin sırası korunur.

Bu koşullar karşılanmazsa, ileti sırasını korumak için ileti gruplarını kullanabilirsiniz, ancak bu, ileti gruplama desteğini hem gönderen hem de alan uygulamaların kullanmasını gerektirir. İleti gruplarıyla ilgili ek bilgi için aşağıdaki konuya bakın:

- [MQMD- MsgFlags alanı](#)
- [MQPMO_LOGICAL_ORDER](#)
- [MQGMO_LOGICAL_ORDER](#)

dağıtım listeleri

Aşağıdaki notlar dağıtım listelerinin kullanımı için geçerlidir.

Dağıtım listeleri aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

1. Bir version-1 ya da version-2 MQPMO kullanarak iletileri bir dağıtım listesine koyabilirsiniz. version-1 MQPMO (ya da RecsPresent sıfıra eşit olan version-2 MQPMO) kullanıyorsanız, uygulama herhangi bir koyma iletilisi kaydı ya da yanıt kaydı sağlayamaz. İleti dağıtım listesindeki bazı kuyruklara başarıyla gönderilirse, diğerlerine değil, hatalarla karşılaşan kuyrukları tanımlayamazsınız.

Uygulama koyma iletilisi kayıtlarını ya da yanıt kayıtlarını sağlarsa, Version alanını MQPMO_VERSION_2 olarak ayarlayın.

RecsPresent 'in sıfır olduğundan emin olarak dağıtım listesinde olmayan tek bir kuyruğa ileti göndermek için version-2 MQPMO' yu da kullanabilirsiniz.

2. Tamamlanma kodu ve neden kodu deęiřtirgeleri ařaęıdaki gibi ayarlanır:

- Daęıtım listesindeki kuyruklara yapılan giriřler aynı řekilde bařarılı ya da bařarısız olursa, tamamlanma kodu ve neden kodu parametreleri ortak sonucu tanımlayacak řekilde ayarlanır. Bu durumda MQRRC yanıt kayıtları (uygulama tarafından saęlandıysa) ayarlanmaz.

Örneęin, her koyma bařarılı olursa, tamamlama kodu ve neden kodu MQRCC_OK ve MQRCC_NONE olarak ayarlanır; tüm kuyruklar koyma için engellendięinden her koyma bařarısız olursa, deęiřtirgeler MQRCC_FAILED ve MQRCC_PUT_INENGELLENIR.

- Daęıtım listesindeki kuyruklara konan konutların tümü aynı řekilde bařarılı ya da bařarısız olmazsa:
 - En az bir koyma bařarılı olursa, tamamlama kodu parametresi MQRCC_WARNING olarak ve tümü bařarısız olursa MQRCC_FAILED olarak ayarlanır.
 - Neden kodu parametresi MQRCC_MULTIPLE_REASON olarak ayarlandı.
 - Yanıt kayıtları (uygulama tarafından saęlandıysa), daęıtım listesindeki kuyruklara iliřkin tek tek tamamlanma kodlarına ve neden kodlarına ayarlanır.

Hedef için açma bařarısız olduęu için hedefe koyma bařarısız olursa, yanıt kaydındaki alanlar MQRCC_FAILED ve MQRCC_OPEN_FAILED; o hedef InvalidDestCountine eklenir.

3. Daęıtım listesindeki bir hedef yerel kuyruęa çözüldüęünde, ileti o kuyruęa olaęan biçimde yerleřtirilir (daęıtım listesi iletisi olarak deęil). Birden çok hedef aynı yerel kuyruęa çözümlenirse, bu tür her hedef için kuyruęa bir ileti yerleřtirilir.

Daęıtım listesindeki bir hedef uzak kuyruęa çözüldüęünde, ileti uygun iletim kuyruęuna yerleřtirilir. Birden çok hedef aynı iletim kuyruęuna çözümlendięinde, bu hedefleri içeren tek bir daęıtım listesi iletisi, bu hedefler uygulama tarafından saęlanan hedefler listesinde bitiřik olmasa da, iletim kuyruęuna yerleřtirilebilir. Ancak, bu yalnızca iletim kuyruęu daęıtım listesi iletilerini destekliyorsa yapılabilir (bkz. [DistLists](#)).

İletim kuyruęu daęıtım listelerini desteklemiyorsa, iletinin olaęan biçimde bir kopyası, bu iletim kuyruęunu kullanan her hedef için iletim kuyruęuna yerleřtirilir.

Uygulama ileti verilerini içeren bir daęıtım listesi iletim kuyruęu için çok büyükse, daęıtım listesi iletisi, her biri daha az hedef içeren daha küçük daęıtım listesi iletilerine bölünür. Uygulama ileti verileri yalnızca kuyruęa sığarsa, daęıtım listesi iletileri hiç kullanılamaz ve kuyruk yöneticisi, iletim kuyruęunu kullanan her hedef için normal biçimde iletinin bir kopyasını oluřturur.

Farklı hedeflerin ileti öncelięi ya da ileti kalıcılıęı farklıysa (bu, uygulama MQRPRI_PRIORITY_AS_Q_DEF ya da MQRPER_PERSISTENCE_AS_Q_DEF belirttięinde ortaya çıkabilir), iletiler aynı daęıtım listesi iletisinde tutulmaz. Kuyruk yöneticisi, farklı öncelik ve kalıcılık deęerlerini barındırmak için gereken sayıda daęıtım listesi iletisi oluřturur.

4. Daęıtım listesine konan bir not ařaęıdakilere neden olabilir:

- Tek bir daęıtım listesi iletisi ya da
- Daha küçük daęıtım listesi iletileri ya da
- Daęıtım listesi iletilerinin ve normal iletilerin karıřımı ya da
- Yalnızca normal iletiler.

Yukarıdakilerin hangilerinin oluřması ařaęıdakilerden hangilerinin oluřmasına baęlıdır:

- Listedeki hedefler yerel, uzak ya da karıřık.
- Hedefler aynı ileti öncelięine ve ileti kalıcılıęına sahip.
- İletim kuyrukları daęıtım listesi iletilerini tutabilir.
- İletim kuyruklarının ileti uzunluęu üst sınırı, iletiyi daęıtım listesi biçiminde barındıracak kadar büyüktür.

Ancak, yukarıdakilerden hangisinin oluřtuęunu dikkate almaksızın, sonuçta ortaya çıkan her *fiziksel* ileti (yani, her normal ileti ya da daęıtım listesi iletisi), ařaęıdaki durumlarda yalnızca *tek* ileti olarak sayılır:

- Uygulamanın bir iş biriminde izin verilen ileti sayısı üst sınırını aşıp aşmadığı denetleniyor (**MaxUncommittedMsgs** kuyruk yöneticisi özneliğine bakın).
 - Tetikleme koşullarının karşılanıp karşılanmadığı denetleniyor.
 - Kuyruk derinlikleri artırılıyor ve kuyrukların kuyruk derinliği üst sınırının aşıp aşılmayacağı denetleniyor.
5. Kuyruk tanımlamalarında, kuyruklar ayrı ayrı açıldığında (örneğin, çözüm yolundaki bir değişiklik), tanıtıcı geçersiz duruma gelmesine neden olacak herhangi bir değişiklik, dağıtım listesi tanıtıcılarının geçersiz olmasına neden olmaz. Ancak, dağıtım listesi tanıtıcısı sonraki bir MQPUT çağrısında kullanıldığında, o kuyruk için bir hata oluşmasına neden olur.

Üst Bilgiler

Uygulama ileti verilerinin başına bir ya da daha çok IBM MQ üstbilgi yapısıyla bir ileti konursa, kuyruk yöneticisi bunların geçerli olduğunu doğrulamak için üstbilgi yapılarında belirli denetimler gerçekleştirir. Kuyruk yöneticisi bir hata saptarsa, çağrı uygun bir neden koduyla başarısız olur. Gerçekleştirilen denetimler, var olan belirli yapılara göre değişir:

- Denetimler yalnızca MQPUT ya da MQPUT1 çağrısında version-2 ya da üstü bir MQMD kullanılıyorsa gerçekleştirilir. İleti verilerinin başında bir MQMDE bulunsu bile, version-1 MQMD kullanılırsa denetimler gerçekleştirilmez.
- Yerel kuyruk yöneticisi tarafından desteklenmeyen yapıların ve iletideki ilk MQDLH 'yi izleyen yapıların geçerliliği denetlenmez.
- MQDH ve MQMDE yapıları, kuyruk yöneticisi tarafından tamamen doğrulanır.
- Diğer yapıların geçerliliği kısmen kuyruk yöneticisi tarafından denetlenir (tüm alanlar denetlenmez).

Kuyruk yöneticisi tarafından gerçekleştirilen genel denetimler şunlardır:

- **StrucId** alanı geçerli olmalıdır.
- **Version** alanı geçerli olmalıdır.
- **StrucLength** alanı, yapıyı ve yapının bir parçasını oluşturan değişken uzunluklu verileri içerecek kadar büyük bir değer belirtmelidir.
- **CodedCharSetId** alanı sıfır olmamalıdır ya da geçersiz bir değer olmalıdır (MQCCSI_DEFAULT, MQCSI_EMBEDDED, MQCCSI_Q_MGR ve MQCCSI_UNDEFINED, çoğu IBM MQ üstbilgi yapısında geçerli değildir).
- Çağrıya ilişkin **BufferLength** parametresi, yapıyı içerecek kadar büyük bir değer belirtmelidir (yapı, iletinin sonunu aşmamalıdır).

Yapılara ilişkin genel denetimlere ek olarak aşağıdaki koşullar da karşılanmalıdır:

- Bir PCF iletisindeki yapıların uzunluklarının toplamı, MQPUT ya da MQPUT1 çağrısında **BufferLength** değiştirilmesiyle belirtilen uzunluğa eşit olmalıdır. PCF iletisi, biçim adı MQFMT_ADMIN, MQFMT_EVENT ya da MQFMT_PCF olan bir iletidir.
- Kesilmiş yapılara izin verilen aşağıdaki durumlar dışında, IBM MQ yapısı kesilmemelidir:
 - Rapor iletileri olan iletiler.
 - PCF iletileri.
 - MQDLH yapısı içeren iletiler. (İlk MQDLH 'yi izleyen yapılar kesilebilir; MQDLH' den önceki yapılar kesilemiyor.)
- Bir IBM MQ yapısı iki ya da daha fazla kesime bölünmemelidir; yapı tamamen bir kesim içinde bulunmalıdır.

Arabellek

Visual Basic programlama dili için a\Ya § daki noktalar kullanır:

- **Buffer** deęiřtirgesinin büyüklüęü **BufferLength** deęiřtirgesinin belirledięi uzunluktan azsa, çağrı başarısız olur. Neden kodu: MQRC_BUFFER_LENGTH_ERROR.
- **Buffer** deęiřtirgesi Stringtipinde olarak bildirildi. Kuyruęa yerleřtirilecek veriler Stringtipinde deęilse, MQPUT yerine MQPUTAny çağrısı.

MQPUTAny çağrısının deęiřtirgeleri MQPUT çağrısıyla aynıdır; ancak, **Buffer** deęiřtirgesi Anytipinde olarak bildirilir ve her tip verinin kuyruęa yerleřtirilmesine izin verir. Ancak bu, Buffer ' in en az BufferLength bayt boyutunda olduęundan emin olmak için denetlenememesi anlamına gelir.

C çağrısı

```
MQPUT (Hconn, Hobj, &MsgDesc, &PutMsgOpts, BufferLength, Buffer,
       &CompCode, &Reason);
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
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 */
```

COBOL çağrısı

```
CALL 'MQPUT' USING HCONN, HOBJ, MSGDESC, PUTMSGOPTS, BUFFERLENGTH,
                  BUFFER, COMPCODE, REASON.
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
** 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.
```

PL/I çağrısı

```
call MQPUT (Hconn, Hobj, MsgDesc, PutMsgOpts, BufferLength, Buffer,
            CompCode, Reason);
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
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 */

```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```

CALL MQPUT, (HCONN, HOBJ, MSGDESC, PUTMSGOPTS, BUFFERLENGTH, X
            BUFFER, COMPCODE, REASON)

```

Değiştirgeleri aşağıdaki gibi bildirin:

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

Visual Basic çağırısı

Windows

```

MQPUT Hconn, Hobj, MsgDesc, PutMsgOpts, BufferLength, Buffer, CompCode,
      Reason

```

Değiştirgeleri aşağıdaki gibi bildirin:

```

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 -Bir ileti koyun

MQPUT1 çağırısı bir iletiyi bir kuyruğa, dağıtım listesine ya da bir konuya koyar.

Kuyruk, dağıtım listesi ya da konunun açık olması gerekmez.

Sözdizimi

MQPUT1 (*Hconn, ObjDesc, MsgDesc, PutMsgOpts, BufferLength, Buffer, CompCode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağırısı tarafından döndürüldü.

z/OS

z/OS for CICS uygulamalarında MQCONN çağırısı atlanabilir ve *Hconn* için belirtilen şu değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

ObjDesc

Tip: MQOD-giriş/çıkış

Bu, iletinin eklendiği kuyruğu ya da iletinin yayınlandığı konuyu tanımlayan bir yapıdır. Ayrıntılar için bkz. [“MQOD-Nesne tanımlayıcı” sayfa 476.](#)

Yapı bir kuyruksa, kullanıcının çıkış kuyruğunu açma yetkisi olmalıdır. Kuyruk bir model kuyruğu olmamalıdır.

MsgDesc

Tip: MQMD-giriş/çıkış

Bu yapı, gönderilmekte olan iletinin özniteliklerini tanımlar ve koyma isteği tamamlandıktan sonra geribildirim bilgilerini alır. Ayrıntılar için bkz. [“MQMD-İleti tanımlayıcı” sayfa 420.](#)

Uygulama bir version-1 MQMD 'si sağlarsa, ileti verilerinin başına, version-1 değil, version-2 MQMD ' de var olan alanlara ilişkin değerleri belirtmek için bir MQMDE yapısı konabilir. MQMD 'deki Format alanını MQMDE' nin var olduğunu göstermek için MQFMT_MD_EXTENSION olarak ayarlayın. Daha fazla ayrıntı için bkz. [“MQMDE-İleti tanımlayıcı uzantısı” sayfa 468 .](#)

MQGMO yapısının MsgHandle alanında ya da MQPMO yapısının OriginalMsgHandle ya da NewMsgHandle alanlarında geçerli bir ileti tanıtıcısı sağlandıysa, uygulamanın MQMD yapısı sağlaması gerekmez. Bu alanlardan birinde hiçbir şey sağlanmazsa, ileti tanıtıcılarıyla ilişkili tanımlayıcıdan iletinin tanımlayıcısı alınır.

PutMsgSeçmeler

Tip: MQPMO-giriş/çıkış

Ayrıntılar için bkz. [“MQPMO-İleti koyma seçenekleri” sayfa 496.](#)

BufferLength

Tip: MQLONG-input

Bufferiçindeki iletinin uzunluğu. Sıfır geçerlidir ve iletinin uygulama verisi içermediğini gösterir. Üst sınır çeşitli etmenlere bağlıdır; **BufferLength** değiştirgesinin açıklaması için bkz. [“MQPUT-Put iletisi” sayfa 742 .](#)

Arabellek

Tip: MQBYTExBufferUzunluk-giriş

Bu, gönderilecek uygulama iletisi verilerini içeren bir arabellektir. Arabelleği, iletideki verilerin niteliğine uygun bir sınırdaki hizalayın. 4 baytlık hizalama çoğu ileti için uygundur (IBM MQ üstbilgi yapılarını içeren iletiler de içinde olmak üzere), ancak bazı iletiler daha sıkı hizalama gerektirebilir. Örneğin, 64 bitlik ikili tamsayı içeren bir ileti 8 baytlık hizalama gerektirebilir.

Buffer karakter ya da sayısal veri içeriyorsa, **MsgDesc** değiştirgesindeki CodedCharSetId ve Encoding alanlarını veriler için uygun değerlere ayarlayın; bu, iletinin alıcısının verileri (gerekliyse) alıcı tarafından kullanılan karakter kümesine ve kodlamaya dönüştürmesini sağlar.

Not: MQPUT1 çağrısındaki diğer tüm değiştirgeler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında (**CodedCharSetId** kuyruk yöneticisi özniteliği ve MQENC_NATIVE ile verilir) olmalıdır.

C programlama dilinde, parametre bir gösterge-to-void olarak bildirilir; herhangi bir veri tipinin adresi parametre olarak belirtilebilir.

BufferLength parametresi sıfırsa, Buffer değiştirgesine başvurulamaz; bu durumda, C ya da System/390 çevirici dilinde yazılmış programlar tarafından geçirilen parametre adresi boş olabilir.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici CompCode.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_MULTIPLE_REASONS

(2136, X'858 ') Birden çok neden kodu döndürüldü.

MQRC_INCOMPLETE_GROUP

(2241, X'8C1') İleti grubu tamamlanmadı.

MQRC_INCOMPLETE_MSG

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

MQRC_PRIORITY_MAXIMUM değerini aşıyor

(2049, X'801 ') İleti Önceliği, desteklenen değer üst sınırını aşıyor.

MQRC_UNKNOWN_REPORT_OPTION

(2104, X'838 ') İleti tanımlayıcıdaki rapor seçenekleri tanınmıyor.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenmiyor.

MQRC_ALIAS_BASE_Q_TYPE_HATA

(2001, X'7D1') Diğer ad temel kuyruğu geçerli bir tip değil.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

(2183, X'887 ') API çıkışı yüklenmiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BACKED_OUT

(2003, X'7D3') Çalışma birimi geri çekildi.

MQRC_BUFFER_ERROR

(2004, X'7D4') Arabellek değiştirgesi geçersiz.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CF_NOT_YOK

(2345, X' 929 ') bağlaşım olanağı kullanılmıyor.

MQRC_CF_STRUC_AUTH_FAILED

(2348, X'92C') Coupling-facility yapısı yetkilendirme denetimi başarısız oldu.

MQRC_CF_STRUC_HATA

(2349, X'92D') Bağlama olanağı yapısı geçerli değil.

MQRC_CF_STRUC_FAILED

(2373, X' 945 ') Bağlama tesisi yapısı başarısız oldu.

MQRC_CF_STRUC_IN_USE

(2346, X'92A') Coupling-facility yapısı kullanımda.

MQRC_CF_STRUC_LIST_HDR_IN_USE

(2347, X'92B') Coupling-facility yapısı liste üstbilgisi kullanımda.

MQRC_CFGR_HATA

(2416, X' 970 ') İleti verilerindeki PCF grup yapısı MQCFGR geçersiz.

MQRC_CFH_HATA

(2235, X'8BB') PCF üstbilgi yapısı geçersiz.

MQRC_CFIF_HATA

(2414, X'96E') İleti verilerindeki PCF tamsayı süzgeci değiştirgesi yapısı geçersiz.

MQRC_CFIL_HATA

(2236, X'8BC') PCF tamsayı listesi parametre yapısı ya da PCIF*64 tamsayı listesi parametre yapısı geçersiz.

MQRC_CFIN_HATA

(2237, X'8BD') PCF tamsayı değiştirgesi yapısı ya da PCIF*64 tamsayı değiştirgesi yapısı geçersiz.

MQRC_CFSF_HATA

(2415, X'96F') İleti verilerindeki PCF dizgi süzgeci değiştirgesi yapısı geçersiz.

MQRC_CFSL_HATA

(2238, X'8BE') PCF dizgi listesi değiştirge yapısı geçersiz.

MQRC_CFST_HATA

(2239, X'8BF') PCF dizgi değiştirgesi yapısı geçersiz.

MQRC_CICS_WAIT_FAILED

(2140, X'85C') Bekleme isteği CICStarafından reddedildi.

MQRC_CLUSTER_EXIT_ERROR (Çıkış)

(2266, X'8DA') Küme iş yükü çıkışı başarısız oldu.

MQRC_CLUSTER_RESOLUTION_HATA

(2189, X'88D') Küme adı çözümlemesi başarısız oldu.

MQRC_CLUSTER_RESOURCE_ERROR

(2269, X'8DD') Küme kaynağı hatası.

MQRC_COD_NOT_VALID_FOR_XCF_Q

(2106, X'83A') COD rapor seçeneği XCF kuyruğu için geçerli değil.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_CONNECTION_NOT_YETKILI

(2217, X'8A9') Bağlantı için yetkili değil.

MQRC_CONNECTION QUIESCING

(2202, X'89A') Bağlantı susturuyor.

MQRC_CONNECTION_DURDURULUYOR

(2203, X'89B') Bağlantı kapatılıyor.

MQRC_CONTENT_HATA

2554 (X'09FA') İleti içeriği, iletinin genişletilmiş ileti seçici ile bir aboneye teslim edilip edilemeyeceğini belirlemek için ayrıştırılmadı.

MQRC_CONTEXT_HANDLE_HATA

(2097, X'831 ') Bu gönderide bulunan kuyruk tanıtıcısı bağlamı saklamıyor.

MQRC_CONTEXT_NOT_VAR

(2098, X'832 ') Bu bağlam, gönderme yapılan kuyruk tanıtıcısı için kullanılamaz.

MQRC_DATA_LENGTH_HATA

(2010, X'7DA') Veri uzunluğu parametresi geçersiz.

MQRC_DB2_NOT_AVAILABLE
(2342, X'926 ') Db2 altsistemi kullanılmıyor.

MQRC_DEF_XMIT_Q_TYPE_HATA
(2198, X'896 ') Varsayılan iletim kuyruğu yerel değil.

MQRC_DEF_XMIT_Q_USAGE_HATA
(2199, X'897 ') Varsayılan iletim kuyruğu kullanım hatası.

MQRC_DH_ERROR
(2135, X'857 ') Dağıtım üstbilgisi yapısı geçersiz.

MQRC_DLH_HATA
(2141, X'85D') Ölü harf üstbilgisi yapısı geçerli değil.

MQRC_EPH_HATA
(2420, X'974 ') Yerleşik PCF yapısı geçersiz.

MQRC_EXPIRY_HATA
(2013, X'7DD') Sona erme süresi geçerli değil.

MQRC_FEEDBACK_HATA
(2014, X'7DE') Geribildirim kodu geçersiz.

MQRC_GLOBAL_UOW_CONFLICT
(2351, X'92F') Genel iş birimleri çakışması.

MQRC_GROUP_ID_HATA
(2258, X'8D2') Grup tanıtıcısı geçersiz.

MQRC_HANDLE_IN_USE_FOR_UOW
(2353, X'931 ') Genel iş birimi için kullanımda.

MQRC_HANDLE_NOT_YOK
(2017, X'7E1') Başka tanıtıcı yok.

MQRC_HCONN_HATA
(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_HEADER_HATA
(2142, X'85E') IBM MQ üstbilgi yapısı geçersiz.

MQRC_IIH_HATA
(2148, X'864 ') IMS bilgi üstbilgisi yapısı geçersiz.

MQRC_LOCAL_UOW_CONFLICT
(2352, X'930 ') Genel iş birimi yerel iş birimiyle çakışıyor.

MQRC_MD_ERROR
(2026, X'7EA') İleti tanımlayıcısı geçersiz.

MQRC_MDE_ERROR
(2248, X'8C8') İleti tanımlayıcı uzantısı geçersiz.

MQRC_MISSING_REPLY_TO_Q
(2027, X'7EB') Yanıt kuyruğu eksik.

MQRC_MISSING_WIH
(2332, X'91C') İleti verileri MQWIH ile başlamıyor.

MQRC_MSG_FLAGS_HATA
(2249, X'8C9') İleti işaretleri geçerli değil.

MQRC_MSG_SEQ_NUMBER_ERROR (QRC_MSG_SEQ_NUMBER_ERROR)
(2250, X'8CA') İleti sıra numarası geçersiz.

MQRC_MSG_TOO_BIG_FOR_Q
(2030, X'7EE') İleti uzunluğu kuyruk için üst sınırdan fazla.

MQRC_MSG_TOO_BIG_FOR_Q_MGR
(2031, X'7EF') Kuyruk yöneticisi için ileti uzunluğu üst sınırından fazla.

MQRC_MSG_TYPE_HATA
(2029, X'7ED') İleti tanımlayıcıdaki ileti tipi geçersiz.

MQRC_MULTIPLE_REASONS
(2136, X'858 ') Birden çok neden kodu döndürüldü.

MQRC_NO_DESTINATIONS_VAR
(2270, X'8DE') Kullanılabilir hedef kuyruk yok.

MQRC_NOT_YETKILI
(2035, X'7F3') Erişim için yetkili değil.

MQRC_OBJECT_HASARLI
(2101, X'835 ') Nesne zarar gördü.

MQRC_OBJECT_IN_USE
(2042, X'7FA') Nesne zaten çıkarılan seçeneklerle açık.

MQRC_OBJECT_LEVEL_UYUMSUZ
(2360, X' 938 ') Nesne düzeyi uyumlu değil.

MQRC_OBJECT_NAME_ERROR
(2152, X'868 ') Nesne adı geçersiz.

MQRC_OBJECT_NOT_UNIQUE
(2343, X' 927 ') Nesne benzersiz değil.

MQRC_OBJECT_Q_MGR_NAME_ERROR
(2153, X'869 ') Nesne kuyruğu yöneticisi adı geçersiz.

MQRC_OBJECT_RECORDS_ERROR (RECORDS_HATASI)
(2155, X'86B') Nesne kayıtları geçerli değil.

MQRC_OBJECT_TYPE_ERROR
(2043, X'7FB') Nesne tipi geçersiz.

MQRC_OD_HATA
(2044, X'7FC') Nesne tanımlayıcı yapısı geçerli değil.

MQRC_OFFSET_HATA
(2251, X'8CB') İleti kesimi görelî konumu geçersiz.

MQRC_OPTIONS_HATA
(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_ORIGINAL_LENGTH_ERROR
(2252, X'8CC') Özgün uzunluk geçersiz.

MQRC_PAGESET_HATA
(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRC_PAGESET_FULL
(2192, X'890 ') Dış depolama ortamı dolu.

MQRC_PCF_HATA
(2149, X'865 ') PCF yapıları geçersiz.

MQRC_PERSISTENCE_HATA
(2047, X'7FF') Kalıcılık geçersiz.

MQRC_PERSISTENT_NOT_ALLOWED
(2048, X'800 ') Kuyruk kalıcı iletileri desteklemiyor.

MQRC_PMO_HATA
(2173, X'87D') Put-message seçenek yapısı geçerli değil.

MQRC_PMO_RECORD_FLAGS_HATA
(2158, X'86E') Put ileti kaydı işaretleri geçerli değil.

MQRC_PRIORITY_HATA
(2050, X'802 ') İleti önceliği geçersiz.

MQRC_PUBLICATION_FAILURE
(2502, X'9C6') Yayın hiçbir aboneye teslim edilmedi.

MQRC_PUT_INENGELLEME
(2051, X'803 ') Kuyruk için çağrılarını engelleme.

MQRC_PUT_MSG_RECORDS_HATA

(2159, X'86F') Put ileti kayıtları geçerli değil.

MQRC_Q_SILINDI

(2052, X'804 ') Kuyruk silindi.

MQRC_Q_FULL

(2053, X'805 ') Kuyruk zaten ileti sayısı üst sınırını içeriyor.

MQRC_Q_MGR_AD_HATASI

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

MQRC_Q_MGR_YOK

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.

MQRC_Q_MGR QUIESCING

(2161, X'871 ') Kuyruk yöneticisi susturuyor.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_Q_SPACE_NOT_KULLANILAMIYOR

(2056, X'808 ') Diskte kuyruk için kullanılabilir alan yok.

MQRC_Q_TYPE_HATA

(2057, X'809 ') Kuyruk tipi geçersiz.

MQRC_RECS_PRESENT_ERROR

(2154, X'86A') Mevcut kayıt sayısı geçerli değil.

MQRC_REMOTE_Q_NAME_ERROR

(2184, X'888 ') Uzak kuyruk adı geçersiz.

MQRC_REPORT_OPTIONS_HATA

(2061, X'80D') İleti tanımlayıcıdaki rapor seçenekleri geçersiz.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_RESPONSE_RECORDS_ERROR

(2156, X'86C') Yanıt kayıtları geçerli değil.

MQRC_RFH_ERROR

(2334, X'91E') MQRFH ya da MQRFH2 yapısı geçersiz.

MQRC_RMH_ERROR (QRC_RMH_HATA)

(2220, X'8AC') Başvuru iletilisi üstbilgi yapısı geçersiz.

MQRC_SECURITY_HATA

(2063, X'80F') Güvenlik hatası oluştu.

MQRC_SEGMENT_LENGTH_ZERO

(2253, X'8CD') İleti bölümündeki verilerin uzunluğu sıfır.

MQRC_SELECTION_NOT_KULLANILAMIYOR

2551 (X'09F7') Yayının olası bir abonesi var, ancak kuyruk yöneticisi yayını aboneye gönderip göndermeyeceğini denetleyemiyor.

MQRC_STOPPED_BY_CLUSTER_EXIT

(2188, X'88C') Çağrı küme iş yükü çıkışı tarafından reddedildi.

MQRC_STORAGE_CLASS_HATA

(2105, X'839 ') Depolama sınıfı hatası.

MQRC_STORAGE_MEDIUM_FULL

(2192, X'890 ') Dış depolama ortamı dolu.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_SUPPRESSED_BY_EXIT

(2109, X'83D') Çağrı çıkış programı tarafından engellendi.

MQRC_SYNCPOINT_LIMIT_ULAŞILDI

(2024, X'7E8') Yürürlükteki iş birimi içinde başka ileti işlenmez.

MQRC_SYNCPOINT_NOT_YOK

(2072, X'818 ') Syncpoint desteği kullanılamıyor.

MQRC_TM_ERROR

(2265, X'8D9') Tetikleyici ileti yapısı geçersiz.

MQRC_TMC_ERROR

(2191, X'88F') Karakter tetikleyicisi ileti yapısı geçersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

MQRC_UNKNOWN_ALIAS_BASE_Q

(2082, X'822 ') Bilinmeyen diğer ad temel kuyruğu.

MQRC_UNKNOWN_DEF_XMIT_Q

(2197, X'895 ') Bilinmeyen varsayılan iletim kuyruğu.

MQRC_UNKNOWN_OBJECT_NAME

(2085, X'825 ') Bilinmeyen nesne adı.

MQRC_UNKNOWN_OBJECT_Q_MGR

(2086, X'826 ') Bilinmeyen nesne kuyruk yöneticisi.

MQRC_UNKNOWN_REMOTE_Q_MGR

(2087, X'827 ') Bilinmeyen uzak kuyruk yöneticisi.

MQRC_UNKNOWN_XMIT_Q

(2196, X'894 ') Bilinmeyen iletim kuyruğu.

MQRC_UOW_ENLISTMENT_HATA

(2354, X' 932 ') Genel iş birimine kayıt başarısız oldu.

MQRC_UOW_MIX_NOT_DESTEKLENMIYOR

(2355, X' 933 ') İş birimi çağrılarının karışımı desteklenmez.

MQRC_UOW_YOK

(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.

MQRC_WIH_ERROR

(2333, X'91D') MQWIH yapısı geçersiz.

MQRC_YANLIŞ_CF_DÜZEYI

(2366, X'93E') Coupling-facility yapısı yanlış düzeyde.

MQRC_ERROR_md_version

(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

MQRC_XMIT_Q_TYPE_HATA

(2091, X'82B') İletim kuyruğu yerel değil.

MQRC_XMIT_Q_USAGE_HATASI

(2092, X'82C') İletim kuyruğu yanlış kullanımla.

MQRC_XQH_HATA

(2260, X'8D4') İletim kuyruğu üstbilgi yapısı geçersiz.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. Bir kuyruğa ileti koymak için hem MQPUT hem de MQPUT1 çağrıları kullanılabilir; kullanılacak çağrı aşağıdaki koşullara bağlıdır:

- Birden çok iletiyi *aynı* kuyruğa yerleştirmek için MQPUT çağrısını kullanın.

Önce MQOO_OUTPUT seçeneğini belirten bir MQOPEN çağrısı, ardından kuyruğa ileti eklemek için bir ya da daha fazla MQPUT isteği gönderilir; son olarak kuyruk bir MQCLOSE çağrısıyla kapatılır. Bu, MQPUT1 çağrısının yinelenen kullanımından daha iyi performans sağlar.

- Kuyruğa yalnızca *bir* ileti koymak için MQPUT1 çağrısı kullanın.

Bu çağrı, MQOPEN, MQPUT ve MQCLOSE çağrılarını tek bir çağrıya sararak, verilmesi gereken çağrı sayısını en aza indirir.

2. Bir uygulama ileti gruplarını kullanmadan aynı kuyruğa ileti dizisi koyarsa, belirli koşullar yerine getirilirse, bu iletilerin sırası korunur. Ancak, çoğu ortamda MQPUT1 çağrısı bu koşulları karşılamaz ve bu nedenle ileti sırasını korumaz. MQPUT çağrısı bu ortamlarda kullanılmalıdır. Ayrıntılar için bkz. [MQPUT kullanım notları](#) .
3. MQPUT1 çağrısı, iletileri dağıtım listelerine koymak için kullanılabilir. Bununla ilgili genel bilgi için MQOPEN ve MQPUT çağrılarında ilişkin kullanım notlarına bakın.

Dağıtım listeleri aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

MQPUT1 çağrısı kullanılırken aşağıdaki farklılıklar söz konusudur:

- a. Uygulama MQRR yanıt kayıtları sağlıyorsa, bunlar MQOD yapısı kullanılarak sağlanmalıdır; bunlar MQPMO yapısı kullanılarak sağlanamaz.
- b. MQRC_OPEN_FAILED neden kodu, yanıt kayıtlarında MQPUT1 tarafından hiçbir zaman döndürülmez; bir kuyruk açılmazsa, o kuyruğa ilişkin yanıt kaydı açma işleminden kaynaklanan neden kodunu içerir.

Bir kuyruğa ilişkin açma işlemi MQCC_WARNING tamamlanma koduyla başarılı olursa, o kuyruğa ilişkin yanıt kaydındaki tamamlanma kodu ve neden kodu, koyma işleminden kaynaklanan tamamlanma ve neden kodlarıyla değiştirilir.

MQOPEN ve MQPUT çağrılarında olduğu gibi, kuyruk yöneticisi yanıt kayıtlarını (sağlandıysa) yalnızca, çağrı sonucu dağıtım listesindeki tüm kuyruklar için aynı olmadığında ayarlar; bu, çağrı tamamlanırken MQRC_MULTIPLE_REASON neden koduyla gösterilir.

4. MQPUT1 çağrısı bir iletiyi küme kuyruğuna koymak için kullanılırsa, çağrı MQOPEN çağrısında MQOO_BIND_NOT_FIXED belirtilmiş gibi davranır.
5. Uygulama ileti verilerinin başına bir ya da daha çok IBM MQ üstbilgi yapısıyla bir ileti konursa, kuyruk yöneticisi bunların geçerli olduğunu doğrulamak için üstbilgi yapılarında belirli denetimler gerçekleştirir. Bununla ilgili daha fazla bilgi için MQPUT çağrısına ilişkin kullanım notlarına bakın.
6. Birden çok uyarı durumu ortaya çıkarsa (**CompCode** değiştirgesine bakın), döndürülen neden kodu, aşağıdaki listede geçerli olan ilk durumdur:
 - a. MQRC_MULTIPLE_REASONS
 - b. MQRC_INCOMPLETE_MSG
 - c. MQRC_INCOMPLETE_GROUP
 - d. MQRC_PRIORITY_AŞAN eds_maximum ya da MQRC_UNKNOWN_REPORT_OPTION

7. Visual Basic programlama dili için aŸa Ÿ daki noktalar kullanılır:

- **Buffer** değiştirgesinin büyüklüğü **BufferLength** değiştirgesinin belirlediği uzunluktan azsa, çağrı başarısız olur. Neden kodu: MQRC_BUFFER_LENGTH_ERROR.
- **Buffer** değiştirgesi Stringtipinde olarak bildirildi. Kuyruğa yerleştirilecek veriler Stringtipinde değilse, MQPUT1 yerine MQPUT1Any ' i arayan.

MQPUT1Any çağrısının değiştirgeleri MQPUT1 çağrısıyla aynıdır; ancak, **Buffer** değiştirgesinin tip olarak bildirilmesi Any, herhangi bir veri tipinin kuyruğa konmasına izin verir. Ancak bu, Buffer ' in en az BufferLength bayt boyutunda olduğundan emin olmak için denetlenememesi anlamına gelir.

8. MQPUT1 çağırısı MQPMO_SYNCPOINT ile yayınlandığında, koyma işleminin zamanuyumsuz olarak tamamlanması için varsayılan davranış değişir. Bu, döndürülen MQOD ve MQMD yapılarındaki belirli alanlara dayalı olan, ancak şimdi tanımlanmamış değerler içeren bazı uygulamaların davranışında bir değişikliğe neden olabilir. Bir uygulama, koyma işleminin zamanuyumlu olarak gerçekleştirildiğinden ve tüm uygun alan değerlerinin tamamlandığından emin olmak için MQPMO_SYNC_RESPONSE değerini belirtebilir.

C çağırısı

```
MQPUT1 (Hconn, &ObjDesc, &MsgDesc, &PutMsgOpts,  
        BufferLength, Buffer, &CompCode, &Reason);
```

Değişirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağırısı

```
CALL 'MQPUT1' USING HCONN, OBJDESC, MSGDESC, PUTMSGOPTS,  
                   BUFFERLENGTH, BUFFER, COMPCODE, REASON.
```

Değişirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağırısı

```
call MQPUT1 (Hconn, ObjDesc, MsgDesc, PutMsgOpts, BufferLength, Buffer,  
            CompCode, Reason);
```

Değişirgeleri aşağıdaki gibi bildirin:

```
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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL MQPUT1, (HCONN, OBJDESC, MSGDESC, PUTMSGOPTS, BUFFERLENGTH, X
             BUFFER, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

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

Visual Basic çağrısı

Windows

```
MQPUT1 Hconn, ObjDesc, MsgDesc, PutMsgOpts, BufferLength, Buffer,
       CompCode, Reason
```

Değiştirgeleri aşağıdaki gibi bildirin:

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-Nesne özniteliklerini ayarla

Bir tanıtıcı ile gösterilen bir nesnenin özniteliklerini değiştirmek için MQSET çağrısını kullanın. Nesne bir kuyruk olmalıdır.

Sözdizimi

MQSET (*Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs, CharAttrLength, CharAttrs, Compcode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. Hconn değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

z/OS

z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve Hconn için belirtilen şu değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

Hobj.

Tip: MQHOBj-giriş

Bu tanıtıcı, ayarlanacak öznitelikleri olan kuyruk nesnesini gösterir. Tanıtıcı değeri, MQOO_SET seçeneğini belirten önceki bir MQOPEN çağırısı tarafından döndürüldü.

SelectorCount

Tip: MQLONG-input

Bu, Selectors dizisinde sağlanan seçicilerin sayısıdır. Ayarlanacak özniteliklerin sayısıdır. Sıfır geçerli bir değerdir. İzin verilen üst sınır 256 'dır.

Seçiciler

Tip: MQLONGxSelectorSayı-giriş

Bu bir **SelectorCount** öznitelik seçicileri dizisidir; her seçici, ayarlanacak bir değere sahip bir özniteliği (tamsayı ya da karakter) tanımlar.

Her seçici, Hobj ' in temsil ettiği kuyruk tipi için geçerli olmalıdır. Daha sonra listelendiği gibi, yalnızca belirli MQIA_ * ve MQCA_ * değerlerine izin verilir.

Seçiciler herhangi bir sırada belirtilebilir. Tamsayı öznitelik seçicilere (MQIA_ * seçiciler) karşılık gelen öznitelik değerleri, IntAttrrs içinde bu seçicilerin Selectors içinde gerçekleştiği sırayla belirtilmelidir. Karakter özniteliği seçicilere (MQCA_ * seçicileri) karşılık gelen öznitelik değerleri, CharAttrrs içinde bu seçicilerin oluştuğu sırada belirtilmelidir. MQIA_ * seçicileri MQCA_ * seçicileriyle serpiştirilebilirler; yalnızca her tipteki görelî sıra önemlidir.

Aynı seçiciyi bir kereden fazla belirtebilirsiniz; bunu yaparsanız, belirli bir seçici için belirtilen son değer geçerli olan değerdir.

Not:

1. Tamsayı ve karakter öznitelik seçicileri iki farklı aralık içinde ayrılır; MQIA_ * seçicileri MQIA_FIRST-MQIA_LAST aralığında ve MQCA_FIRST-MQCA_LAST aralığındaki MQCA_ * seçicileri içinde bulunur.
Her aralık için, MQIA_LAST_KULLANILAN ve MQCA_LAST_KULLANILAN değişmezleri, kuyruk yöneticisinin kabul ettiği en yüksek değeri tanımlar.
2. Önce tüm MQIA_ * seçicileri oluşursa, Selectors ve IntAttrrs dizilerinde karşılık gelen öğeleri adreslemek için aynı öge numaraları kullanılabilir.
3. **SelectorCount** parametresi sıfırsa, Selectors değiştirilmesine başvurulamaz; bu durumda, C ya da System/390 çevirici dilinde yazılmış programlar tarafından geçirilen parametre adresi boş olabilir.

Ayarlanabilen öznitelikler aşağıdaki tabloda listelenir. Bu çağrı kullanılarak başka bir öznitelik ayarlanamaz. MQCA_ * öznitelik seçicileri için, CharAttrrs içinde gerekli olan dizginin bayt cinsinden uzunluğunu tanımlayan değişmez ayrıç içinde sağlanır.

Çizelge 554. Kuyruklar için MQSET öznitelik seçicileri		
Seçici	Açıklama	Not
MQCA_TRIGGER_DATA	Tetikleyici verileri (MQ_TRIGGER_DATA_LENGTH).	
MQIA_DIST_LIST	Dağıtım listesi desteği.	1
MQIA_INHIBIT_GET	Alma işlemlerine izin verilip verilmediğini belirler.	
MQIA_INHIBIT_PUT	Koyma işlemlerine izin verilip verilmediğini belirler.	
MQIA_TRIGGER_CONTROL	Tetikleme kontrolü.	
MQIA_TRIGGER_DEPTH	Tetikleme derinliği.	
MQIA_TRIGGER_MSG_PRIORITY	Tetikleyiciler için eşik ileti önceliği.	
MQIA_TRIGGER_TYPE	Tetikleyici tipi.	

Not:

1. Yalnızca aşağıdaki altyapılarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

IntAttrSayısı

Tip: MQLONG-input

Bu, IntAttr dizisindeki öğelerin sayısıdır ve en az **Selectors** değiştirgesindeki MQIA_* seçicilerinin sayısı olmalıdır. Sıfır, yoksa geçerli bir değerdir.

IntAttr

Tip: MQLONGxIntAttrCount -input

Bu, IntAttrCount tamsayı öznitelik değerlerinin bir dizisidir. Bu öznitelik değerleri, Selectors dizisindeki MQIA_* seçicileri ile aynı sırada olmalıdır.

IntAttrCount ya da **SelectorCount** parametresi sıfırsa, IntAttr değiştirgesine başvurulamaz; bu durumda, C ya da System/390 çevirici dilinde yazılmış programlar tarafından geçirilen parametre adresi boş olabilir.

CharAttrUzunluğu

Tip: MQLONG-input

Bu, CharAttr parametresinin bayt cinsinden uzunluğudur ve en azından Selectors dizisinde belirtilen karakter özniteliklerinin uzunluklarının toplamı olmalıdır. Selectors içinde MQCA_* seçicileri yoksa sıfır geçerli bir değerdir.

CharAttr

Tip: MQCHAR x CharAttrUzunluk-giriş

Bu, karakter özniteliği değerlerini içeren, birbirine bitiştirilmiş arabellektir. Arabelleğin uzunluğu **CharAttrLength** değiştirgesiyle verilir.

Karakter öznitelikleri, Selectors dizisindeki MQCA_* seçicileri ile aynı sırada belirtilmelidir. Her karakter özniteliğinin uzunluğu sabittir (bkz. [Seçiciler](#)). Bir öznitelik için ayarlanacak değer, özniteliğin tanımlı uzunluğundan daha az boşluk olmayan karakter içeriyorsa, öznitelik değerinin özniteliğin tanımlı uzunluğuyla eşleşmesi için CharAttr içindeki değeri sağa boşluklarla doldurun.

CharAttrLength ya da **SelectorCount** parametresi sıfırsa, CharAttr değiştirgesine başvurulamaz; bu durumda, C ya da System/390 çevirici dilinde yazılmış programlar tarafından geçirilen parametre adresi boş olabilir.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici CompCode.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Baędařtırıcı kullanılmıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852 ') Baędařtırıcı hizmet modülü yüklenmiyor.

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu.

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

(2183, X'887 ') API çıkışı yüklenmiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CF_NOT_YOK

(2345, X' 929 ') Baęlama olanaęı kullanılmıyor.

MQRC_CF_STRUC_FAILED

(2373, X' 945 ') Baęlama tesisi yapısı başarısız oldu.

MQRC_CF_STRUC_IN_USE

(2346, X'92A') Coupling-facility yapısı kullanımda.

MQRC_CF_STRUC_LIST_HDR_IN_USE

(2347, X'92B') Coupling-facility yapısı liste üstbilgisi kullanımda.

MQRC_CHAR_ATTR_LENGTH_HATA

(2006, X'7D6') Karakter özniteliklerinin uzunluęu geçersiz.

MQRC_CHAR_ATTRS_HATA

(2007, X'7D7') Karakter öznitelikleri dizgisi geçersiz.

MQRC_CICS_WAIT_FAILED

(2140, X'85C') Bekleme isteęi CICStarafından reddedildi.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine baęlantı kaybedildi.

MQRC_CONNECTION_NOT_YETKILI

(2217, X'8A9') Baęlantı için yetkili deęil.

MQRC_CONNECTION_DURDURULUYOR

(2203, X'89B') Baęlantı kapatılıyor.

MQRC_DB2_NOT_AVAILABLE

(2342, X' 926 ') Db2 altsistemi kullanılmıyor.

MQRC_HCONN_HATA

(2018, X'7E2') Baęlantı tanıtıcısı geçersiz.

MQRC_HOBJ_HATA

(2019, X'7E3') Nesne tanıtıcısı geçerli deęil.

MQRC_INHIBIT_VALUE_ERROR

(2020, X'7E4') İnhibe-alma ya da engelleme-koyma kuyruęu öznitelięi deęeri geçerli deęil.

MQRC_INT_ATTR_COUNT_HATA

(2021, X'7E5') Tamsayı özniteliklerinin sayısı geçersiz.

MQRC_INT_ATTRS_ARRAY_HATA

(2023, X'7E7') Tamsayı öznitelikleri dizisi geçerli deęil.

MQRC_NOT_OPEN_FOR_SET

(2040, X'7F8') Kuyruk ayarlanmak üzere açılmadı.

MQRC_OBJECT_CHANGED

(2041, X'7F9') Nesne tanımı açıldıktan sonra deęiştirildi.

MQRC_OBJECT_HASARLI

(2101, X'835 ') Nesne zarar gördü.

MQRC_PAGESET_HATA

(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.

MQRC_Q_SILINDI

(2052, X'804 ') Kuyruk silindi.

MQRC_Q_MGR_AD_HATASI

(2058, X'80A') Kuyruk yöneticisi adı geçerli deęil ya da bilinmiyor.

MQRC_Q_MGR_YOK

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_SELECTOR_COUNT_HATA

(2065, X'811 ') Seçici sayısı geçerli deęil.

MQRC_SELECTOR_ERROR

(2067, X'813 ') Öznitelik seçici geçersiz.

MQRC_SELECTOR_LIMIT_AŞILDI

(2066, X'812 ') Seçicilerin sayısı çok büyük.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_SUPPRESSED_BY_EXIT

(2109, X'83D') Çaęrı çıkış programı tarafından engellendi.

MQRC_TRIGGER_CONTROL_HATA

(2075, X'81B') Tetikleme denetimi öznitelięine ilişkin deęer geçersiz.

MQRC_TRIGGER_DEPTH_HATA

(2076, X'81C') Tetikleme derinlięi öznitelięine ilişkin deęer geçersiz.

MQRC_TRIGGER_MSG_PRIORITY_ERR

(2077, X'81D') Tetikleyici ileti öncelięi öznitelięine ilişkin deęer geçersiz.

MQRC_TRIGGER_TYPE_ERROR (QRC_TRIGGER_TYPE_HATA)

(2078, X'81E') Tetikleyici tipi öznitelięine ilişkin deęer geçersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. Bu çağrıyı kullanarak uygulama, bir tamsayı öznitelikleri dizisi ya da bir karakter öznitelięi dizileri derlemi ya da her ikisini birden belirtebilir. Hata ortaya çıkmazsa, belirtilen özniteliklerin tümü aynı anda ayarlanır. Bir hata oluşursa (örneğin, bir seçici geçerli deęilse ya da bir öznitelięi geçerli olmayan bir deęere ayarlama girişiminde bulunulursa), çağrı başarısız olur ve öznitelik ayarlanmaz.
2. Özniteliklerin deęerleri MQINQ çağrısı kullanılarak belirlenebilir; ayrıntılar için bkz. "[MQINQ-Nesne özniteliklerini sorgu](#)" sayfa 696 .

Not: MQINQ çağrısı kullanılarak sorulabilecek deęerleri olan tüm özniteliklerin deęerleri MQSET çağrısı kullanılarak deęiştirilemez. Örneęin, bu çağrıyla hiçbir işlem nesnesi ya da kuyruk yöneticisi öznitelięi ayarlanamaz.

3. Öznitelik değişiklikleri, kuyruk yöneticisinin yeniden başlatılmasından sonra korunur (kuyruk yöneticisinin yeniden başlatılmasından sonra devam etmeyen geçici dinamik kuyruklarda yapılan değişiklikler dışında).
4. MQSET çağrısının kullanıldığı bir model kuyruğunun özniteliklerini değiştiremezsiniz. Ancak, MQOPEN çağrısıyla MQOO_SET seçeneğini kullanarak bir model kuyruğu açarsanız, MQOPEN çağrısıyla yaratılan dinamik yerel kuyruğun özniteliklerini ayarlamak için MQSET çağrısı kullanabilirsiniz.
5. Ayarlanmakta olan nesne bir küme kuyruğuysa, açmanın başarılı olması için küme kuyruğunun yerel bir eşgörünümü olmalıdır.

Nesne öznitelikleriyle ilgili ek bilgi için aşağıdaki konuya bakın:

- [“Kuyruklara ilişkin öznitelikler” sayfa 826](#)
- [“Ad listesi öznitelikleri” sayfa 858](#)
- [“Süreç tanımlamalarına ilişkin öznitelikler” sayfa 860](#)
- [“Kuyruk yöneticisine ilişkin öznitelikler” sayfa 789](#)

C çağrısı

```
MQSET (Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttrs,
CharAttrLength, CharAttrs, &CompCode, &Reason);
```

Değiştirelileri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağrısı

```
CALL 'MQSET' USING HCONN, HOBJ, SELECTORCOUNT, SELECTORS-TABLE,
INTATTRCOUNT, INTATTRS-TABLE, CHARATTRLENGTH,
CHARATTRS, COMPCODE, REASON.
```

Değiştirelileri aşağıdaki gibi bildirin:

```
** 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.
```

PL/I çağırısı

```
call MQSET (Hconn, Hobj, SelectorCount, Selectors, IntAttrCount,
            IntAttr, CharAttrLength, CharAttr, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 IntAttr(n)     fixed bin(31); /* Array of integer attributes */
dcl CharAttrLength fixed bin(31); /* Length of character attributes
                                buffer */
dcl CharAttr       char(n);       /* Character attributes */
dcl CompCode       fixed bin(31); /* Completion code */
dcl Reason         fixed bin(31); /* Reason code qualifying
                                CompCode */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQSET, (HCONN,HOBJ,SELECTORCOUNT,SELECTORS,INTATTRCOUNT, X
            INTATTRS,CHARATTRLENGTH,CHARATTRS,COMPCODE,REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

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

Visual Basic çağırısı

```
MQSET Hconn, Hobj, SelectorCount, Selectors, IntAttrCount, IntAttr,
      CharAttrLength, CharAttr, CompCode, Reason
```

Değiştirgeleri aşağıdaki gibi bildirin:

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 IntAttr	As Long	'Array of integer attributes'
Dim CharAttrLength	As Long	'Length of character attributes buffer'
Dim CharAttr	As String	'Character attributes'
Dim CompCode	As Long	'Completion code'
Dim Reason	As Long	'Reason code qualifying CompCode'

MQSETMP-İleti özelliğini ayarla

İleti tanıtıcısının bir özelliğini ayarlamak ya da değiştirmek için MQSETMP çağırısını kullanın.

Sözdizimi

MQSETMP (*Hconn, Hmsg, SetPropOpts, Name, PropDesc, Type, ValueLength, Value, Compcode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıttıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir.

Değer, **Hmsg** değiştirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eşleşmelidir. İleti tanıtıcısı MQHC_UNASSOCIATED_HCONN kullanılarak yaratıldıysa, ileti tanıtıcısının bir özelliğini ayarlarken iş parçacığında geçerli bir bağlantı kurulmalıdır; tersi durumda, çağrı MQRC_CONNECTION_BROKEN neden koduyla başarısız olur.

İt

Tip: MQHMSG-giriş

Bu, değiştirilecek ileti tanıtıcısıdır. Değer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

SetPropSeçmeler

Tip: MQSMPO-giriş

İleti özelliklerinin nasıl ayarlandığını denetleyin.

Bu yapı, uygulamaların ileti özelliklerinin nasıl ayarlanacağını denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQSETMP çağrısındaki bir giriş değiştirgesidir. Daha fazla bilgi için bkz. [MQSMPO](#).

Ad

Tip: MQCHARV-giriş

Bu, ayarlanacak özelliğin adıdır.

Özellik adlarının kullanımıyla ilgili ek bilgi için [Özellik adları](#) ve [Özellik adı kısıtlamaları](#) konusuna bakın.

PropDesc

Tip: MQPD-giriş/çıkış

Bu yapı, aşağıdakiler de içinde olmak üzere bir özelliğin özniteliklerini tanımlamak için kullanılır:

- özellik desteklenmiyorsa ne olur
- özelliğin ait olduğu ileti bağlamı
- Akarken özelliğin hangi iletilere kopyalandığı

Bu yapıyla ilgili ek bilgi için [MQPD](#) başlıklı konuya bakın.

Tip

Tip: MQLONG-input

Ayarlanmakta olan özelliğin veri tipi. Aşağıdakilerden biri olabilir:

MQTYPE_BOOLEAN

Bir Boole. *ValueLength* 4 olmalıdır.

MQTYPE_BYTE_STRING

Bir bayt dizilimi. *ValueLength* sıfır ya da daha büyük olmalıdır.

MQTYPE_INT8

8 bitlik işaretli bir tamsayı. *ValueLength* 1 olmalıdır.

MQTYPE_INT16

16 bit işaretli tamsayı. *ValueLength* 2 olmalıdır.

MQTYPE_INT32

32 bit işaretli tamsayı. *ValueLength* 4 olmalıdır.

MQTYPE_INT64

64 bitlik imzalı bir tamsayı. *ValueLength* 8 olmalıdır.

MQTYPE_FLOAT32

32 bit kayan noktalı sayı. *ValueLength* 4 olmalıdır.

Not: Bu tip, IBM COBOL for z/OSkullanan uygulamalarda desteklenmez.

MQTYPE_FLOAT64

64 bit kayan noktalı sayı. *ValueLength* 8 olmalıdır.

Not: Bu tip, IBM COBOL for z/OSkullanan uygulamalarda desteklenmez.

MQTYPE_STRING

Bir karakter dizgisi. *ValueLength* sıfır ya da daha büyük ya da MQVL_NULL_TERMINATED özel değeri olmalıdır.

MQTYPE_NULL (boş değerli)

Özellik var, ancak boş değerli. *ValueLength* sıfır olmalıdır.

ValueLength

Tip: MQLONG-input

Değer parametresindeki özellik değerinin bayt cinsinden uzunluğu. Sıfır yalnızca boş değerler, dizgiler ya da byte dizgileri için geçerlidir. Sıfır, özelliğin var olduğunu, ancak değer karakter ya da bayt içermediğini gösterir.

Tip değiştirilmesinde MQTYPE_STRING ayarlandıysa, değer sıfırdan büyük ya da sıfıra eşit ya da aşağıdaki özel değer olmalıdır:

MQVL_NULL_TERMINATED

Değer, dizgide karşılaşılan ilk boş değerle sınırlandı. Boş değer, dizginin bir parçası olarak içerilmez. MQTYPE_STRING de ayarlanmamışsa bu değer geçersizdir.

Not: MQVL_NULL_TERMINATED değeri, değer karakter kümesindeki boş değerse, bir dizgiyi sonlandırmak için kullanılan boş karakter.

Değer

Tip: MQBYTExValueUzunluk-giriş

Ayarlanacak özelliğin değeri. Arabellek, değerdeki verilerin niteliğine uygun bir sınırdan hizalanmalıdır.

C programlama dilinde, parametre bir gösterge-to-void olarak bildirilir; herhangi bir veri tipinin adresi parametre olarak belirtilebilir.

ValueLength sıfır ise, *Değer* seçeneğine başvuramaz. Bu durumda, C ya da System/390 çevirici dilinde yazılmış programlardan geçirilen parametre adresi boş olabilir.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_RFH_FORMAT_HATA

(2421, X'0975 ') Özellikleri içeren bir MQRFH2 klasörü ayrıştırılmadı.

CompCode MQCC_FAILED ise:

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'089C') Bağdaştırıcı kullanılamıyor.

MQRC_ADAPTER_SERV_LOAD_ERROR (UYGULAMA_HATASI)

(2130, X'852') Bağdaştırıcı hizmet modülü yüklenemiyor.

MQRC_ASID_MISMATCH

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

MQRC_BUFFER_ERROR

(2004, X'07D4') Değer parametresi geçersiz.

MQRC_BUFFER_LENGTH_ERROR

(2005, X'07D5') Değer uzunluğu parametresi geçersiz.

MQRC_CALL_IN_PROGRESS

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_HMSG_HATA

(2460, X'099C') İleti tanıtıcısı göstergesi geçersiz.

MQRC_MSG_HANDLE_IN_USE

(2499, X'09C3') İleti tanıtıcısı kullanımda.

MQRC_OPTIONS_HATA

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_PD_ERROR (QRC_PD_HATA)

(2482, X'09B2') Özellik tanımlayıcı yapısı geçersiz.

MQRC_NAMETY_NAME_ERROR

(2442, X'098A') Özellik adı geçersiz.

MQRC_19TY_TYPE_HATA

(2473, X'09A9') Özellik veri tipi geçersiz.

MQRC_PROP_NUMBER_FORMAT_HATA

(2472, X'09A8') Değer verilerinde sayı biçimi hatası saptandı.

MQRC_SMPO_HATA

(2463, X'099F') İleti özellik seçenekleri yapısını ayarla geçerli değil.

MQRC_SOURCE_CCSID_HATA

(2111, X'083F') Özellik adı kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

C çağrısı

```
MQSETMP (Hconn, Hmsg, &SetPropOpts, &Name, &PropDesc, Type, ValueLength, &Value, &CompCode, &Reason);
```

Değiştiregeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağırısı

```
CALL 'MQSETMP' USING HCONN, HMSG, SETMSGOPTS, NAME, PROPDSC, TYPE,
                    VALUELENGTH, VALUE, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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 PROPDSC.
   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.
```

PL/I çağırısı

```
call MQSETMP (Hconn, Hmsg, SetPropOpts, Name, PropDesc, Type, ValueLength,
              Value, CompCode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
dcl Hconn      fixed bin(31); /* Connection handle */
dcl Hmsg       fixed bin(63); /* Message handle */
dcl SetPropOpts like MQSMPO; /* 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 */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQSETMP, (HCONN,HMSG,SETMSGHOPTS,NAME,PROPDSC,TYPE,VALUELENGTH,
              VALUE,COMPCODE,REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

HCONN	DS	F	Connection handle
HMSG	DS	D	Message handle
SETMSGOPTS	CMQSMPOA	,	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-Durum bilgilerini al

Durum bilgilerini almak için MQSTAT çağrısına bakın. Döndürülen durum bilgisi tipi, çağrıda belirtilen Tip değerine göre belirlenir.

Sözdizimi

MQSTAT (*Hconn, Type, Stat, Compcode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve *Hconn* için belirtilen şu değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

Tip

Tip: MQLONG-input

İstenen durum bilgilerinin tipi. > Geçerli değerler şunlardır:

MQSTAT_TYPE_ASYNC_HATASI

Önceki zamanuyumsuz koyma işlemlerine ilişkin bilgileri döndürür.

MQSTAT_TYPE_RECONNECTION

Yeniden bağlantıyla ilgili bilgileri döndürür. Bağlantı yeniden bağlanıyorsa ya da yeniden bağlanamıyorsa, bilgiler bağlantının yeniden bağlanmaya başlamasına neden olan hatayı açıklar.

Bu değer yalnızca istemci bağlantıları için geçerlidir. Diğer bağlantı tiplerinde, çağrı

MQRC_ENVIRONMENT_ERROR neden koduyla başarısız olur.

MQSTAT_TYPE_RECONNECTION_HATA

Yeniden bağlanma ile ilgili önceki bir hataya ilişkin bilgileri döndürür. Bağlantı yeniden bağlanamazsa, bilgiler yeniden bağlantının başarısız olmasına neden olan hatayı açıklar.

Bu değer yalnızca istemci bağlantıları için geçerlidir. Diğer bağlantı tiplerinde, çağrı

MQRC_ENVIRONMENT_ERROR neden koduyla başarısız olur.

Durum

Tip: MQSTS-giriş/çıkış

Durum bilgisi yapısı. Ayrıntılar için bkz. "[MQSTS-Durum raporlama yapısı](#)" sayfa 585.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_API_EXIT_ERROR (UYGULAMA_HATASI)

(2374, X' 946 ') API çıkışı başarısız oldu

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

(2183, X'887 ') API çıkışı yüklenemiyor.

MQRC_CALL_IN_PROGRESS

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

MQRC_CONNECTION_BROKEN

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

MQRC_CONNECTION_DURDURULUYOR

(2203, X'89B') Bağlantı kapatılıyor.

MQRC_FUNCTION_NOT_SUPPORTED

(2298, X'8FA') İstenen işlem yürürlükteki ortamda kullanılamıyor.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_Q_MGR_DURDURULUYOR

(2162, X'872')-Kuyruk yöneticisi durduruluyor

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_STAT_TYPE_ERROR

(2430, X'97E' MQSTAT tipiyle ilgili hata

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_STS_HATA

(2426, X'97A') MQSTS yapısıyla ilgili hata

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

1. MQSTAT tipine ilişkin bir MQSTAT_TYPE_ASYNC_ERROR çağrısı, önceki zamanuyumsuz MQPUT ve MQPUT1 işlemleriyle ilgili bilgileri döndürür. MQSTAT çağrısından dönüşte iletilen MQSTS yapısı, bu bağlantıya ilişkin ilk kaydedilen zamanuyumsuz uyarı ya da hata bilgilerini içerir. Daha fazla hata ya da uyarı ilkini izlerse, normalde bu değerleri değiştirmezler. Ancak, MQCC_WARNINGtamamlama koduyla bir hata oluşursa, bunun yerine MQCC_FAILED tamamlama koduyla sonraki bir hata döndürülür.
2. Bağlantı kurulduğundan bu yana ya da son MQSTAT çağrısından bu yana herhangi bir hata oluşmamışsa, MQSTS yapısında bir CompCode / MQCC_OK ve MQRC_NONE Nedeni döndürülür.
3. Bağlantı tanıtıcısı altında işlenen zamanuyumsuz çağrılarının sayısı, üç sayaç alanı (PutSuccessCount, PutWarningCount ve PutFailureCount) aracılığıyla döndürülür. Bu sayaçlar, zamanuyumsuz bir işlem her başarıyla işlendiğinde, bir uyarı aldığı anda ya da başarısız olduğunda kuyruk yöneticisi tarafından artırılır (muhasabe amacıyla bir dağıtım listesine konan sayıların, dağıtım listesi başına bir kereden fazla değil, hedef kuyruk başına bir kez sayıldığını unutmayın). Bir sayaç, AMQ_LONG_MAXpozitif değeri üst sınırının üzerinde artırılmaz.

4. MQSTAT ' e yapılan başarılı bir çağrı, önceki hata bilgileriyle ya da sayılarla sonuçlanır.

5. MQSTAT ' in davranışı, sağladığınız **MQSTAT Type** parametresinin değerine bağlıdır.

6. MQSTAT_TYPE_ASYNC_HATASI

- MQSTAT tipine ilişkin bir MQSTAT_TYPE_ASYNC_ERROR çağrısı, önceki zamanuyumsuz MQPUT ve MQPUT1 işlemleriyle ilgili bilgileri döndürür. MQSTAT çağrısından dönüşte iletilen MQSTS yapısı, bu bağlantıya ilişkin ilk kaydedilen zamanuyumsuz uyarı ya da hata bilgilerini içerir. Daha fazla hata ya da uyarı ilkini izlerse, normalde bu değerleri değiştirmezler. Ancak, MQCC_WARNINGtamamlama koduyla bir hata oluşursa, bunun yerine MQCC_FAILED tamamlama koduyla sonraki bir hata döndürülür.
- Bağlantı kurulduğundan bu yana ya da son MQSTAT çağrısından bu yana herhangi bir hata oluşmamışsa, MQSTS yapısında bir CompCode / MQCC_OK ve MQRC_NONE Nedeni döndürülür.
- Bağlantı tanıtıcısı altında işlenen zamanuyumsuz çağrılarının sayısı, üç sayaç alanı (PutSuccessCount, PutWarningCount ve PutFailureCount) aracılığıyla döndürülür. Bu sayaçlar, zamanuyumsuz bir işlem her başarıyla işlendiğinde, bir uyarı aldığı anda ya da başarısız olduğunda kuyruk yöneticisi tarafından artırılır (muhasabe amacıyla bir dağıtım listesine konan sayıların, dağıtım listesi başına bir kereden fazla değil, hedef kuyruk başına bir kez sayıldığını unutmayın). Bir sayaç, AMQ_LONG_MAXpozitif değeri üst sınırının üzerinde artırılmaz.
- MQSTAT ' e yapılan başarılı bir çağrı, önceki hata bilgileriyle ya da sayılarla sonuçlanır.

MQSTAT_TYPE_RECONNECTION

Yeniden bağlantı sırasında bir olay işleyici içinde MQSTAT ' i Type ayarı MQSTAT_TYPE_RECONNECTION olarak çağırdığınızı varsayın. Bu örnekleri göz önünde bulundurun.

İstemci yeniden bağlantı kurmayı deniyor ya da yeniden bağlanamadı.

MQSTS yapısında CompCode MQCC_FAILED ve Reason MQRC_CONNECTION_BROKEN ya da MQRC_Q_MGR_QUIESCINGolabilir. ObjectType MQOT_Q_MGR, ObjectName kuyruk yöneticisinin adıdır ve ObjectQMgrName boştur.

İstemci yeniden bağlantıyı başarıyla tamamladı ya da bağlantısı kesilmedi.

CompCode MQSTS yapısında MQCC_OK ve Reason MQRC_NONE

MQSTAT ' e yapılan sonraki çağrılar aynı sonuçları döndürür.

MQSTAT_TYPE_RECONNECTION_HATA

Bir MQI çağrısını MQRC_RECONNECT_FAILED almaya yanıt olarak Type ayarı MQSTAT_TYPE_RECONNECTION_ERROR olan MQSTAT ögesini çağırdığınızı varsayın. Bu örnekleri göz önünde bulundurun.

Farklı bir kuyruk yöneticisiyle yeniden bağlantı kurulurken bir kuyruk yeniden açılırken yetki hatası oluştu.

MQSTS yapısındaki CompCode MQCC_FAILED ve Reason yeniden bağlantının başarısız olmasının nedenidir; örneğin, MQRC_NOT_AUTHORIZED. ObjectType , soruna neden olan nesnenin tipidir; örneğin, MQOT_QUEUE, ObjectName kuyruğun adıdır ve ObjectQMgrName kuyruğun sahibi olan kuyruk yöneticisinin adıdır.

Yeniden bağlantı sırasında bir yuva bağlantı hatası oluştu.

MQSTS yapısındaki CompCode MQCC_FAILED ve Reason yeniden bağlantının başarısız olmasının nedenidir; örneğin, MQRC_HOST_NOT_AVAILABLE. ObjectType MQOT_Q_MGR, ObjectName kuyruk yöneticisinin adıdır ve ObjectQMgrName boştur.

MQSTAT ' e yapılan sonraki çağrılar aynı sonuçları döndürür.

C çağrısı

```
MQSTAT (Hconn, StatType, &Stat, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
MQHCONN Hconn;          /* Connection Handle */
MQLONG StatType;       /* Status type */
MQSTS Stat;            /* Status information structure */
MQLONG CompCode;       /* Completion code */
MQLONG Reason;         /* Reason code qualifying CompCode */
```

COBOL çağırısı

```
CALL 'MQSTAT' USING HCONN, STATTYPE, STAT, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
**      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.
```

PL/I çağırısı

```
call MQSTAT (Hconn, StatType, Stat, Compcode, Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 Assembler çağırısı

```
CALL MQSTAT,(HCONN,STATTYPE,STAT,COMPCODE,REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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-Aboneliğin kaydedili

Belirli bir konuya ilişkin uygulama aboneliğini kaydettirmek için MQSUB çağırısına bakın.

Sözdizimi

MQSUB (*Hconn*, *SubDesc*, *Hobj*, *Hsub*, *Compcode*, *Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıttıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve *Hconniçin* belirtilen şu değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıttıcısı.

SubDesc

Tip: MQSD-giriş/çıkış

Bu, uygulama tarafından kaydedilmekte olan, kullanılmakta olan nesneyi tanımlayan bir yapıdır. Ek bilgi için bkz. [“MQSD-Abonelik tanımlayıcısı” sayfa 561](#).

Hobj

Tip: MQHOBJ-giriş/çıkış

Bu tanıttıcı, bu aboneliğe gönderilen iletileri almak için oluşturulan erişimi temsil eder. Bu iletiler belirli bir kuyruksa saklanabilir ya da kuyruk yöneticisi saklama alanını belirli bir kuyruk kullanmadan yönetebilir.

Belirli bir kuyruğu kullanmak için, abonelik yaratıldığında bunu abonelik ile ilişkilendirmeniz gerekir. Bunu iki şekilde yapabilirsiniz:

- DEFINE SUB MQSC komutunu kullanarak ve o komutu bir kuyruk nesnesi adıyla belirterek.
- MQSO_CREATE ile MQSUB çağrılırken bu tanıttıcıyı sağlayarak

Bu tanıttıcı, çağrıda bir giriş değıştirgesi olarak sağlandıysa, aşağıdaki seçeneklerden en az birini kullanan bir kuyruğun önceki bir MQOPEN çağrısından döndürülen geçerli bir nesne tanıttıcısı olmalıdır:

- MQOO_INPUT_*
- MQOO_GÖZ AT
- MQOO_OUTPUT (kuyruk uzak kuyruksa)

Durum böyle değilse, çağrı MQRC_HOBJ_ERROR ile başarısız olur. Bir konu nesnesine çözülen diğer ad kuyruğuna ilişkin bir nesne tanıttıcısı olamaz. Bu durumda, çağrı MQRC_HOBJ_ERROR ile başarısız olur.

Kuyruk yöneticisi bu aboneliğe gönderilen iletilerin saklanmasını yönetirse, bu, aboneliği yarattığınızda MQSO_MANAGED seçeneği kullanılarak ayarlanmalıdır. Kuyruk yöneticisi daha sonra bu tanıttıcıyı çağrıda bir çıkış parametresi olarak döndürür. Döndürülen tutamaç, yönetilen tanıttıcı olarak bilinir. MQHO_NONE belirtilirse, ancak MQSO_MANAGED belirtilmezse, çağrı MQRC_HOBJ_ERROR ile başarısız olur.

Kuyruk yöneticisi tarafından yönetilen bir tanıttıcı size döndürüldüğünde, bunu göz atma seçenekleriyle ya da göz atma seçenekleri olmadan, MQINQ çağrısında ya da MQCLOSE üzerinde bir MQGET ya da MQCB çağrısında kullanabilirsiniz. Bunu MQPUT, MQSUB, MQSET üzerinde kullanamazsınız; bunu yapma girişimi MQRC_NOT_OPEN_FOR_OUTPUT, MQRC_HOBJ_ERROR ya da MQRC_NOT_OPEN_FOR_SET ile başarısız olur.

Bu abonelik MQSD yapısındaki MQSO_RESUME seçeneği kullanılarak sürdürülüyorsa, MQSO_MANAGED MQHO_NONE olarak ayarlanarak tanıttıcı bu değıştirmede uygulamaya döndürülebilir. Aboneliğin yönetilen bir tanıttıcı kullanıp kullanmadığı ve DEFINE SUB kullanılarak yaratılan aboneliklerin, o komutta tanımlanan abonelik kuyruğuna ilişkin tanıttıcı ile birlikte sağlanması yararlı olabilir. Denetimci olarak yaratılan bir aboneliğin sürdürülmesi durumunda, kuyruk MQOO_INPUT_AS_Q_DEF ve MQOO_BROWSE ile açılır. Diğer seçenekleri belirtmeniz gerekirse, uygulamanın abonelik kuyruğunu açık olarak açması ve çağrıda nesne tanıttıcısını sağlaması gerekir. Kuyruk açılırken bir sorun ortaya çıkarsa, çağrı MQRC_INVALID_DESTINATION ile başarısız olur.

Hobj sağlanırsa, özgün MQSUB çağrısındaki *Hobj* ile eşdeğer olmalıdır. Bu, MQOPEN çağrısından döndürülen bir nesne tanıtıcısı sağlandığı anlamına gelir; tanıtıcı, daha önce kullanılan kuyruk ile aynı olmalıdır. Aynı kuyruk değilse, çağrı MQRC_HOBJ_ERROR ile başarısız olur.

Bu abonelik MQSD yapısında MQSO ALTER seçeneği kullanılarak değiştiriliyorsa, farklı bir *Hobj* sağlanabilir. Kuyruğa teslim edilen ve bu parametre aracılığıyla daha önce tanımlanan yayınlar o kuyrukta kalır ve **Hobj** parametresi artık farklı bir kuyruğu gösteriyorsa, bu iletileri almak uygulamanın sorumluluğundadır.

Çizelge 555. Çeşitli abonelik seçenekleriyle hobj kullanma		
Seçenekler	Hobj	Açıklama
MQSO_CREATE + MQSO_MANAGED	Girişte yoksayıldı	Kuyruk yöneticisi tarafından yönetilen iletilerin depolandığı bir abonelik oluşturur
MQSO_OLUŞTURMA	Geçerli bir nesne tanıtıcısı	İletilerin hedefi olarak belirli bir kuyruğu sağlayan bir abonelik oluşturur.
MQSO_RESUME	MQHO_NONE	Yönetilen olsun ya da olmasın, önceden yaratılmış bir aboneliği sürdürür ve kuyruk yöneticisinin uygulama tarafından kullanılmak üzere nesne tanıtıcısını döndürmesini sağlar.
MQSO_RESUME	Geçerli, eşleşen bir nesne tanıtıcısı	İletiler için hedef olarak belirli bir kuyruğu kullanan önceden oluşturulmuş bir aboneliği sürdürür ve belirli açık seçeneklerle bir nesne tanıtıcısı kullanır.
MQSO_ALTER + MQSO_MANAGED	MQHO_NONE	Önceden belirli bir kuyruğu kullanan var olan bir aboneliği değiştirir, bu nedenle artık yönetilen bir abonelik olur. Hedef sınıfı (yönetilen ya da yönetilmeyen) değiştirilemez.
MQSO_ALTER	Geçerli bir nesne tanıtıcısı	Yönetilse de yönetilmese de var olan bir aboneliği, artık belirli bir kuyruğu kullanacak şekilde değiştirir. MQSO_MANAGED seçeneği kullanılmadığında, sağlanan kuyruk değiştirilebilir, ancak hedef sınıfı (yönetilen ya da yönetilmeyen) değiştirilemez.

Sağlandığı ya da döndürüldüğü fark etmez, *Hobj*, bu aboneliğe gönderilen yayınlama iletilerini almak isteyen sonraki MQGET ya da MQCB çağrılarında belirtilmelidir.

Hobj tanıtıcısı, MQCLOSE çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde (uygulama bağlantısı kesilinceye kadar) geçerli olmaz. Döndürülen nesne tanıtıcısının kapsamı, çağrıda belirlenen bağlantı tanıtıcısıyla aynı. Tanıtıcı kapsamıyla ilgili bilgi için [Hconn \(MQHCONN\)-output](#) başlıklı konuya bakın. *Hobj* tanıtıcısının MQCLOSE, *Hsub* tanıtıcısını etkilemez.

Hsub.

Tip: MQHOBJS-output

Bu tanıtıcı, yapılan aboneliği gösterir. İki işlem daha için kullanılabilir:

- Abonelik oluşturulurken MQSO_PUBLICATIONS_ON_REQUEST seçeneği kullanıldığında yayınların gönderilmesini istemek için sonraki bir MQSUBRQ çağrısında kullanılabilir.
- Yapılan aboneliği kaldırmak için sonraki bir MQCLOSE çağrısında kullanılabilir. *Hsub* tanıtıcısı, MQCLOSE çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde geçerli olmaktan çıkar. Döndürülen nesne tanıtıcısının kapsamı, çağrıda belirlenen bağlantı tanıtıcısıyla aynı. *Hsub* tanıtıcısının MQCLOSE, *Hobj* tanıtıcısını etkilemez.

Bu tanıtıcı bir MQGET ya da MQCB çağrısına geçirilemez. **Hobj** parametresini kullanmalısınız. MQCLOSE ya da MQSUBRQ dışında bir IBM MQ çağrısında bu tanıtıcıyı kullanamazsınız. Bu tanıtıcı değerinin MQRC_HOBJ_ERROR içindeki diğer IBM MQ çağrılarında geçirilmesi.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı

MQCC_UYARISI

Uyarı (kısmi tamamlama)

MQCC_FAILED

Arama başarısız oldu

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise, neden kodu aşağıdaki gibidir:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise, neden kodu aşağıdakilerden biridir:

MQRC_CLUSTER_RESOLUTION_HATA

(2189, X'88D') Küme adı çözümlemesi başarısız oldu.

MQRC_DURABILITY_NOT_ALLOWED

2436 (X'0984 ') MQSO_DURABLE seçeneği kullanılarak MQSUB çağrısı başarısız oldu.

MQRC_FUNCTION_NOT_SUPPORTED

2298 (X'08FA') İstenen işlev yürürlükteki ortamda kullanılamıyor.

MQRC_HOBJ_HATA

2019 (X'07E3') Nesne tanıtıcısı Hobj geçersiz.

MQRC_IDENTITY_MISMATCH

2434 (X'0982 ') Abonelik adı, var olan abonelikle eşleşiyor.

MQRC_NOT_YETKILI

2035 (X'07F3') Kullanıcının işlemi gerçekleştirme yetkisi yok.

MQRC_NO_ABONELIK

2428 (X'097C') Tanımlanmış abonelik adı yok.

MQRC_OBJECT_STRING_HATA

2441 (X'0989 ') Nesne dizgisi alanı geçersiz.

MQRC_OPTIONS_HATA

2046 (X'07FE') Options parametresi ya da alanı geçerli olmayan seçenekler ya da geçerli olmayan seçenekler birleşimi içeriyor.

MQRC_Q_MGR QUIESCING

2161 (X'0871 ') Kuyruk yöneticisi susturuyor.

MQRC_RECONNECT_Q_MGR REQD

2555 (X'09FB' X) MQCNO_RECONNECT_Q_MGR seçeneği gereklidir.

MQRC_RETAINED_MSG_Q_ERROR

2525 (X'09DD') Abone olunan konu dizgisi için var olan yayınlar alınamıyor.

MQRC_RETAINED_NOT_TESLIM edilemedi

2526 (X'09DE') Abone olunan konu dizisi için var olan alıkonan yayınlar, abonelik hedef kuyruğuna teslim edilemez ve teslim edilmez.

MQRC_SD_ERROR

2424 (X'0978 ') Abonelik tanımlayıcısı (MQSD) geçerli değil.

MQRC_SELECTION_NOT_KULLANILAMIYOR

2551 (X'09F7') Seçim dizgisi IBM MQ seçici sözdizimini izlemiyor ve genişletilmiş ileti seçimi sağlayıcısı yok.

MQRC_SELECTION_STRING_HATA

2519 (X'09D7') Seçim dizgisi, MQCHARV yapı belgelerinde açıklandığı gibi belirtilmelidir.

MQRC_SELECTOR_SYNTAX_ERROR

2459 (X'099B') Bir MQOPEN, MQPUT1ya da MQSUB çağrısı yayınlandı, ancak sözdizimi hatası içeren bir seçim dizgisi belirtildi.

MQRC_SUB_USER_DATA_ERROR (QRC_SUB_USER_DATA_HATA)

2431 (X'097F') SubUserVeri alanı geçersiz.

MQRC_SUB_NAME_ERROR (MQRC_AD_HATASI)

2440 (X'0988 ') SubName alanı geçersiz.

MQRC_SUB_ALREADY_EXISTS

2432 (X'0980 ') Aboneliği zaten var.

MQRC_SUB_USER_DATA_ERROR (QRC_SUB_USER_DATA_HATA)

2431 (X'097F') SubUserVeri alanı geçersiz.

MQRC_TOPIC_STRING_HATASI

2425 (X'0979 ') Konu dizgisi geçerli değil.

MQRC_UNKNOWN_OBJECT_NAME

2085 (X'0825 ') MQSD ObjectName alanında tanımlanmış nesne bulunamıyor.

MQRC_SUB_JOIN_NOT_ALTERABLE

29440 (X'7300 ') Abonelik paylaşım kipi var olan abonelikle uyumlu değil. JMS dışı bir uygulamada JMS 2.0 paylaşılan aboneliğini sürdürme girişimi sırasında bu hata döndürülebilir.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

- Abonelik, önceden tanımlanmış bir konu nesnesinin kısa adı, konu dizgisinin tam adı kullanılarak ya da iki kısmın birleştirilmesiyle oluşturulur. “MQSD-Abonelik tanımlayıcısı” sayfa 561’inde *ObjectName* ve *ObjectString* açıklamasına bakın.
- Kuyruk yöneticisi, bir MQSUB çağrısı yayınlandığında, uygulamanın çalıştırıldığı kullanıcı kimliğinin erişime izin verilmeden önce uygun yetki düzeyine sahip olduğunu doğrulamak için güvenlik denetimleri gerçekleştirir. Uygun konu nesnesi konu sıradüzeninde bulunur ve abone olma yetkisinin ayarlandığından emin olmak için bu konu nesnesi üzerinde bir yetki denetimi yapılır. MQSO_MANAGED seçeneği kullanılmazsa, çıkışa ilişkin yetkinin ayarlandığından emin olmak için hedef kuyrukta bir yetki denetimi yapılır. MQSO_MANAGED seçeneği kullanılırsa, yönetilen kuyrukta çıkış ya da sorgu erişimi için yetki denetimi yapılmaz.
- Giriş olarak bir Hobj belirtmezseniz, MQSUB çağrısı iki tanıtıcı, bir nesne tanıtıcısı (Hobj) ve bir abonelik tanıtıcısı (Hsub) ayırır.
- MQSO_MANAGED seçeneği kullanıldığında MQSUB çağrısında döndürülen Hobj, geriletme eşiği ve Aşırı geriletme yeniden kuyruğa alma adı gibi öznitelikleri bulmak için sorulabilir. Yönetilen kuyruğun adını da sorabilirsiniz, ancak bu kuyruğu doğrudan açmayı denememelisiniz.
- Abonelikler, birden fazla grubun yayınıla eşleştiği abonelikler grubuna yalnızca tek bir yayının teslim edilmesine izin vererek gruplandırılabilir. Abonelikler MQSO_GROUP_SUB seçeneği kullanılarak gruplanır ve abonelikleri gruplamak için

- aynı adlandırılmış kuyruğu (MQSO_MANAGED seçeneğini kullanmayan) aynı kuyruk yöneticisinde kullanma-MQSUB çağrısında Hobj değiştirilmesiyle gösterilir
- aynı SubCorrelkimliğini paylaş
- aynı SubLevel olmalıdır

Bu öznitelikler, grupta olduğu kabul edilen abonelik kümesini tanımlar ve bir abonelik gruplandığında değiştirilemeyen özniteliklerdir. SubLevel ' in değiştirilmesi, MQRC_SUBLEVEL_NOT_ALTERABLE ile sonuçlanır ve diğerlerinden herhangi birinin (abonelik gruplanmamışsa değiştirilebilir) değiştirilmesi MQRC_GROUPING_NOT_ALTERABLE ile sonuçlanır.

- MQSUB çağrısının başarıyla tamamlanması, işlemin tamamlandığı anlamına gelmez. Bu çağrının tamamlanıp tamamlanmadığını denetlemek için Dağıtılmış ağlara ilişkin zamanuyumsuz komutların tamamlanıp tamamlanmadığını denetleme başlıklı konudaki DEFINE SUB adımına bakın.
- MQSD ' deki alanlar, MQSO_RESUME seçeneğini kullanan bir MQSUB çağrısından dönüşte doldurulur. Döndürülen MQSD, MQSD ' ye uygulanan abonelikte yapmanız gereken değişikliklerle MQSO ALTER seçeneğini kullanan bir MQSUB çağrısına doğrudan geçirilebilir. Bazı alanlarda, çizelgede belirtildiği gibi, dikkat edilmesi gereken özel noktalar vardır.

Çizelge 556. MQSD ' deki alanlar için dikkat edilmesi gereken özel noktalar	
MQSD ' de alan adı	Dikkat edilmesi gereken özel noktalar
Erişim ya da yaratma seçenekleri	Bazı seçenekler, MQSUB çağrısından dönüşte ilk durumuna getirilir. MQSD ' yi bir MQSUB çağrısında yeniden kullanırsanız, gerekli seçeneğin belirttik olarak ayarlanması gerekir.
Dayanıklılık seçenekleri, Hedef seçenekleri, Kayıt Seçenekleri ve Genel Arama Kartı seçenekleri	Bu seçenekler uygun şekilde ayarlanır
Yayınlama seçenekleri	Bu seçenekler, yalnızca MQSO_CREATE için geçerli olan MQSO_NEW_PUBLICATIONS_ONLY dışında uygun olarak ayarlanır.
Diğer seçenekler	Bu seçenekler, bir MQSUB çağrısından dönüşte değiştirilmez. API çağrılarının nasıl yayınlandığını ve abonelikte birlikte nasıl saklanmadığını denetler. Bunlar, MQSD ' yi yeniden kullanan sonraki MQSUB çağrılarında gerektiği şekilde ayarlanmalıdır.
ObjectName	Bir MQSUB çağrısından dönüşte yalnızca bu giriş alanı değiştirilmez.
ObjectString	Bir MQSUB çağrısından dönüşte yalnızca bu giriş alanı değiştirilmez. Bir arabellek sağlanırsa, <i>ResObjectString</i> alanında kullanılan Tam konu adı döndürülür.
AlternateUserTanıtıcısı ve AlternateSecurityTanıtıcısı	Bir MQSUB çağrısından dönüşte yalnızca bu giriş alanları değişmez. API çağrılarının nasıl yayınlandığını ve abonelikte birlikte nasıl saklanmadığını denetler. Bunlar, MQSD ' yi yeniden kullanan sonraki MQSUB çağrılarında gerektiği gibi ayarlanmalıdır.
SubExpiry	MQSO_RESUME seçeneği kullanılarak bir MQSUB çağrısından dönüş yapılıyorsa, bu alan aboneliğin geri kalan süre bitimine değil, özgün süre bitimine ayarlanır. MQSO ALTER seçeneğini kullanarak MQSD ' yi bir MQSUB çağrısında yeniden kullanırsanız, aboneliğin süre bitimini yeniden saymaya başlayacak şekilde sıfırlarsınız.

Çizelge 556. MQSD ' deki alanlar için dikkat edilmesi gereken özel noktalar (devamı var)	
MQSD ' de alan adı	Dikkat edilmesi gereken özel noktalar
SubName	Bu alan, MQSUB çağrısındaki bir giriş alanıdır ve çıkışta değiştirilmez.
SubUserVeri ve SelectionString	Bu değişken uzunluklu alanlar, bir arabellek sağlandıysa, MQSO_RESUME seçeneği kullanılarak bir MQSUB çağrısının çıkışında ve <i>V\$BufSize</i> içinde artı bir arabellek uzunluğu kullanılarak döndürülür. Arabellek sağlanmazsa, MQCHARV ' nin <i>V\$Length</i> alanında yalnızca uzunluk döndürülür. Sağlanan arabellek, alanı döndürmek için gereken alandan küçükse, sağlanan arabellekte yalnızca <i>V\$BufSize</i> byte döndürülür. MQSD ' yi MQSO_ALTER seçeneğini kullanarak bir MQSUB çağrısında yeniden kullanırsanız ve bir arabellek sağlanmazsa, ancak sıfır dışında bir <i>V\$Length</i> sağlanırsa, bu uzunluk alanının var olan uzunluğuyla eşleşirse, alanda değişiklik yapılmaz.
SubCorrelTanıtıcısı ve PubAccountingSimgesi	MQSO_SET_CORREL_ID kullanmazsanız, <i>SubCorrelId</i> kuyruk yöneticisi tarafından oluşturulur. MQSO_SET_IDENTITY_CONTEXT kullanmazsanız, <i>PubAccountingToken</i> kuyruk yöneticisi tarafından oluşturulur. MQSD ' de bu alanlar, MQSO_RESUME seçeneği kullanılarak MQSUB çağrısından döndürülür. Bunlar kuyruk yöneticisi tarafından üretiliyorsa, oluşturulan değer MQSO_CREATE ya da MQSO_ALTER seçeneği kullanılarak bir MQSUB çağrısında döndürülür.
PubPriority, SubLevel & PubApplIdentityData	Bu alanlar MQSD ' de döndürülür.
ResObjectDizesi	Bir arabellek sağlanırsa, MQSD ' de yalnızca bu çıkış alanı döndürülür.

C çağırısı

```
MQSUB (Hconn, &SubDesc, &Hobj, &Hsub, &CompCode, &Reason)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağırısı

```
CALL 'MQSUB' USING HCONN, SUBDESC, HOBJ, HSUB, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

** 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.

```

PL/I çağırısı

```
call MQSUB (Hconn, SubDesc, Hobj, Hsub, CompCode, Reason)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

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 */

```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQSUB, (HCONN, SUBDESC, HOBJ, HSUB, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```

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-Abonelik isteği

Abone MQSO_PUBLICATIONS_ON_REQUEST ile kaydedildiğinde, alıkonan yayın için istekte bulunmak üzere MQSUBRQ çağırısı kullanın.

Sözdizimi

MQSUBRQ (*Hconn, Hsub, Action, SubRqOpts, Compcode, Reason*)

Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri önceki bir MQCONN ya da MQCONNX çağırısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağırısı atlanabilir ve *Hconn* için belirtilen şu değer:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

Hsub.

Tip: MQHOBj-giriş

Bu tanıtıcı, güncelleme isteğinde bulunulacak aboneliği gösterir. Önceki bir MQSUB çağrısından *Hsub* değeri döndürüldü.

Eylem

Tip: MQLONG-input

Bu parametre, abonelikte istenen işlemi denetler. Aşağıdaki değer belirtilmelidir:

MQSR_ACTION_YAYIN

Bu işlem, belirtilen konu için bir güncelleme yayınının gönderilmesini ister. Yalnızca abone, aboneliği yaptığında MQSUB çağrısında MQSO_PUBLICATIONS_ON_REQUEST seçeneğini belirttiğinde kullanılabilir. Kuyruk yöneticisinin konu için alıkonan bir yayını varsa, bu aboneye gönderilir. Yoksa, arama başarısız olur. Bir uygulamaya alıkonan bir yayın gönderilirse, bu, o yayının MQIsRetained ileti özelliğiyle gösterilir.

Hsub parametresi tarafından temsil edilen var olan abonelikteki konu joker karakterler içerebileceğinden, abone birden çok alıkonan yayın alabilir.

SubRqSeçmeler

Tip: MQSRO-giriş/çıkış

Bu seçenekler MQSUBRQ işlemini denetler, ayrıntılar için bkz. "[MQSRO-Abonelik isteği seçenekleri](#)" sayfa 583 .

Herhangi bir seçenek gerekmiyorsa, C ya da S/390 derleyicisinde yazılan programlar, bir MQSRO yapısının adresini belirtmek yerine boş bir değiştirge adresi belirtebilir.

CompCode

Tip: MQLONG-output

Tamamlanma kodu; aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı

MQCC_UYARISI

Uyarı (kısmi tamamlama)

MQCC_FAILED

Arama başarısız oldu

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_FUNCTION_NOT_SUPPORTED

2298 (X'08FA') İstenen işlev yürürlükteki ortamda kullanılamıyor.

MQRC_NO_RETAINED_MSG

2437 (X'0985 ') Bu konu için saklanan hiçbir yayın yok.

MQRC_OPTIONS_HATA

2046 (X'07FE') Options parametresi ya da alanı geçerli olmayan seçenekler ya da geçerli olmayan seçenekler birleşimi içeriyor.

MQRC_Q_MGR QUIESCING

2161 (X'0871 ') Kuyruk yöneticisi susturuyor.

MQRC_SRO_HATA

2438 (X'0986 ') MQSUBRQ çağrısında, Abonelik İsteği Seçenekleri MQSRO geçerli değil.

MQRC_RETAINED_MSG_Q_ERROR

2525 (X'09DD') Abone olunan konu dizgisi için var olan yayınlar alınamıyor.

MQRC_RETAINED_NOT_TESLIM edilemedi

2526 (X'09DE') Abone olunan konu dizesi için var olan alıkonan yayınlar, abonelik hedef kuyruğuna teslim edilemez ve teslim edilmez.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

Kullanım notları

Aşağıdaki kullanım notları, MQSR_ACTION_YAYINI işlem kodunun kullanımı için geçerlidir:

1. Bu komut başarıyla tamamlanırsa, belirtilen abonelik eşleşen alıkonan yayınlar aboneliğe gönderilir ve aboneliği yaratan özgün MQSUB fiilinde döndürülen Hobj kullanılarak MQGET ya da MQCB kullanılarak alınabilir.
2. Aboneliği yaratan özgün MQSUB komutu tarafından abone olunan konu bir genel arama karakteri içeriyorsa, birden çok alıkonan yayın gönderilebilir. Bu çağrı sonucunda gönderilen yayınların sayısı, SubRqOpts yapısındaki NumPubs alanında kaydedilir.
3. Bu komut MQRC_NO_RETAINED_MSG neden koduyla tamamlanırsa, belirtilen konu için şu anda alıkonan yayın yoktur. #
4. Bu komut MQRC_RETAINED_MSG_Q_ERROR ya da MQRC_RETAINED_NOT_ALLOWED neden koduyla tamamlanırsa, belirtilen konu için şu anda alıkonan yayınlar var, ancak teslim edilemedikleri anlamına gelen bir hata oluştu.
5. Bu çağrıyı yapabilmesi için uygulamanın konuya ilişkin geçerli bir aboneliği olmalıdır. Abonelik, uygulamanın önceki bir eşgörünümünde yapıldıysa ve abonelik için geçerli bir tanıtıcı yoksa, uygulamanın bu çağrıda kullanmak üzere bir tanıtıcı elde etmek için önce MQSUB ' u MQSO_RESUME seçeneğiyle çağırması gerekir.
6. Yayınlar, bu uygulamanın geçerli aboneliğiyle kullanılmak üzere kayıtlı hedefe gönderilir. Yayınların başka bir yere gönderilmesi gerekiyorsa, abonelik önce MQSO_ALTER seçeneğiyle MQSUB çağrısı kullanılarak değiştirilmelidir.

C çağrısı

```
MQSUB (Hconn, Hsub, Action, &SubRqOpts, &CompCode, &Reason)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL çağrısı

```
CALL 'MQSUBRQ' USING HCONN, HSUB, ACTION, SUBRQOPTS, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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 CMQSR0V.
** Completion code
01 COMPCODE PIC S9(9) BINARY.
** Reason code qualifying COMPCODE
01 REASON PIC S9(9) BINARY.
```

PL/I çağırısı

```
call MQSUBRQ (Hconn, Hsub, Action, SubRqOpts, CompCode, Reason)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 MQSR0; /* Options that control the action of MQSUBRQ */
dcl CompCode fixed bin(31); /* Completion code */
dcl Reason fixed bin(31); /* Reason code qualifying CompCode */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağırısı

```
CALL MQSUBRQ,(HCONN, HSUB, ACTION, SUBRQOPTS,COMPCODE,REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
HCONN DS F Connection handle
HSUB DS F Subscription handle
ACTION DS F Action requested by MQSUBRQ
SUBRQOPTS CMQSR0A , Options that control the action of MQSUBRQ
COMPCODE DS F Completion code
REASON DS F Reason code qualifying COMPCODE
```

Nesnelerin öznitelikleri

Bu konu derlemi, yalnızca MQINQ işlev çağırısının konusu olabilen IBM MQ nesnelerini listeler ve sorulabilecek özniteliklerin ve kullanılacak seçicilerin ayrıntılarını verir.

Kuyruk yöneticisine ilişkin öznitelikler

Bazı kuyruk yöneticisi öznitelikleri belirli somutlamalar için düzeltilmiştir; diğerleri ALTER QMGRMQSC komutu kullanılarak değiştirilebilir.

Öznitelikler, DISPLAY QMGRkomutu kullanılarak da görüntülenebilir. Çoğu kuyruk yöneticisi özniteliği, özel bir MQOT_Q_MGR nesnesi açılarak ve döndürülen tanıtıcı ile MQINQ çağırısı kullanılarak sorulabilir.

Aşağıdaki çizelge, kuyruk yöneticisine özgü öznitelikleri özetler. Öznitelikler alfabetik sırayla açıklanır.


Not: Bu kısımda gösterilen özniteliklerin adları MQINQ çağırısıyla kullanılan açıklayıcı adlardır; adlar PCF komutlarıyla aynıdır. Öznitelikleri tanımlamak, değiştirmek ya da görüntülemek için MQSC komutları kullanıldığında, diğer kısa adlar kullanılır; ek bilgi için [MQSC komutları](#) konusuna bakın.

Çizelge 557. Kuyruk yöneticisine ilişkin öznitelikler	
Öznitelik	Açıklama
AccountingConnGeçersiz Kıl	Muhasebe ayarlarını geçersiz kıl.
AccountingInterval	Ara muhasebe kayıtlarının ne sıklıkta yazılacağı.
ActivityConnOverride	Etkinlik ayarlarını geçersiz kıl.

Çizelge 557. Kuyruk yöneticisine ilişkin öznitelikler (devamı var)

Öznitelik	Açıklama
ActivityTrace	IBM MQ MQI uygulama etkinliği izlemesinin derlemesini denetler.
AdoptNewMCACheck	Yeni MCA 'nın benimsenip benimsenmeyeceğini belirlemek için öğeler denetlendi.
AdoptNewMCAType	Belirli bir kanal tipindeki bir MCA 'nın artık kalmış bir eşgörünümünün otomatik olarak yeniden başlatılıp başlatılmayacağını belirler.
AlterationDate	Tanımının en son değiştirildiği tarih
AlterationTime	Tanımlamanın en son değiştirildiği zaman
AuthorityEvent	Yetkilendirme (Yetkili Değil) olaylarının oluşturulup oluşturulmayacağını denetler
BridgeEvent	Köprü olayları için denetim özniteliği.
ChannelAutoDef	Otomatik kanal tanımına izin verilip verilmediğini denetler
ChannelAutoDefEvent	Kanal otomatik tanımlama olaylarının oluşturulup oluşturulmayacağını denetler
ChannelAutoDefExit	Otomatik kanal tanımı için kullanıcı çıkışının adı
ChannelEvent	Kanal olayları için denetim özniteliği.
ChannelInitiatorDenetimi	Kanal başlatıcı için denetim özniteliği
ChannelMonitoring	Kanallar için çevrimiçi izleme verileri
ChannelStatistics	Kanallar için istatistik verilerinin toplanmasını denetler.
ChinitAdapters	IBM MQ çağrılarını işlemek için bağdaştırıcı alt görevlerinin sayısı.
ChinitDispatchers	Kanal başlatıcısı için kullanılacak dağıtıcıların sayısı.
	IBM kullanımı için ayrılmıştır.
ChinitTraceAutoStart	Kanal başlatıcı izlemesinin otomatik olarak başlatılıp başlatılmayacağını belirler.
ChinitTraceTableSize	Kanal başlatıcısının izleme veri alanının boyutu.
ClusterSenderMonitoringDefault	Küme gönderen kanalları için çevrimiçi izleme verileri varsayılan değeri
ClusterSenderİstatistikleri	Küme gönderen kanalları için istatistik izleme bilgilerinin toplanmasını denetler.
ClusterWorkloadVerileri	Küme iş yükü çıkışına ilişkin kullanıcı verileri
ClusterWorkloadÇık	Küme iş yükü yönetimi için kullanıcı çıkışının adı
ClusterWorkloadUzunluğu	Küme iş yükü çıkışına geçirilen ileti verileri uzunluğu üst sınırı
CLWLMRUChannel	Küme iş yükü dengelemesi için en son kullanılan kanalların sayısı
CLWLUseQ	Küme iş yükü uzak kuyruk kullanıyor.
CodedCharSetId	Kodlanmış karakter takımı tanıtıcısı
CommandEvent	Komut olayları için denetim özniteliği.
CommandInputQName özniteliği	Komut giriş kuyruğu adı
CommandLevel	Komut düzeyi
CommandServerDenetim özniteliği	Komut sunucusu için denetim özniteliği.
Yapılandırma Olayı özniteliği	Yapılandırma olayları için denetim özniteliği.
DeadLetterQName	Gönderilmeyen iletiler kuyruğunun adı
DefClusterXmitQueueTip	Varsayılan küme iletim kuyruğu tipi
DefXmitQName	Varsayılan iletim kuyruğu adı
DistLists	Dağıtım listesi desteği
DNSGroup	İş Yükü Yöneticisi Dinamik Etki Alanı Adı Hizmetleri desteği kullanılırken TCP dinleyicisine ilişkin grubun adı.
DNSWLM	TCP dinleyicisinin Dynamic Domain Name Services için Workload Manager 'a kaydedilip kaydedilmeyeceğini belirler.
ExpiryInterval	Süresi dolan iletilere ilişkin taramalar arasındaki aralık
IGQPutAuthority	Grup içi kuyruğa alma koyma yetkisi
IGQUserId	Grup içi kuyruğa alma kullanıcı kimliği
InhibitEvent	Engelleme (Girmeyi Engelleme ve Engelleme) olaylarının oluşturulup oluşturulmayacağını denetler

Çizelge 557. Kuyruk yöneticisine ilişkin öznelikler (devamı var)

Öznelik	Açıklama
 InitialKey 1	Parola koruma sisteminin ilk anahtarı.
IPAddressVersion	Internet Protocol adresinin sürümü
IntraGroupqueuing	Grup içi kuyruğa alma desteği
ListenerTimer	APPC ya da TCP/IP hatasından sonra dinleyiciyi yeniden başlatma girişimleri arasındaki zaman aralığı.
LocalEvent	Yerel hata olaylarının oluşturulup oluşturulmayacağını denetler
LoggerEvent	Günlüğe kaydedici olaylarının oluşturulup oluşturulmayacağını denetler
LUGroupName	Kuyruk paylaşım grubuna ilişkin gelen iletileri işleyen LU 6.2 dinleyicisi için soysal LU adı.
LUAdı	Giden LU 6.2 iletileri için kullanılacak LU adı.
LU62ARMSuffix	SYS1.PARMLIB üyesi APPCPMxx.
LU62Channels	LU 6.2kullanan yürürlükteki kanal ya da bağlı istemci sayısı üst sınırı.
MaxActiveKanalları	Herhangi bir zamanda etkin olabilecek kanal sayısı üst sınırı.
MaxChannels	Geçerli kanal sayısı üst sınırı.
MaxHandles	Tanıtıcı sayısı üst sınırı
MaxMsgUzunluğu	İleti uzunluğu üst sınırı (byte)
MaxPriority özneliği	Öncelik üst sınırı
MaxPropertiesLength (Uzunluk Üst Sınırı)	Özellik verilerinin bayt cinsinden uzunluk üst sınırı
MaxUncommittedİletiler	Bir iş birimindeki kesinleştirilmemiş ileti sayısı üst sınırı
MQIAccounting	MQI verileri için muhasebe bilgileri toplamasını denetler.
MQIStatistics	Kuyruk yöneticisine ilişkin istatistik izleme bilgilerinin toplanmasını denetler.
MsgMarkBrowseInterval	Kuyruk yöneticisinin göz atılan iletilerden işareti kaldırabileceği aralık.
OutboundPortMin	<i>OutboundPortMin</i> ile, giden kanallar bağlanırken kullanılacak kapı numaraları aralığını tanımlar.
OutboundPortMax	<i>OutboundPortMax</i> ile, giden kanallar bağlanırken kullanılacak kapı numaraları aralığını tanımlar.
PerformanceEvent	Performansla ilgili olayların oluşturulup oluşturulmayacağını denetler
Hizmet olarak sunulan	Kuyruk yöneticisinin çalıştığı platform
PubSubNPInputMsg	Teslim edilmemiş bir giriş iletilerinin atılıp atılmayacağı (ya da alınıp tutulmayacağı)
PubSubNPResponse	Teslim edilmeyen davranışını denetler
PubSubMaxMsgRetryCount	Başarısız bir komut iletileri işlenirken (syncpoint altında) yeniden deneme sayısı
PubSubSyncPoint	Yalnızca kalıcı (ya da tüm) iletilerin eşitleme noktası altında işlenip işlenmeyeceğini belirler
PubSubKipi	Kuyruğa alınan yayınlama/abone olma arabiriminin çalışıp çalışmadığını belirler
QMgrDesc	Kuyruk yöneticisi açıklaması
QMgrIdentifier	Kuyruk yöneticisinin dahili olarak oluşturulan benzersiz tanıtıcısı
QMgrName	Kuyruk yöneticisi adı
QSGName	Kuyruk paylaşım grubunun adı
QueueAccounting	Kuyruklara ilişkin hesap bilgilerinin toplanmasını denetler.
QueueMonitoring	Kuyruklar için çevrimiçi izleme verileri
QueueStatistics	Kuyruklara ilişkin istatistik verilerinin toplanmasını denetler.
ReceiveTimeout	TCP/IP kanalının etkin değil durumuna dönmeden önce veri bekleyeceği süre.
ReceiveTimeoutMin	<i>ReceiveTimeout</i> niteleyicisi.
ReceiveTimeoutTip	TCP/IP kanalının etkin değil durumuna dönmeden önce veri için bekleyeceği süre alt sınırı.
RemoteEvent	Uzak hata olaylarının oluşturulup oluşturulmayacağını denetler
RepositoryName	Bu kuyruk yöneticisinin havuz hizmetleri sağladığı kümenin adı
RepositoryNamelist	Bu kuyruk yöneticisinin havuz hizmetleri sağladığı kümelerin adlarını içeren ad listesi nesnesinin adı

Çizelge 557. Kuyruk yöneticisine ilişkin öznitelikler (devamı var)	
Öznitelik	Açıklama
ScyCase	Güvenlik profilleri vakası
SharedQMGrAdı	Paylaşılan kuyruk yöneticisi adı
"SPLCAP" sayfa 822	IBM MQ Bir kuyruk yöneticisi için gelişmiş ileti güvenliği koruması açık ya da kapalı.
SSLCRLNamelist <u>1</u>	Kimlik doğrulama bilgisi nesnelerinin adlarını içeren ad listesi nesnesinin adı.
SSLCryptoHardware <u>1</u>	Şifreleme donanımı yapılandırma dizesi.
SSLEvent	TLS olayları için denetim özniteliği.
SSLFIPS Gerekli	Şifreleme için yalnızca FIPS sertifikalı algoritmaları kullanın.
SSLKeyRepository <u>1</u>	TLS anahtar havuzunun konumu.
SSLKeyRepositoryParola <u>1</u>	TLS anahtar havuzu için parola.
SSLKeyResetSayı	TLS anahtar sıfırlama sayısı.
SSLTasks <u>1</u>	TLS çağrılarını işlemek için sunucu alt görevlerinin sayısı.
StatisticsInterval	İstatistik izleme verilerinin yazma sıklığı.
StartStopOlayı	Başlatma ve durdurma olaylarının oluşturulup oluşturulmayacağını denetler
SyncPoint	Eşitleme noktası kullanılabilirliği
TCPKanalları	TCP/IP kullanan yürürlükteki kanal ya da bağlı istemci sayısı üst sınırı.
TCPKeepAlive	Bağlantının diğer ucunu denetlemek için TCP KEEPALIVE kullanılıp kullanılmayacağını belirler.
TCPAdı	Kullanmakta olduğunuz TCP/IP sisteminin adı.
TCPStackType	Kanal başlatıcının TCP/IP adreslerini nasıl kullanabileceğini belirler.
TraceRouteKayıt özniteliği	İzleme rotası bilgilerinin kaydedilmesini denetler.
TriggerInterval	Tetikleyici-ileti aralığı
Sürüm	Sürüm
XrCapability	Telemetry komutlarının desteklenip desteklenmediğini belirtir.
Notlar:	
1. Bu öznitelik MQINQ çağrısı kullanılarak sorulamaz ve bu kısımda tanımlanmaz. Bu özneneliğin ayrıntıları için Kuyruk Yöneticisinin Değiştirilmesi konusuna bakın.	

İlgili görevler

MQI istemcisinde çalıştırma zamanında yalnızca FIPS onaylı CipherSpecs kullanılmasının belirtilmesi

İlgili başvurular

AIX, Linux, and Windows için Federal Bilgi İşleme Standartları (FIPS)

AccountingConnGeçersiz Kılma (MQLONG)

Bu, uygulamaların Qmgr özniteliğindeki ACCTMQI ve ACCTQDATA değerlerini geçersiz kılmalarını sağlar.

Değer aşağıdakilerden biridir:

MQMON_DISABLED





Uygulamalar, MQCONNX çağrısındaki MQCNO yapısındaki Seçenekler alanını kullanarak ACCTMQI ve ACCTQ Qmgr özniteliklerinin ayarını geçersiz kılamaz. Bu varsayılan değerdir.

MQMON_ENABLED

Uygulamalar, MQCNO yapısındaki Seçenekler alanını kullanarak ACCTQ ve ACCTMQI Qmgr özniteliklerini geçersiz kılabilir.

Bu değerde yapılan değişiklikler yalnızca, öznitelikte yapılan değişiklikten sonra kuyruk yöneticisine yapılan bağlantılar için geçerlidir.

Bu öznitelik yalnızca aşağıdaki altyapılarda desteklenir:

-  IBM i
-   AIX and Linux
-  Windows

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_ACCOUNTING_CONN_OVERRIDE seçicisini kullanın.

AccountingInterval (MQLONG)

Bu, ara muhasebe kayıtlarının ne kadar süre (saniye cinsinden) yazılacağını belirtir.

Değer, varsayılan değeri 1800 (30 dakika) olan 0-604800 aralığında bir tamsayıdır. Ara kayıtları kapatmak için 0 değerini belirtin.

Bu öznitelik yalnızca aşağıdaki altyapılarda desteklenir:

-  IBM i
-   AIX and Linux
-  Linux
-  Windows

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_ACCOUNTING_INTERVAL seçicisini kullanın.

ActivityConnGeçersiz Kılma (MQLONG)

Bu, uygulamaların kuyruk yöneticisi özniteliğindeki ACTVTRC değerinin ayarını geçersiz kılmasını sağlar.

Değer aşağıdakilerden biridir:

MQMON_DISABLED

Uygulamalar, MQCONNX çağrısındaki MQCNO yapısındaki Seçenekler alanını kullanarak ACTVTRC kuyruk yöneticisi özniteliğinin ayarını geçersiz kılabilir. Bu varsayılan değerdir.

MQMON_ENABLED

Uygulamalar, MQCNO yapısındaki Seçenekler alanını kullanarak ACTVTRC kuyruk yöneticisi özniteliğini geçersiz kılabilir.

Bu değerde yapılan değişiklikler yalnızca, öznitelikte yapılan değişiklikten sonra kuyruk yöneticisine yapılan bağlantılar için geçerlidir.

Bu öznitelik yalnızca Çoklu platformlar üzerinde desteklenir.

Bu özniteliğin değerini belirlemek için MQINQ çağrısıyla MQIA_ACTIVITY_CONN_OVERRIDE seçicisini kullanın.

ActivityTrace (MQLONG)

Bu, IBM MQ MQI uygulama etkinliği izlemesinin derlemesini denetler.

Değer aşağıdakilerden biridir:

MQMON_ON

IBM MQ MQI uygulama etkinliği izlemesini toplayın.

MQMON_OFF

IBM MQ MQI uygulama etkinliği izlemesini toplanmayın. Bu varsayılan değerdir.

ACTVCONO kuyruk yöneticisi özniteliğini ENABLED olarak ayarlarsanız, bu değer MQCNO yapısındaki Seçenekler alanını kullanarak tek tek bağlantılar için geçersiz kılınabilir.

Bu değerde yapılan değişiklikler yalnızca, öznitelikte yapılan değişiklikten sonra kuyruk yöneticisine yapılan bağlantılar için geçerlidir.

Bu öznitelik yalnızca Çoklu platformlar üzerinde desteklenir.

Bu özneliğin değerini belirlemek için MQINQ çağrısıyla MQIA_ACTIVITY_TRACE seçicisini kullanın.

AdoptNewMCACheck (MQLONG)

Bu, zaten etkin olan bir MCA ile aynı ada sahip yeni bir gelen kanal saptandığında MCA 'nın benimsenip benimsenmeyeceğini belirlemek için denetleyecek öğeleri tanımlar.

Değer aşağıdakilerden biridir:

MQADOPT_CHECK_Q_MGR_NAME

Kuyruk yöneticisi adını denetleyin.

MQADOPT_CHECK_NET_ADDR

Ağ adresini denetleyin.


MQADOPT_CHECK_ALL

Kuyruk yöneticisi adını ve ağ adresini denetleyin. Mümkünse, kanallarınızın yanlışlıkla ya da kötü niyetle kapatılmasını önlemek için bu denetimi gerçekleştirin. Bu varsayılan değerdir.

MQADOPT_CHECK_NONE

Herhangi bir öğeyi işaretmeyin.

Bu öznelikte yapılan değişiklikler, bir kanal bir sonraki kanal benimseme girişiminde yürürlüğe girer.

 Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_ADOPTNEWMCA_CHECK seçicisini kullanın.

AdoptNewMCAType (MQLONG)

Bu, AdoptNewMCACheck özneliğiyle eşleşen yeni bir gelen kanal isteği saptandığında, belirli bir kanal tipindeki bir MCA 'nın artık kalmış bir eşgörünümünün otomatik olarak yeniden başlatılıp başlatılmayacağını belirtir.

Aşağıdaki değerlerden biridir:

MQADOPT_TYPE_NO

Artık kanal örneklerinin benimsenmesi gerekmez. Bu varsayılan değerdir.

MQADOPT_TYPE_ALL

Tüm kanal tiplerini benimseyin.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_ADOPTNEWMCA_TYPE seçicisini kullanın.

AlterationDate (MQCHAR12)

Bu, tanımın en son değiştirildiği tarihtir. Tarih biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için sondaki iki boşlukla doldurulur.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_ALTERATION_DATE seçicisini kullanın. Bu özneliğin uzunluğu MQ_DATE_LENGTH tarafından verilir.

AlterationTime (MQCHAR8)

Bu, tanımın en son değiştirildiği zamandır. Saatin biçimi şöyledir: HH.MM.SS.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_ALTERATION_TIME seçicisini kullanın. Bu özneliğin uzunluğu MQ_TIME_LENGTH tarafından verilir.

AuthorityEvent (MQLONG)

Bu, yetkilendirme (Yetkili Değil) olaylarının oluşturulup oluşturulmayacağını denetler. Aşağıdaki değerlerden biridir:

MQEVN_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVN_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_AUTHORITY_EVENT seçicisini kullanın.

BridgeEvent (MQLONG)

Bu, IMS köprü olaylarının oluşturulup oluşturulmadığını belirtir.

Değer aşağıdakilerden biridir:

MQEVN_ENABLED

IMS köprü olaylarını aşağıdaki gibi oluşturun:

MQRC_BRIDGE_BASLATILDI
MQRC_BRIDGE_DURDURULDU

MQEVN_DISABLED

IMS köprü olayları oluşturmayın; bu varsayılan değerdir.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_BRIDGE_EVENT seçicisini kullanın.

ChannelAutoDef (MQLONG)

Bu öznitelik, MQCHT_ALICI ve MQCHT_SVRCONN tipindeki kanalların otomatik tanımlamasını denetler. MQCHT_CLUSSDR kanallarının otomatik tanımlaması her zaman etkindir. Değer aşağıdakilerden biridir:

MQCHAD_DISABLED

Kanal otomatik tanımlaması geçersiz kılındı.

MQCHAD_ENABLED

Kanal otomatik tanımlaması etkinleştirildi.

 Bu öznitelik yalnızca [Çoklu platformlar](#) üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_CHANNEL_AUTO_DEF seçicisini kullanın.

ChannelAutoDefEvent (MQLONG)

Bu, kanal otomatik tanımlama olaylarının oluşturulup oluşturulmayacağını denetler. MQCHT_ALICI, MQCHT_SVRCONN ve MQCHT_CLUSSDR tipi kanallar için geçerlidir. Değer aşağıdakilerden biridir:

MQEVN_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVN_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

 Bu öznitelik yalnızca [Çoklu platformlar](#) üzerinde desteklenir.


Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_CHANNEL_AUTO_DEF_EVENT seçicisini kullanın.

ChannelAutoDefExit (MQCHARn)

Bu, otomatik kanal tanımı için kullanıcı çıkışının adıdır. Bu ad boş değilse ve *ChannelAutoDef*, MQCHAD_ENABLED değerine sahipse, kuyruk yöneticisinin kanal tanımlaması yaratmak üzere olduğu her zaman çıkış çağrılır. Bu, MQCHT_ALICI, MQCHT_SVRCONN ve MQCHT_CLUSSDR tipi kanallar için geçerlidir. Çıkış aşağıdakilerden birini yapabilir:

- Kanal tanımını deęişiklik yapmadan oluřturun.
- Oluřturulan kanal tanımının özniteliklerini deęiřtirin.
- Kanalın oluřturulmasını tamamen engelle.

Not: Bu öznitelięin hem uzunluęu hem de deęeri ortama özgüdür. Çeřitli ortamlarda bu öznitelięin deęerinin ayrıntıları için “MQCD-Kanal tanımlaması” sayfa 1454 içindeki MQCD yapısına giriş kısmına bakın.

 z/OS üzerinde bu öznitelik yalnızca küme gönderen ve küme alıcı kanalları için geçerlidir.

Bu öznitelięin deęerini saptamak için MQINQ çağrısıyla MQCA_CHANNEL_AUTO_DEF_EXIT seçicisini kullanın. Bu öznitelięin uzunluęu MQ_EXIT_NAME_LENGTH tarafından verilir.

ChannelEvent (MQLONG)

Bu, kanal olaylarının oluřturulup oluřturulmayacaęını belirtir.

Ařaęıdaki deęerlerden biridir:

MQEVR_KURAL Dıřı DURUMU

Yalnızca ařaęıdaki kanal olaylarını oluřtur:

- MQRC_CHANNEL_ACTIVATED
- MQRC_CHANNEL_CONV_HATA
- MQRC_CHANNEL_NOT_ACTIVATED
- Ařaęıdaki ReasonQualifiersile MQRC_CHANNEL_STOPPED:

MQRQ_CHANNEL_STOPPED_ERROR
MQRQ_CHANNEL_STOPPED_RETRY
MQRQ_CHANNEL_DURDURED_DISABLED

MQRC_CHANNEL_STOPD_BY_USER

MQEVR_ENABLED

Tüm kanal olaylarını oluřturun. Yani, EXCEPTION tarafından oluřturulanlara ek olarak ařaęıdaki kanal olaylarını da oluřturun:

- MQRC_CHANNEL_STARTı
- Ařaęıdaki ReasonQualifierile MQRC_CHANNEL_STOPPED:

MQRQ_CHANNEL_STOPPED_OK

MQEVR_DISABLED

Kanal olayları oluřturmayın; bu varsayılan deęerdir.

Bu öznitelięin deęerini saptamak için MQINQ çağrısıyla MQIA_CHANNEL_EVENT seçicisini kullanın.

ChannelInitiatorDenetimi (MQLONG)

Bu, kuyruk yöneticisi bařlatıldıęında kanal bařlatıcının bařlatılıp bařlatılmayacaęını belirtir.

Ařaęıdaki deęerlerden biridir:

MQSVC_CONTROL_MANUAL

Kanal bařlatıcı otomatik olarak bařlatılmayacak.

MQSVC_CONTROL_Q_MGR

Kuyruk yöneticisi bařlatıldıęında kanal bařlatıcı otomatik olarak bařlatılır.

Bu öznitelięin deęerini saptamak için MQINQ çağrısıyla MQIA_CHINIT_CONTROL seçicisini kullanın.

ChannelMonitoring (MQLONG)

Bu öznitelik, kanallar için çevrimiçi izleme verilerini belirtir.

Deęer ařaęıdakilerden biridir:

MQMON_YOK

MONCHL kanal özniteliğinin ayarından bağımsız olarak tüm kanallar için kanal izleme için veri toplamayı devre dışı bırakın. Bu varsayılan değerdir.

MQMON_OFF

MONCHL kanal özniteliğinde QMGR ' yi belirleyen kanallar için izleme verilerinin toplanmasını kapatın.

MQMON_LOW

MONCHL kanal özniteliğinde QMGR ' yi belirten kanallar için veri toplama oranının düşük olduğu bir veri toplamayla izleme verilerini toplamayı açın.

MQMON_ORTAMı

MONCHL kanal özniteliğinde QMGR ' yi belirten kanallar için veri toplama oranının orta bir oranıyla izleme verileri toplamayı açın.

MQMON_YükSEKLIK

MONCHL kanal özniteliğinde QMGR ' yi belirten kanallar için yüksek veri toplama oranıyla izleme verileri toplamayı açın.

z/OS

z/OS sistemlerinde, bu parametrenin etkinleştirilmesi, seçtiğiniz değerden bağımsız olarak istatistik verilerinin toplanmasını açar. LOW, MEDIUMya da HIGH değerinin belirlenmesi sonuçlarınız için fark yaratmaz.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_MONITORING_CHANNEL seçicisini kullanın.

ChannelStatistics (MQLONG)

Bu, kanallar için istatistik verilerinin toplanmasını denetler.

Değer aşağıdakilerden biridir:

MQMON_YOK

STATCHL kanal özniteliği ayarından bağımsız olarak, tüm kanallara ilişkin kanal istatistikleri için veri toplamayı geçersiz kılın. Bu varsayılan değerdir.

MQMON_OFF

STATCHL kanal özniteliğinde QMGR ' yi belirleyen kanallar için istatistik verileri toplamayı kapatın.

MQMON_LOW

STATCHL kanal özniteliğinde QMGR ' yi belirten kanallar için düşük veri toplama oranıyla istatistik verilerini toplama işlemi açın.

MQMON_ORTAMı

STATCHL kanal özniteliğinde QMGR ' yi belirten kanallar için veri toplama oranının orta bir oranıyla istatistik verileri toplamayı açın.

MQMON_YükSEKLIK

STATCHL kanal özniteliğinde QMGR ' yi belirten kanallar için yüksek veri toplama oranıyla istatistik verileri toplamayı açın.

Çoğu sistem için MEDIUM kullanmanız önerilir. Ancak, her saniye yüksek hacimli iletileri işleyen bir kanal için, LOW seçeneğini belirleyerek örnekleme düzeyini azaltmak isteyebilirsiniz. Ayrıca, yalnızca birkaç iletiyi işleyen ve en güncel bilgilerin önemli olduğu bir kanal için HIGH seçeneğini belirlemek isteyebilirsiniz.

z/OS

z/OS sistemlerinde, bu parametrenin etkinleştirilmesi, seçtiğiniz değerden bağımsız olarak istatistik verilerinin toplanmasını açar. LOW, MEDIUMya da HIGH değerinin belirlenmesi sonuçlarınız için fark yaratmaz. Kanal muhasebe kayıtlarını toplamak için bu parametre etkinleştirilmelidir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_STATISTICS_CHANNEL seçicisini kullanın.

ChinitAdapters (MQLONG)

Bu, IBM MQ çağrılarını işlemek için kullanılacak bağdaştırıcı alt görevlerinin sayısıdır. Değer 0-9999 arasında olmalı ve varsayılan değeri 8 olmalıdır.

Bağdaştırıcıların dağıtıcılara oranı (ChinitDispatchers özneliği) yaklaşık 8-5 olmalıdır. Ancak, yalnızca birkaç kanalınız varsa, bu parametrenin değerini varsayılan değerden düşürmeniz gerekmez. Şu değerleri kullanabilirsiniz: bir test sistemi için 8 (varsayılan); bir üretim sistemi için 20. İdeal olarak, IBM MQ çağrılarının daha fazla paralelliği sağlayan 20 bağdaştırıcıya sahip olun. Bu, kalıcı iletiler için önemlidir. Kalıcı olmayan iletiler için daha az sayıda bağdaştırıcı daha iyi olabilir.

Bu öznelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_CHINIT_ADAPTERS seçicisini kullanın.

ChinitDispatchers (MQLONG)

Bu, kanal başlatıcısı için kullanılacak dağıtıcıların sayısıdır. Değer 0-9999 arasında olmalı ve varsayılan değeri 5 olmalıdır.

Bir kılavuz olarak, 50 mevcut kanal için bir dağıtıcıya izin verin. Ancak, yalnızca birkaç kanalınız varsa, bu özneliğin değerini varsayılan değerden düşürmeniz gerekmez. TCP/IP kullanıyorsanız, burada daha büyük bir değer belirtseniz bile, TCP/IP kanalları için kullanılan dağıtıcıların en fazla sayısı 100 'dür. Şu ayarları kullanabilirsiniz: test sistemleri, 5 (varsayılan); üretim sistemleri, 20 (1000 'e kadar aktif kanalı işlemek için 20 dağıtıcıya gereksinim duyarsınız).

Bu öznelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin değerini belirlemek için MQINQ çağrısıyla MQIA_CHINIT_DISPATCHERS seçicisini kullanın.

ChinitTraceAutoStart (MQLONG)

Kanal başlatıcı izlemesinin otomatik olarak başlatılıp başlatılmayacağını belirler.

Değer aşağıdakilerden biridir:

MQTRAXSTR_YES

Kanal başlatıcı izlemesini otomatik olarak başlat. Bu varsayılan değerdir.

MQTRAXSTR_NO

Kanal başlatıcı izlemesini otomatik olarak başlatmayın.

Bu öznelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_CHINIT_TRACE_AUTO_START seçicisini kullanın.

ChinitTraceTableSize (MQLONG)

Bu, kanal başlatıcısının izleme verileri alanının boyutudur (MB).

Değer, varsayılan değeri 2 olan 0-2048 aralığında olmalıdır.

Not: Büyük z/OS veri alanlarını kullandığınızda, sisteminizde ilgili z/OS sayfalama etkinliklerini destekleyecek yeterli yardımcı bellek bulunduğundan emin olun. SYS1.DUMP veri kümeleri.

Bu öznelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_CHINIT_TRACE_TABLE_SIZE seçicisini kullanın.

ClusterSenderMonitoringDefault (MQLONG)

Bu, otomatik olarak tanımlanan küme gönderen kanallarının ChannelMonitoring özneliğinin yerine konacak değeri belirtir.

Değer aşağıdakilerden biridir:

MQMON_Q_MGR

Çevrimiçi izleme verilerinin toplanması, kuyruk yöneticisi **ChannelMonitoring** özneliğinin ayarından devralınır. Bu varsayılan değerdir.

MQMON_OFF

Kanal için izleme devre dışı bırakıldı

MQMON_LOW

ChannelMonitoring MQMON_NONE değilse, izleme düşük veri toplama hızıyla etkinleştirilir ve sistem başarımı üzerinde en az etkiye sahiptir. Toplanan veriler büyük olasılıkla en güncel veriler olmayacak.

MQMON_ORTAMI

ChannelMonitoring MQMON_NONE değilse, izleme, sistem başarımı üzerinde sınırlı etkiye sahip, orta hızda veri toplama ile etkinleştirilir.

MQMON_YÜKSEKLİK

ChannelMonitoring MQMON_NONE değilse, izleme, sistem başarımı üzerinde olası bir etkiye sahip yüksek veri toplama hızıyla etkinleştirilir. Toplanan veriler, kullanılabilir en güncel verilerdir.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_MONITORING_AUTO_CLUSSDR seçicisini kullanın.

ClusterSenderStatistics (MQLONG)

Küme gönderen kanalları havuzdaki CLUSRCVR tanımından otomatik olarak tanımlanabildiğinden, ALTER kanalını kullanarak bu otomatik olarak tanımlanan küme gönderen kanalları için STATCHL özneliğinin ayarını değiştiremezsiniz. Bu kanallar için, çevrimiçi izleme verilerinin toplanıp toplanmayacağına ilişkin karar, bu kuyruk yöneticisi özneliğinin ayarına bağlıdır.

Değer aşağıdakilerden biridir:

MQMON_Q_MGR

Otomatik olarak tanımlanan küme gönderen kanalları için istatistik verileri toplanması, kuyruk yöneticisi özneliği STATCHL değerini temel alır. Bu varsayılan değerdir.

MQMON_OFF

Otomatik tanımlı küme gönderen kanalları için istatistik verilerini toplama işlemini kapatın.

MQMON_LOW

Veri toplama oranının düşük olduğu otomatik tanımlı küme gönderen kanalları için istatistik verileri toplamayı etkinleştirin.

MQMON_ORTAMI

Veri toplamanın orta oranına sahip otomatik tanımlı küme gönderen kanalları için istatistik verileri toplamayı etkinleştirin.

MQMON_YÜKSEKLİK

Veri toplama oranının yüksek olduğu otomatik tanımlı küme gönderen kanalları için istatistik verileri toplamayı etkinleştirin.

Çoğu sistem için MEDIUM öneririz. Ancak, her saniye yüksek hacimli iletileri işleyen otomatik tanımlı bir küme gönderen kanalı için, LOW seçeneğini belirleyerek örnekleme düzeyini azaltmak isteyebilirsiniz. Ayrıca, yalnızca birkaç iletiyi işleyen ve en güncel bilgilerin önemli olduğu bir kanal için HIGH seçeneğini belirlemek isteyebilirsiniz.



z/OS sistemlerinde, bu parametrenin etkinleştirilmesi, seçtiğiniz değerden bağımsız olarak istatistik verilerinin toplanmasını açar. LOW, MEDIUMya da HIGH değerinin belirlenmesi sonuçlarınız için fark yaratmaz. Kanal muhasebe kayıtlarını toplamak için bu parametre etkinleştirilmelidir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_STATISTICS_AUTO_CLUSSDR seçicisini kullanın.

ClusterWorkloadData (MQCHAR32)

Bu, çağrıldığında küme iş yükü çıkışına geçirilen kullanıcı tanımlı 32 baytlık bir karakter dizisidir. Çıkışa geçirilemeyecek veri yoksa, dizgi boş olur.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_CLUSTER_WORKLOAD_DATA seçicisini kullanın.

ClusterWorkloadExit (MQCHARn)

Bu, küme iş yükü yönetimi için kullanıcı çıkışının adıdır. Bu ad boş değilse, bir iletinin bir küme kuyruğuna konması ya da bir küme gönderen kuyruğundan diğerine taşınması durumunda çıkış çağrılır. Çıkış, kuyruk yöneticisi tarafından seçilen kuyruk eşgörünümünü iletinin hedefi olarak kabul edebilir ya da başka bir kuyruk eşgörünümü seçebilir.

Not: Bu özneliğin hem uzunluğu hem de değeri ortama özgüdür.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_CLUSTER_WORKLOAD_EXIT seçicisini kullanın. Bu özneliğin uzunluğu MQ_EXIT_NAME_LENGTH tarafından verilir.

ClusterWorkloadUzunluğu (MQLONG)

Küme iş yükü çıkışına geçirilen ileti verileri uzunluğu üst sınırıdır. Çıkışa geçirilen verilerin gerçek uzunluğu, aşağıdakilerin en alt sınırıdır:

- İletinin uzunluğu.
- Kuyruk yöneticisinin **MaxMsgLength** özneliği.
- **ClusterWorkloadLength** özneliği.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_CLUSTER_WORKLOAD_LENGTH seçicisini kullanın.

CLWLMRUChannel (MQLONG)

Bu, küme iş yükü seçimi algoritması tarafından kullanılacak en son kullanılan küme kanalı sayısı üst sınırını belirler.

Bu, 1-999999999 aralığındaki bir değerdir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_CLWL_MRU_KANAL seçicisini kullanın.

CLWLUseQ (MQLONG)

Küme iş yükü için uzak kuyrukların kullanılıp kullanılmayacağını belirler.

Değer aşağıdakilerden biridir:

MQCLWL_USEQ_ANY

Hem yerel, hem de uzak kuyrukları kullanın.

MQCLWL_USEQ_LOCAL

Uzak kuyrukları kullanmayın. Bu varsayılan değerdir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_CLWL_USEQ seçicisini kullanın.

CodedCharSetId (MQLONG)

Bu, kuyruk yöneticisi tarafından MQI 'da tanımlanan tüm karakter dizgisi alanları için kullanılan nesne adları ve kuyruk yaratma tarihi ve saati gibi karakter kümesini tanımlar. Karakter takımı, nesne adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır. İletide taşınan uygulama verileri için geçerli değildir. Değer ortama bağlıdır:

- z/OS işletim sisteminde, kuyruk yöneticisi başlatıldığında sistem parametrelerinden değer belirlenir; varsayılan değer 500 'dür.
- Windows' da değer, kuyruk yöneticisini yaratan kullanıcının birincil CODEPAGE 'dir.
- IBM i' da değer, kuyruk yöneticisi ilk yaratıldığında ortamda ayarlanan değerdir.
- AIX and Linux' da bu değer, kuyruk yöneticisini yaratan kullanıcının ülke değerine ilişkin varsayılan CODESET değeridir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_CODED_CHAR_SET_ID seçicisini kullanın.

CommandEvent (MQLONG)

Bu, komut olaylarının aşağıdaki gibi oluşturulup oluşturulmayacağını belirtir:

MQEVR_DISABLED

Komut olayları oluşturmayın. Bu varsayılandır.

MQEVR_ENABLED

Komut olayları oluşturun.

MQEVR_NO_DISPLAY

MQINQ dışındaki tüm başarılı komutlar için komut olayları oluşturulur.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_COMMAND_EVENT seçicisini kullanın.

CommandInputQName (MQCHAR48)

Bu, yerel kuyruk yöneticisinde tanımlanan komut giriş kuyruğunun adıdır. Bu, yetki verildiyse, kullanıcıların komut gönderebileceği bir kuyruktur. Kuyruğun adı ortama bağlıdır:

- z/OS işletim sistemi üzerinde, kuyruğun adı SYSTEM.COMMAND.INPUT; MQSC ve PCF komutları gönderilebilir. MQSC komutlarının ayrıntıları için MQSC komutları ve PCF komutlarının ayrıntıları için [Programlanabilir Komut Biçimleri Tanımlamaları](#) başlıklı konuya bakın.
- Diğer tüm ortamlarda, kuyruğun adı SYSTEM.ADMIN.COMMAND.QUEUEve yalnızca PCF komutları gönderilebilir. Ancak MQSC komutu MQCMD_ESCAPE tipinde bir PCF komutuna dahil edilirse, bu kuyruğa bir MQSC komutu gönderilebilir. Escape komutuna ilişkin bilgi için bkz. [Escape](#).

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_COMMAND_INPUT_Q_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

CommandLevel (MQLONG)

Not: Sunucu ve istemciler de içinde olmak üzere tüm IBM MQ bileşenlerine ilişkin HP-UX işletim sistemi desteği IBM MQ 9.1 içinde kaldırılmıştır.

Bu, kuyruk yöneticisi tarafından desteklenen sistem denetimi komutlarının düzeyini gösterir. Bu, aşağıdaki değerlerden biri olabilir:

MQCMDL_LEVEL_800

Sistem denetim komutlarının 800. düzeyi.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 801.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 802.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem kontrol komutlarının 900. düzeyi.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzey 901.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzey 902.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzey 903.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzey 904.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzey 905.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 910.

Bu deęer ařaęıdaki sürümler tarafından döndürölür:

- 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

Sistem kontrol komutlarının seviye 911 'i.

Bu deęer ařaęıdaki sürümler tarafından döndürölür:

- 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

Sistem denetim komutlarının düzeyi 912.

Bu deęer ařaęıdaki sürümler tarafından döndürölür:

- 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

Sistem denetim komutlarının düzeyi 913.

Bu deęer ařaęıdaki sürümler tarafından döndürölür:

- 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

Sistem denetim komutlarının düzeyi 914.

Bu deęer ařaęıdaki sürümler tarafından döndürölür:

- 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

Sistem denetim komutlarının düzeyi 915.

Bu deęer ařaęıdaki sürümler tarafından döndürölür:

- 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

Sistem denetim komutlarının düzeyi 910.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 920.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 921.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 922.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 923.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 924.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 925.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 930.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 931.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

Sistem denetim komutlarının düzeyi 932.

Bu değer aşağıdaki sürümler tarafından döndürülür:

- 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

CommandLevel özniteliğinin belirli bir değerine karşılık gelen sistem denetim komutları kümesi, **Platform** özniteliğinin değerine göre değişir; her ikisi de hangi sistem denetim komutlarının desteklendiğine karar vermek için kullanılmalıdır.

Bu özniteliğin değerini belirlemek için MQINQ çağrısıyla MQIA_COMMAND_LEVEL seçicisini kullanın.

CommandServerDenetimi (MQLONG)

Kuyruk yöneticisi başlatıldığında komut sunucusunun başlatılıp başlatılmayacağını belirler.

Değer, aşağıdaki değerlerden herhangi biri olabilir:

MQSVC_CONTROL_MANUAL

Komut sunucusu otomatik olarak başlatılmayacak.

MQSVC_CONTROL_Q_MGR

Kuyruk yöneticisi başlatıldığında komut sunucusu otomatik olarak başlatılır.

Bu öznitelik z/OS üzerinde desteklenmez.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_CMD_SERVER_CONTROL seçicisini kullanın.

ConfigurationEvent (MQLONG)

Yapılandırma olaylarının oluşturulup oluşturulmayacağını denetler.

Bu özneteliğin değerini saptamak için MQINQ çağrısıyla MQIA_CONFIGURATION_EVENT seçicisini kullanın.

Değer, aşağıdaki değerlerden herhangi biri olabilir:

MQEVR_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVR_ENABLED

Olay raporlama etkinleştirildi.

Multi

CurrentQFileSize (MQLONG)

Kuyruk dosyasının megabayt cinsinden yürürlükteki büyüklüğü, en yakın megabayta yuvarlanır.

Çizelge 558. Bu özneteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu kuyruk durumu özneteliğinin değeri, kuyruğun büyüklüğü ne olursa olsun, en yakın megabayta yuvarlanır. Varsayılan öznetelikleri olan yeni bir kuyruk için **CurrentQFileSize** değeri şudur: 1.

Bu özneteliğin değer üst sınırı 99,999,9999 MB ve bu öznetelik için varsayılan değer yok.

Multi

CurrentMaxQFileSize (MQLONG)

Bir kuyrukta kullanılmakta olan blok büyüklüğü göz önüne alındığında, kuyruk dosyasının büyüyebileceği büyüklük üst sınırı en yakın megabayta yuvarlanır.

Çizelge 559. Bu özneteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu alanın kullanımı iki katlıdır:

- **MaxQFileSize** değerini geçerli blok boyutu için varsayılan değere ayarlarsanız, **CurrentMaxQFileSize** varsayılan değer denkleme olduğu gerçek değeri gösterir.
- **CurrentMaxQFileSize** **MaxQFileSize** ile eşleşmezse, daha büyük bir ayrıntı düzeyini benimsemek için kuyruğun boşaltılması gerektiğini bilirsiniz.

Not: Kuyruk dosyalarının boyutunun, blok boyutunun ve ayrıntı düzeyinin değiştirilmesine ilişkin ek bilgi için [IBM MQ kuyruk dosyalarının değiştirilmesi](#) başlıklı konuya bakın.

Bu özneteliğin değer üst sınırı: 99,999,9999 MB ve varsayılan değer yok. Değer, şu anda ayarlanan değer üst sınırıdır; varsayılan özneteliklere sahip yeni bir kuyruk için **CurrentMaxQFileSize** değeri 2.088.960 MB 'dir.

DeadLetterQName (MQCHAR48)

Bu, yerel kuyruk yöneticisinde teslim edilmeyen (teslim edilmeyen) kuyruk olarak tanımlanan bir kuyruğun adıdır. İletiler doğru hedeflerine yönlendirilemezse, bu kuyruğa gönderilir.

Örneğin, aşağıdaki durumda iletiler bu kuyruğa yerleştirilir:

- Kuyruk yöneticisine, o kuyruk yöneticisinde henüz tanımlanmamış bir kuyruğa yönlendirilmiş bir ileti gelir.
- Bir kuyruk yöneticisine bir ileti gelir, ancak gönderildiği kuyruk iletiyi alamaz; bunun nedeni aşağıdakilerden biri olabilir:

- Kuyruk dolu
- Koyma istekleri engellenmiş
- Gönderen düğümün kuyruğa ileti koyma yetkisi yok

Uygulamalar, iletileri gönderilemeyen iletiler kuyruğuna da yerleştirebilirler.

Rapor iletileri sıradan iletilerle aynı şekilde işlenir; rapor iletisi hedef kuyruğuna teslim edilemezse (genellikle özgün iletinin ileti tanımlayıcısındaki *ReplyToQ* alanı tarafından belirlenen kuyruk), rapor iletisi teslim edilmeyen iletiler (teslim edilmeyen iletiler) kuyruğuna yerleştirilir.

Not: Süre bitimine uğrayan iletiler (bkz. *MQMD-Expiry alanı*) atıldıklarında bu kuyruğa **aktarılmaz** . Ancak, gönderen uygulama tarafından istenirse, bir süre bitimi rapor iletisi (MQRO_SÜRE bitimi) oluşturulmaya devam eder ve *ReplyToQ* kuyruğuna gönderilir.

Koyma isteğini yayınlayan uygulamaya, MQPUT ya da MQPUT1 çağrısının döndürdüğü neden koduyla (örneğin, isteklerin engellendiği yerel bir kuyruğa konan bir ileti) sorunla ilgili zamanuyumlu olarak bildirim gönderildiğinde, iletiler gitmeyen iletiler (teslim edilmeyen iletiler) kuyruğuna konmaz.

Teslim edilmeyen iletiler (teslim edilmeyen iletiler) kuyruğundaki iletilerin uygulama iletisi verilerinin başında bazen bir MQDLH yapısı vardır. Bu yapı, iletinin neden teslim edilmeyen ileti kuyruğuna yerleştirildiğini gösteren ek bilgiler içerir. Bu yapının daha fazla ayrıntısı için bkz. "[MQDLH-Harf üstbilgisi](#)" sayfa 352 .

Bu kuyruk, MQUS_NORMAL **Usage** özniteliğine sahip yerel bir kuyruk olmalıdır.

Bir kuyruk yöneticisi gitmeyen (teslim edilmeyen) bir kuyruğu desteklemiyorsa ya da bir kuyruk tanımlanmamışsa, adın tümü boşluktur. Tüm IBM MQ kuyruk yöneticileri, teslim edilmeyen (teslim edilmeyen) bir kuyruğu destekler, ancak varsayılan olarak bu kuyruk tanımlanmaz.

Teslim edilmeyen ileti kuyruğu başka bir nedenle tanımlanmamışsa, dolu değilse ya da kullanılamaz durumdaysa, ileti kanalı aracısı tarafından iletilebilecek bir ileti, iletim kuyruğunda alıkonur.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_DEAD_LETTER_Q_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

DefClusterXmitQueueType (MQLONG)

DefClusterXmitQueueType özniteliği, iletileri küme alıcı kanallarına göndermek için küme gönderen kanalları tarafından varsayılan olarak hangi iletim kuyruğunun seçileceğini denetler.

DefClusterXmitQueueType değerleri MQCLXQ_SCTQ ya da MQCLXQ_CHANNEL' dir.

MQCLXQ_SCTQ

Tüm küme gönderen kanallar SYSTEM.CLUSTER.TRANSMIT.QUEUE' den ileti gönderir. İletim kuyruğuna yerleştirilen iletilerin `correlID` , iletinin gönderileceği küme gönderen kanalı belirler.

SCTQ , bir kuyruk yöneticisi tanımlandığında ayarlanır.

MQCLXQ_CHANNEL

Her küme gönderen kanal farklı bir iletim kuyruğundan ileti gönderir. Her iletim kuyruğu, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUEmodel kuyruğundan kalıcı dinamik bir kuyruk olarak yaratılır.

Kuyruk yöneticisi özniteliği DefClusterXmitQueueType, CHANNEL, varsayılan yapılandırma, tek tek küme iletim kuyruklarıyla ilişkilendirilmiş küme gönderen kanalları olarak değiştirilir. İletim kuyrukları, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUEmodel kuyruğundan yaratılan kalıcı dinamik kuyruklardır. Her iletim kuyruğu bir küme-gönderen kanalla ilişkilendirilir. Bir küme gönderen kanal bir küme iletim kuyruğuna hizmet sunarken, iletim kuyruğu tek bir kümede yalnızca bir kuyruk yöneticisine ilişkin iletiler içerir. Kümeleri, bir kümedeki her kuyruk yöneticisinin tek bir küme kuyruğu içereceği şekilde yapılandırabilirsiniz. Bu durumda, bir kuyruk yöneticisinden her küme kuyruğuna ileti trafiği iletilerden diğer kuyruklara ayrı olarak aktarılır. olarak ayarlanırsa

Değeri sorgulamak için MQINQ' i çağırın ya da bir Sorgu Kuyruğu Yöneticisi (MQCMD_INQUIRE_Q_MGR) PCF komutu gönderin, MQIA_DEF_CLUSTER_XMIT_Q_TYPE seçicisini ayarlayın. Değeri değiştirmek

için, MQIA_DEF_CLUSTER_XMIT_Q_TYPE seçicisini ayarlayarak bir Change Queue Manager (MQCMD_CHANGE_Q_MGR) PCF komutu gönderin.

İlgili başvurular

[Kuyruk Yöneticisini Değiştir](#)

[Sorma Kuyruğu Yöneticisi](#)

“MQINQ-Nesne özniteliklerini sorgu” sayfa 696

MQINQ çağrısı, bir nesnenin özniteliklerini içeren bir dizi tamsayı ve karakter dizgisi döndürür.

DefXmitQName (MQCHAR48)

Bu, hangi iletim kuyruğunun kullanılacağına ilişkin başka bir belirti yoksa, uzak kuyruk yöneticilerine iletimde kullanılan iletim kuyruğunun adıdır.

Varsayılan iletim kuyruğu yoksa, ad tamamen boştur. Bu özniteliğin ilk değeri boş.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_DEF_XMIT_Q_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

DistLists (MQLONG)

Bu, yerel kuyruk yöneticisinin MQPUT ve MQPUT1 çağrılarında dağıtım listelerini destekleyip desteklemediğini gösterir. Aşağıdaki değerlerden biridir:

MQDL_SUPPORTED

Dağıtım listeleri desteklenir.

MQDL_NOT_DESTEKLENMIYOR

Dağıtım listeleri desteklenmiyor.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_DIST_LIST seçicisini kullanın.

DNSGroup (MQCHAR18)

Bu parametre artık kullanılmıyor. Bkz. [IBM MQ 8.0 içinde ne değişti?](#)

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_DNS_GROUP seçicisini kullanın. Bu özniteliğin uzunluğu MQ_DNS_GROUP_NAME_LENGTH tarafından verilir.

DNSWLM (MQLONG)

Bu parametre artık kullanılmıyor. Bkz. [IBM MQ 8.0 içinde ne değişti?](#)

Değer aşağıdakilerden biridir:

MQDNSWLM_YES

Bu değer, daha önceki bir yayın düzeyinden geçirilen bir kuyruk yöneticisinde görülebilir. Değer yoksayıldı.

MQDNSWLM_NO

Bu, kuyruk yöneticisi tarafından desteklenen tek değerdir.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_DNS_WLM seçicisini kullanın.

ExpiryInterval (MQLONG)

Bu, kuyruk yöneticisinin süresi dolan iletileri arayan kuyrukları tarama sıklığını gösterir. Bu değer, 1-99 999 999 999 aralığındaki bir zaman aralığı ya da aşağıdaki özel değer olabilir:

MQEXPI_OFF

Kuyruk yöneticisi, süresi dolan iletileri arayan kuyrukları taramaz.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_EXPIRY_INTERVAL seçicisini kullanın.

z/OS

Bu öznitelik yalnızca z/OSüzerinde desteklenir.

IGQPutAuthority (MQLONG)

Bu öznitelik, yerel kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiye geçerlidir. Yerel grup içi kuyruk aracısı (IGQ aracısı) paylaşılan iletim kuyruğundan bir iletiyi kaldırdığında ve iletiyi yerel bir kuyruğa yerleştirdiğinde gerçekleştirilen yetki denetimi tipini belirtir. Değer aşağıdakilerden biridir:

MQIGQPA_DEFAULT

Yetki için denetlenen kullanıcı kimliği, ileti paylaşılan iletim kuyruğunda olduğunda iletiyle ilişkili ayrı MQMD 'deki *UserIdentifier* alanının değeridir. Bu, iletiyi paylaşılan iletim kuyruğuna yerleştiren programın kullanıcı kimliğidir ve genellikle uzak kuyruk yöneticisinin altında çalıştığı kullanıcı tanıtıcısıyla aynıdır.

RESLEVEL profili birden çok kullanıcı kimliğinin denetleneceğini belirtiyorsa, yerel IGQ aracısının (*IGQUserId*) kullanıcı kimliği de denetlenir.

MQIGQPA_CONTEXT

Yetki için denetlenen kullanıcı kimliği, ileti paylaşılan iletim kuyruğunda olduğunda iletiyle ilişkili ayrı MQMD 'deki *UserIdentifier* alanının değeridir. Bu, iletiyi paylaşılan iletim kuyruğuna yerleştiren programın kullanıcı kimliğidir ve genellikle uzak kuyruk yöneticisinin altında çalıştığı kullanıcı tanıtıcısıyla aynıdır.

RESLEVEL tanıtımı, birden çok kullanıcı kimliğinin denetleneceğini belirtiyorsa, yerel IGQ aracısının (*IGQUserId*) kullanıcı kimliği ve *yerleşik* MQMD 'deki *UserIdentifier* alanının değeri de denetlenir. İkinci kullanıcı kimliği genellikle iletiyi oluşturan uygulamanın kullanıcı kimliğidir.

MQIGQPA_ONLY_IGQ

Yetki için denetlenen kullanıcı kimliği, yerel IGQ aracısının (*IGQUserId*) kullanıcı kimliğidir.

RESLEVEL tanıtımı birden çok kullanıcı kimliğinin denetleneceğini belirtiyorsa, tüm denetimler için bu kullanıcı kimliği kullanılır.

MQIGQPA_ALTERNATE_OR_IGQ

Yetki için denetlenen kullanıcı kimliği, yerel IGQ aracısının (*IGQUserId*) kullanıcı kimliğidir.

RESLEVEL tanıtımı birden çok kullanıcı kimliğinin denetleneceğini belirtiyorsa, *yerleşik* MQMD 'deki *UserIdentifier* alanının değeri de denetlenir. Bu kullanıcı kimliği genellikle iletiyi oluşturan uygulamanın kullanıcı kimliğidir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_IGQ_PUT_AUTHORITY seçicisini kullanın.

z/OS

Bu öznitelik yalnızca z/OSüzerinde desteklenir.

IGQUserId (MQLONG)

Bu öznitelik, yerel kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiye geçerlidir. Yerel grup içi kuyruğa alma aracısıyla (IGQ aracısı) ilişkili kullanıcı kimliğini belirtir. Bu tanıtıcı, IGQ aracısı iletileri yerel kuyruklara koyduğunda yetki denetimi yapılabilecek kullanıcı tanıtıcılarından biridir. Denetlenen gerçek kullanıcı tanıtıcıları, **IGQPutAuthority** özneliğinin ayarına ve dış güvenlik seçeneklerine bağlıdır.

IGQUserId boşsa, IGQ aracısıyla ilişkilendirilmiş bir kullanıcı kimliği yoktur ve ilgili yetki denetimi gerçekleştirilmez (diğer kullanıcı tanıtıcıları yetki için denetlenmeye devam edebilir).

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_IGQ_USER_ID seçicisini kullanın. Bu özneliğin uzunluğu MQ_USER_ID_LENGTH tarafından verilir.

z/OS

Bu öznitelik yalnızca z/OSüzerinde desteklenir.

InhibitEvent (MQLONG)

Bu, engelleme (Yerleştirmeyi Engelleme ve Engelleme) olaylarının oluşturulup oluşturulmayacağını denetler. Değer aşağıdakilerden biridir:

MQEVN_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVN_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_INHIBIT_EVENT seçicisini kullanın.

z/OS üzerinde, bu özniteliğin değerini saptamak için MQINQ çağrısı kullanamazsınız.

IntraGroupqueuing (MQLONG)

Bu öznitelik, yerel kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiye geçerlidir. Kuyruk paylaşım grubu için grup içi kuyruğa alma özelliğinin etkinleştirilip etkinleştirilmediğini gösterir. Değer aşağıdakilerden biridir:

MQIGQ_DISABLED

Kuyruk paylaşım grubundaki diğer kuyruk yöneticilerine gönderilen tüm iletiler geleneksel kanallar kullanılarak iletilir.

MQIGQ_ENABLED

Aşağıdaki koşul yerine getirilirse, kuyruk paylaşım grubundaki diğer kuyruk yöneticilerine gönderilen iletiler paylaşılan iletim kuyruğu kullanılarak iletilir:


- İleti verilerinin ve iletim üstbilgisinin uzunluğu 63 KB ' yi (64 512 bayt) geçmez.

İletim üstbilgisi için MQXQH büyüklüğünden biraz daha fazla yer ayrılması önerilir; bu amaçla sabit MQ_MSG_HEADER_LENGTH sağlanır.

Bu durum karşılanmazsa, ileti geleneksel kanallar kullanılarak iletilir.

Not: Grup içi kuyruğa alma etkinleştirildiğinde, paylaşılan iletim kuyruğu kullanılarak iletilen iletilerin sırası, geleneksel kanallar kullanılarak iletilenlere göre korunmaz.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_INTRA_GROUP_queuing seçicisini kullanın.

 Bu öznitelik yalnızca z/OS üzerinde desteklenir.

IPAddressVersion (MQLONG)

Hangi IP adresi sürümünün (IPv4 ya da IPv6) kullanılacağını belirler.

Bu öznitelik yalnızca IPv4 ve IPv6 çalıştıran sistemler için geçerlidir ve aşağıdaki koşullardan biri geçerli olduğunda yalnızca *TransportType* MQXPY_TCP ' ye sahip olarak tanımlanan kanalları etkiler:

- Kanalın *ConnectionName* , hem IPv4 hem de IPv6 adresine çözülen bir anasistem adıdır ve **LocalAddress** parametresi belirtilmez.
- Kanalın *ConnectionName* ve *LocalAddress* , hem IPv4 hem de IPv6 adreslerine çözümleyen anasistem adlarıdır.

Değer, aşağıdaki değerlerden herhangi biri olabilir:

MQIPADDR_IPV4

IPv4 kullanılır.

MQIPADDR_IPV6

IPv6 kullanılır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_IP_ADDRESS_VERSION seçicisini kullanın.

ListenerTimer (MQLONG)

Bu, bir APPC ya da TCP/IP hatası ortaya çıkarsa, IBM MQ ' in dinleyiciyi yeniden başlatma girişimi arasındaki zaman aralığıdır (saniye cinsinden). Değer, varsayılan değeri 60 olan 5 ile 9999 arasında olmalıdır.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin deęerini saptamak için MQINQ çağrısıyla MQIA_LISTENER_TIMER seçicisini kullanın.

LocalEvent (MQLONG)

Bu, yerel hata olaylarının oluşturulup oluşturulmayacağını denetler. Deęer aşağıdakilerden biridir:

MQEVN_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVN_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin deęerini saptamak için MQINQ çağrısıyla MQIA_LOCAL_EVENT seçicisini kullanın.

z/OS üzerinde, bu özneliğin deęerini saptamak için MQINQ çağrısı kullanamazsınız.

LoggerEvent (MQLONG)

Bu, kurtarma günlüğü olaylarının oluşturulup oluşturulmayacağını denetler. Deęer aşağıdakilerden biridir:

MQEVN_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVN_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin deęerini saptamak için, MQINQ çağrısıyla MQIA_LOGGER_EVENT seçicisini kullanın.

 Bu öznitelik yalnızca [Çoklu platformlar](#) üzerinde desteklenir.

LUGroupName (MQCHAR8)

Bu, kuyruk paylaşım grubuna ilişkin gelen iletimleri işleyen LU 6.2 dinleyicisinin soysal LU adıdır. Bu adı boş bırakırsanız, bu dinleyiciyi kullanamazsınız.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin deęerini saptamak için MQINQ çağrısıyla MQCA_LU_GROUP_NAME seçicisini kullanın. Bu özneliğin uzunluğu MQ_LU_NAME_LENGTH tarafından verilir.

LUName (MQCHAR8)

Bu, giden LU 6.2 iletimleri için kullanılacak LU ' nun adıdır. Bunu, dinleyicinin gelen iletimler için kullandığı LU ' ya ayarlayın. Bu adı boş bırakırsanız, APPC/MVS varsayılan LU kullanılır; bu deęişken olduğundan, LU6.2 kullanıyorsanız her zaman LUName deęerini ayarlayın.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin deęerini saptamak için MQINQ çağrısıyla MQCA_LU_NAME seçicisini kullanın. Bu özneliğin uzunluğu MQ_LU_NAME_LENGTH tarafından verilir.

LU62ARMSuffix (MQCHAR2)

Bu, SYS1.PARMLIB üyesi APPCPMxx. z/OS komutu SET APPC=xx, ARM kanal başlatıcısını yeniden başlattığında yayınlanır. Bu adı boş bırakırsanız, SET APPC=xx yayınlanmaz.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin deęerini saptamak için, MQINQ çağrısıyla MQCA_LU62_ARM_SUFFIX seçicisini kullanın. Bu özneliğin uzunluğu MQ_ARM_SUFFIX_LENGTH tarafından verilir.

LU62Channels (MQLONG)

Bu, LU 6.2 iletim protokolünü kullanan, yürürlükteki kanal ya da bağlanabilen istemci sayısı üst sınırıdır.

Değer, varsayılan değeri 200 olan 0-9999 aralığında olmalıdır. Bu değeri sıfıra ayarlarsanız, LU 6.2 iletim protokolü kullanılmaz.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_LU62_CHANNELS seçicisini kullanın.

MaxActiveKanalları (MQLONG)

Bu öznitelik, herhangi bir zamanda *etkin* olabilecek kanal sayısı üst sınırıdır.

Varsayılan değer, MaxChannels özniteliği için belirtilen değerdir.

z/OS için değer, 1-9 999 aralığında olmalıdır.

Diğer tüm platformlar için varsayılan değer 999 999 999 999 'dur. Bu, etkin kanal sayısının sınırsız olduğu anlamına gelir ya da sınır koymak için gerçek bir sayıya ayarlanabilir.

MQ Appliance IBM MQ Appliance üzerindeki **MaxActiveChannels** değerini değiştirmemelisiniz. İstemci kanalı sayısı üst sınırını sınırlamak istiyorsanız, her bir SVRCONN kanalı için sınırlar tanımlamak üzere SVRCONN kanal tanımlarındaki kanal başına MAXINST ve MAXINSTC özniteliklerini kullanın, IBM MQ Appliance belgelerinde [IBM MQ Appliance kuyruk yöneticisi yapılandırmasına](#) bakın.

MaxActiveChannels parametresi yalnızca z/OS üzerinde bir kuyruk yöneticisi özniteliğidir. Diğer platformlarda **MaxActiveChannels**, qm.ini dosyasındaki bir özniteliktir. **MaxActiveChannels** özniteliğini diğer altyapılarda nasıl ayarlayacağınıza ilişkin bilgi için [Dağıtılmış kuyruğa alma için yapılanış kütüğü stanzları](#) konusuna bakın.

Bu özniteliğin değerini saptamak için, **MQINQ** çağrısıyla MQIA_ACTIVE_KANAL seçicisini kullanın.

İlgili kavramlar

[Kanal durumları](#)

MaxChannels (MQLONG)

Bu öznitelik, *yürürlükteki* olabilecek kanal sayısı üst sınırıdır (bağlı istemcilere sahip sunucu bağlantısı kanalları da içinde olmak üzere).

z/OS için değer, varsayılan değeri 200 olan 1-9 999 aralığında olmalıdır.

MQ Appliance IBM MQ Appliance için, varsayılan değer 999 999 999 999 'dur ve değiştirilmemelidir. İstemci kanalı sayısı üst sınırını sınırlamak istiyorsanız, her bir SVRCONN kanalı için sınırlar tanımlamak üzere SVRCONN kanal tanımlarındaki kanal başına MAXINST ve MAXINSTC özniteliklerini kullanın, IBM MQ Appliance belgelerinde [IBM MQ Appliance kuyruk yöneticisi yapılandırmasına](#) bakın.

Ağdaki bağlantılara hizmet vermekle meşgul olan bir sistem için varsayılan ayardan daha yüksek bir sayı gerekebilir. Test sırasında sisteminizin davranışını gözlemleyerek, ortamınız için doğru olan değeri belirleyin.

Diğer tüm platformlar için varsayılan değer 100 'dür. Gerekiyorsa, geçerli kanal sayısı üst sınırını sınırlamak için **MaxChannels** değerini farklı bir değere ayarlayabilirsiniz.

MaxChannels parametresi yalnızca z/OS üzerinde bir kuyruk yöneticisi özniteliğidir. Diğer platformlarda **MaxChannels**, qm.ini dosyasındaki bir özniteliktir. **MaxChannels** özniteliğini diğer altyapılarda nasıl ayarlayacağınıza ilişkin bilgi için [Dağıtılmış kuyruğa alma için yapılanış kütüğü stanzları](#) konusuna bakın.

Bu özniteliğin değerini saptamak için, **MQINQ** çağrısıyla MQIA_MAX_KANALLI seçiciyi kullanın.

İlgili kavramlar

[Kanal durumları](#)

MaxHandles (MQLONG)

Bu, herhangi bir görevin eşzamanlı olarak kullanabileceği açık tanıtıcı sayısı üst sınırıdır. Tek bir kuyruk için (ya da kuyruk olmayan bir nesne için) her başarılı MQOPEN çağrısı tek bir tanıtıcı kullanır. Bu tanıtıcı, nesne kapatıldığında yeniden kullanılabilir duruma gelir. Ancak, bir dağıtım listesi açıldığında, dağıtım listesindeki her kuyruğa ayrı bir tanıtıcı ayrılır; böylece, MQOPEN çağrısı dağıtım listesindeki kuyruklar kadar tanıtıcı kullanır. Bu, *MaxHandles* için uygun bir değere karar verilirken dikkate alınmalıdır.

MQPUT1 çağrısı, işleminin bir parçası olarak bir MQOPEN çağrısı gerçekleştirir; sonuç olarak, MQPUT1 , MQOPEN ' in kullanacağı sayıda tanıtıcı kullanır, ancak bu tanıtıcıları yalnızca MQPUT1 çağrısının kendisini çağırıldığı süre boyunca kullanır.

z/OS sistemlerinde görev , CICS görevi, MVS görevi ya da IMS bağımlı bölgesi anlamına gelir.

Değer, 1-999 999 999 999 aralığındadır. Varsayılan değer ortam tarafından belirlenir:

- z/OS için varsayılan değer 100 'dür.
- Diğer tüm ortamlarda varsayılan değer 256 'dir.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_MAX_TANITICI seçiciyi kullanın.

MaxMsgUzunluğu (MQLONG)

Bu, kuyruk yöneticisinin işleyebileceği en uzun *fiziksel* iletinin uzunluğudur. Ancak, **MaxMsgLength** kuyruk yöneticisi özneliği **MaxMsgLength** kuyruk özneliğinden bağımsız olarak ayarlanabildiğinden, bir kuyruğa yerleştirilebilecek en uzun fiziksel ileti bu iki değerden daha azdır.

Kuyruk yöneticisi bölümlenmeyi destekliyorsa, bir uygulama iki **MaxMsgLength** özneliklerinden daha uzun bir mantıksal ileti yerleştirebilir, ancak uygulama MQMD ' de MQMF_SEGMENTATION_ALLOWED işaretini belirtiyorsa. Bu işaret belirtilirse, mantıksal ileti uzunluğu üst sınırı 999 999 999 bayttır, ancak genellikle işletim sistemi ya da uygulamanın çalıştığı ortam tarafından zorunlu kılınan kaynak kısıtlamaları alt sınırla sonuçlanır.

MaxMsgLength özneliği için alt sınır 32 KB 'dir (32 768 bayt). Üst sınır 100 MB (104 857 600 bayt).

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_MAX_MSG_LENGTH seçicisini kullanın.

MaxPriority (MQLONG)

Bu, kuyruk yöneticisi tarafından desteklenen ileti önceliği üst sınırıdır. Öncelikler sıfır (en düşük) ile *MaxPriority* (en yüksek) arasında değişir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_MAX_PRIORITY seçicisini kullanın.

MaxPropertiesLength (MQLONG)

Bu, bir iletiyle akabilecek özelliklerin boyutunu denetlemek için kullanılır. Bu, hem özellik adını bayt cinsinden, hem de özellik değerinin boyutunu bayt cinsinden içerir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_MAX_PROPERTIES_LENGTH seçicisini kullanın.

Multi

MaxQFileBüyüküğü (MQLONG)

Bir kuyruk dosyasının büyüyebileceği megabayt cinsinden büyüklük üst sınırı.

Çizelge 560. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bir kuyruk dosyası, yürürlükteki kuyruk dosyası boyutundan daha düşük bir değere yapılandırılırsa, büyüklük üst sınırını aşabilir. Bu durumda, kuyruk dosyası artık yeni iletileri kabul etmez, ancak var olan iletilerin tüketilmesine izin verir. Kuyruk dosyası büyüklüğü yapılandırılan değer altına düştüğünde, yeni iletilerin kuyruğa konmasına izin verilir.

Not: Bu şekil, kuyruk yöneticisinin seçilen boyuta ulaşmak için daha büyük bir blok boyutu kullanması gerekebileceğinden, kuyrukta yapılandırılan özniteliğin değerinden farklı olabilir. Kuyruk dosyalarının boyutunun, blok boyutunun ve ayrıntı düzeyinin değiştirilmesine ilişkin ek bilgi için [IBM MQ kuyruk dosyalarının değiştirilmesi](#) başlıklı konuya bakın.

Bu öznitelik artırıldığından ayrıntı düzeyinin değiştirilmesi gerektiğinde, AMQERR günlüklerine AMQ7493W öge boyu değışti uyarı iletisi yazılır. Bu, IBM MQ ' un yeni ayrıntı düzeyini benimsemesi için kuyruğun boşaltılması için plan yapmanız gerektiğini gösterir.

Bu özniteliğin üst sınır değeri 267.386.880 MB ve varsayılan değeri ve geçirilen değeri 2.088.960 MB olup bu, 512 'lik bir ayrıntı düzeyine sahip bir kuyruk için geçerli üst sınır değeri.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_MAX_Q_FILE_SIZE seçicisini kullanın.

MaxUncommittedİletiler (MQLONG)

Bu, bir iş biriminde var olabilecek kesinleştirilmemiş ileti sayısı üst sınırıdır. Kesinleştirilmemiş iletilerin sayısı, yürürlükteki iş biriminin başlangıcından bu yana aşağıdakilerin toplamıdır:

- MQPMO_SYNCPOINT seçeneğiyle uygulama tarafından konan iletiler
- MQGMO_SYNCPOINT seçeneğiyle uygulama tarafından alınan iletiler
- MQPMO_SYNCPOINT seçeneğiyle konan iletiler için kuyruk yöneticisi tarafından oluşturulan tetikleyici iletiler ve COA rapor iletileri
- MQGMO_SYNCPOINT seçeneğiyle alınan iletiler için kuyruk yöneticisi tarafından oluşturulan COD rapor iletileri

Aşağıdaki iletiler kesinleştirilmemiş olarak sayılmaz:

- Uygulama tarafından bir iş biriminin dışına konan ya da alınan iletiler
- Bir iş biriminin dışına konan ya da alınan iletilerin sonucu olarak kuyruk yöneticisi tarafından oluşturulan tetikleyici iletiler ya da COA/COD rapor iletileri
- Kuyruk yöneticisi tarafından oluşturulan süre bitim raporu iletileri (süre bitimi rapor iletisine neden olan çağrı MQGMO_SYNCPOINT belirtse bile)
- Kuyruk yöneticisi tarafından oluşturulan olay iletileri (olay iletisine neden olan çağrı MQPMO_SYNCPOINT ya da MQGMO_SYNCPOINT belirtse bile)

Not:

1. Kural dışı durum raporu iletileri, Message Channel Agent (MCA) ya da uygulama tarafından oluşturulur ve uygulama tarafından konan ya da alınan olağan iletilerle aynı şekilde işlenir.
2. Bir ileti ya da bölüm MQPMO_SYNCPOINT seçeneğiyle birlikte konduğunda, kesinleştirilmemiş iletilerin sayısı, koymadan elde edilen fiziksel iletilerin sayısından bağımsız olarak, bir artırılır. (Kuyruk yöneticisinin iletiyi ya da kesimi alt bölümlere ayırması gerekiyorsa, birden çok fiziksel ileti oluşabilir.)
3. MQPMO_SYNCPOINT seçeneğiyle bir dağıtım listesi konduğunda, kesinleştirilmemiş iletilerin sayısı bir *oluşturulan her fiziksel ileti için* artırılır. Bu, bir kişi kadar küçük ya da dağıtım listesindeki hedef sayısı kadar büyük olabilir.

Bu özniteliğin alt sınırı 1; üst sınır 999 999 999 'dur. Varsayılan değeri 10000 'dir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_MAX_UNCOMMITTED_MSGS seçicisini kullanın.

MQIAccounting (MQLONG)

Bu, MQI verileri için muhasebe bilgileri toplamasını denetler.

Değeri aşağıdakilerden biridir:

MQMON_ON

API muhasebe verilerini toplayın.

MQMON_OFF

API muhasebe verilerini toplanmayın. Bu varsayılan değerdir.

ACCTCONO kuyruk yöneticisi özniteliğini ENABLED olarak ayarlarsanız, bu değer MQCNO yapısındaki Seçenekler alanını kullanan tek tek bağlantılar için geçersiz kılınabilir. Bu değerde yapılan değişiklikler yalnızca, öznitelikte yapılan değişiklikten sonra kuyruk yöneticisine yapılan bağlantılar için geçerlidir.

Bu öznitelik yalnızca Çoklu platformlar üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_ACCOUNTING_MQI seçicisini kullanın.

MQIStatistics (MQLONG)

Bu, kuyruk yöneticisine ilişkin istatistik izleme bilgilerinin toplanmasını denetler.

Değer aşağıdakilerden biridir:

MQMON_ON

MQI istatistiklerini toplayın.

MQMON_OFF

MQI istatistiklerini toplamayın. Bu varsayılan değerdir.

Bu öznitelik yalnızca Çoklu platformlar üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_STATISTICS_MQI seçicisini kullanın.

MsgMarkBrowseInterval (MQLONG)

Kuyruk yöneticisinin göz atma iletilerinden işareti otomatik olarak kaldırabileceği zaman aralığı (milisaniye).

Bu, kuyruk yöneticisinin göz atma iletilerinden işareti otomatik olarak kaldırabileceği bir zaman aralığıdır (milisaniye cinsinden).

Bu öznitelik, MQGMO_MARK_BROWSE_CO_OP ileti alma seçeneği kullanılarak MQGET çağrısı tarafından göz atma olarak işaretlenmiş iletilerin göz atma olarak kalacağı zaman aralığını açıklar.

Kuyruk yöneticisi, bu yaklaşık aralıktan daha uzun bir süre için işaretlendiklerinde, işbirliği içinde çalışan tanıtıcı kümesi için göz atma olarak işaretlenmiş olarak işaretlenmiş iletilerin işaretini otomatik olarak geri çekebilir.

Bu, MQGMO_MARK_BROWSE_HANDLE ileti alma seçeneği kullanılarak MQGET çağrısıyla elde edilen, göz atma olarak işaretlenmiş herhangi bir iletinin durumunu etkilemez.

Üst sınır 999 999 999 ve varsayılan değer 5000 'dir. *MsgMarkBrowseInterval* için -1 özel değeri, sınırsız bir zaman aralığını temsil eder.



Uyarı: Bu değer, varsayılan değer olan 5000 'in altında olmamalıdır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_MSG_MARK_BROWSE_INTERVAL seçicisini kullanın.

OutboundPortÜst Sınırı (MQLONG)

Bu, giden kanalları bağlamak için kullanılacak kapı numaralarının OutboundPortMin ve OutboundPortMax tarafından tanımlanan aralıktaki en yüksek kapı numarasıdır.

Değer, 0-65535 aralığındaki bir tamsayıdır ve OutboundPortMin değerine eşit ya da bu değerden büyük olmalıdır. Varsayılan değer 0'dır.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_OUTBOUND_PORT_MAX seçicisini kullanın.

OutboundPortAlt Sınırı (MQLONG)

Bu, giden kanalları bağlamak için kullanılacak kapı numaralarının OutboundPortMin ve OutboundPortMax tarafından tanımlanan aralıktaki en düşük kapı numarasıdır.

Değer, 0-65535 aralığındaki bir tamsayıdır ve OutboundPortüst sınır değerine eşit ya da bu değerden küçük olmalıdır. Varsayılan değer 0'dır.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin deęerini saptamak için MQINQ çağrısıyla MQIA_OUTBOUND_PORT_MIN seçicisini kullanın.

PerformanceEvent (MQLONG)

Bu, performansla ilgili olayların oluşturulup oluşturulmayacağını denetler. Aşağıdaki deęerlerden biridir:

MQEVR_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVR_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin deęerini saptamak için MQINQ çağrısıyla MQIA_PERFORMANCE_EVENT seçicisini kullanın.

Platform (MQLONG)

Bu, kuyruk yöneticisinin çalıştığı işletim sistemini gösterir:

MQPL_AIX

AIX (MQPL_UNIX ile aynı deęer).

MQPL_APPLIANCE

IBM MQ Appliance

MQPL_MVS

z/OS (MQPL_ZOS ile aynı deęer).

MQPL_OS390

z/OS (MQPL_ZOS ile aynı deęer).

MQPL_OS400

IBM i.

MQPL_UNIX

UNIX.

MQPL_WINDOWS_NT

Windows sistemleri.

MQPL_ZOS

z/OS.

Bu özneliğin deęerini saptamak için MQINQ çağrısıyla MQIA_PLATFORM seçicisini kullanın.

PubSubNPInputMsg (MQLONG)

Teslim edilmemiş bir giriş iletilinin atılıp atılmayacağı ya da alınıp tutulmayacağı.

Deęer aşağıdakilerden biridir:

MQUNDELIVERED_DISCARD

Kalıcı olmayan giriş iletileri işlenemezse atılabilir.

Bu varsayılan deęerdir.

MQUNDELIVERED_TUT

Kalıcı olmayan giriş iletileri işlenemezse atılamaz. Bu durumda, kuyruğa alınan yayınlama/abone olma arabirimi işlemi uygun aralıklarla yeniden denemeye devam eder ve sonraki iletileri işlemeye devam etmez.

Bu özneliğin deęerini saptamak için MQINQ çağrısıyla MQIA_PUBSUB_NP_MSG seçicisini kullanın.

PubSubNPResponse (MQLONG)

Teslim edilmemiş yanıt iletilerinin davranışını denetler.

Deęer aşağıdakilerden biridir:

MQUNDELIVERED_NORMAL

Yanıt kuyruğuna yerleştirilemeyen kalıcı olmayan yanıtlar, DLQ ' ya yerleştirilemezlerse, atılır.

MQUNDELIVERED_SAFE

Yanıt kuyruğuna yerleştirilemeyen kalıcı olmayan yanıtlar, teslim edilmeyen iletiler kuyruğuna konur. Yanıt ayarlanamazsa ve DLQ ' ya yerleştirilemezse, kuyruğa alınan yayınlama/abone olma arabirimi yürürlükteki işlemi geriye işleyip uygun aralıklarla yeniden dener ve sonraki iletileri işlemeye devam etmez.

MQUNDELIVERED_DISCARD

Kalıcı olmayan yanıtlar yanıt kuyruğuna yerleştirilmez.

Bu, yeni kuyruk yöneticileri için varsayılan değerdir.

MQUNDELIVERED_TUT

Kalıcı olmayan yanıtlar, gitmeyen iletiler kuyruğuna yerleştirilmez ya da atılmaz. Bunun yerine, kuyruğa alınan yayınlama/abone olma arabirimi yürürlükteki işlemi geri alır ve uygun aralıklarla yeniden dener.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_PUBSUB_NP_RESP seçicisini kullanın.

Geçirilen kuyruk yöneticileri için varsayılan değer.

Kuyruk yöneticisi IBM MQ V6.0' dan geçirildiyse, bu özniteliğin ilk değeri, aşağıdaki çizelgede gösterildiği gibi, geçişten önce *DiscardNonPersistentResponse* ve *DLQNonPersistentResponse* değerlerine bağlıdır.

		DLQNonPersistentYanıt		
		Evet	Hayır	Ayarlanmadı
DiscardNonPersistentResponse	Evet	MQUNDELIVERED_NORMAL	MQUNDELIVERED_DISCARD	MQUNDELIVERED_NORMAL
	Hayır	MQUNDELIVERED_SAFE	MQUNDELIVERED_TUT	MQUNDELIVERED_SAFE
	Ayarlanmadı	SyncPointPersistent = No, MQUNDELIVERED_SAFE ise, MQUNDELIVERED_NORMAL	SyncPointPersistent = No, MQUNDELIVERED_SÜRDÜRME ise, başka bir MQUNDELIVERED_DISCARD	SyncPointPersistent = No, MQUNDELIVERED_SAFE ise, MQUNDELIVERED_NORMAL

PubSubMaxMsgRetryCount (MQLONG)

Eşitleme noktası altındaki başarısız bir komut iletileri işlenirken yeniden deneme sayısı.

Değer aşağıdakilerden biridir:

0-999 999 999

Varsayılan değer 5'tir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_PUBSUB_MAXMSG_RETRY_COUNT seçicisini kullanın.

PubSubSyncPoint (MQLONG)

Yalnızca kalıcı iletilerin mi, yoksa tüm iletilerin mi eşitleme noktası altında işlendiğini belirler.

Değer aşağıdakilerden biridir:

MQSYNCPOINT_IFPER

Bu, kuyruğa alınan yayınlama/abone olma arabiriminin, eşitleme noktası dışında kalıcı olmayan iletiler almasını sağlar. Yardımcı program syncpoint dışında bir yayın alırsa, yardımcı program yayını syncpoint dışında bilinen abonelere iletir.

Bu varsayılan değerdir.

MQSYNCPOINT_YES

Bu, kuyruğa alınan yayınlama/abone olma arabiriminin, eşitleme noktası altındaki tüm iletileri almasını sağlar.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_PUBSUB_SYNC_PT seçicisini kullanın.

PubSubKipi (MQLONG)

Yayınlama/abone olma motorunun ve kuyruğa alınan yayınlama/abone olma arabiriminin çalışıp çalışmadığını, bu nedenle uygulamaların uygulama programlama arabirimini ve kuyruğa alınan yayınlama/

abone olma arabirimi tarafından izlenmekte olan kuyrukları kullanarak yayınlama/abone olma işlemlerini gerçekleştirmesine izin verir.

Değer aşağıdakilerden biridir:

MQPSM_COMPAT

Yayınlama/abone olma altyapısı çalışıyor. Bu nedenle, uygulama programlama arabirimini kullanarak yayınlama/abone olma mümkündür. Kuyruğa alınan yayınlama/abone olma arabirimi çalışmıyor; bu nedenle, kuyruğa alınan yayınlama/abone olma arabirimi tarafından izlenen kuyruklara konan hiçbir ileti üzerinde işlem yürütülmez. Bu ayar, kuyruğa alınan yayınlama/abone olma arabiriminin olağan olarak okuduğu kuyrukları okuması gerektiğinden, bu kuyruk yöneticisini kullanan WebSphere Message Broker V6 ya da önceki sürümlerle uyumluluk için kullanılır.

MQPSM_DISABLED

Yayınlama/abone olma altyapısı ve kuyruğa alınan yayınlama/abone olma arabirimi çalışmıyor. Bu nedenle, uygulama programlama arabirimini kullanarak yayınlama/abone olma yapılamaz. Kuyruğa alınan yayınlama/abone olma arabirimi tarafından izlenen kuyruklara konan yayınlama/abone olma iletileri üzerinde işlem uygulanmaz.

MQPSM_ENABLED

Yayınlama/abone olma altyapısı ve kuyruğa alınan yayınlama/abone olma arabirimi çalışıyor. Bu nedenle, kuyruğa alınan yayınlama/abone olma arabirimi tarafından izlenmekte olan kuyruklar ve uygulama programlama arabirimi kullanılarak yayınlama/abone olma mümkündür. Bu, kuyruk yöneticisinin ilk varsayılan değeridir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_PUBSUB_MODE seçicisini kullanın.

QMGrDesc (MQCHAR64)

Kuyruk yöneticisini açıklayan bir açıklama için bu alanı kullanın. Alanın içeriği kuyruk yöneticisi için önemli değil, ancak kuyruk yöneticisi alanın yalnızca görüntülenebilecek karakterleri içermesini gerektirebilir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, bu alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 byte olabilir).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özneliğinde tanımlandığı gibi) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

- z/OS' da varsayılan değer, ürün adı ve sürüm numarasıdır.
- Diğer tüm ortamlarda varsayılan değer boşluktur.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_Q_MGR_DESC seçicisini kullanın. Bu özneliğin uzunluğu MQ_Q_MGR_DESC_LENGTH tarafından verilir.

QMGrIdentifier (MQCHAR48)

Bu, kuyruk yöneticisi için dahili olarak oluşturulan benzersiz bir addır.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_Q_MGR_IDENTIFIER seçicisini kullanın. Bu özneliğin uzunluğu MQ_Q_MGR_IDENTIFIER_LENGTH tarafından verilir.

Bu öznelik aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

ve bu sistemlere bağlı IBM MQ istemcileri.

QMgrName (MQCHAR48)

Bu, uygulamanın bağı olduğu kuyruk yöneticisinin adı olan yerel kuyruk yöneticisinin adıdır.

Benzersiz bir ileti tanıtıcısı oluşturmak için adın ilk 12 karakteri kullanılır (bkz. MQMD- MsgId alanı). Bu nedenle, ileti tanıtıcılarının kuyruk yöneticisi ağında benzersiz olması için, iletişim kurabilen kuyruk yöneticilerinin ilk 12 karakterden farklı adları olmalıdır.


z/OS işletim sisteminde, ad altsistem adıyla aynıdır; bu ad boş olmayan 4 karakterle sınırlıdır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_Q_MGR_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir.

QSGName (MQCHAR4)

Bu, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun adıdır. Yerel kuyruk yöneticisi bir kuyruk paylaşım grubuna ait değilse, ad boş olur.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_QSG_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_QSG_NAME_LENGTH tarafından verilir.

 Bu öznitelik yalnızca z/OS üzerinde desteklenir.

QueueAccounting (MQLONG)

Bu, kuyruklara ilişkin hesap bilgilerinin toplanmasını denetler.

Değer aşağıdakilerden biridir:

MQMON_YOK

ACCTQ kuyruk muhasebesi özniteliğinin ayarından bağımsız olarak, kuyruklara ilişkin hesap verilerini toplamayın. Bu varsayılan değerdir.

MQMON_OFF

ACCTQ kuyruk özniteliğinde QMGR değerini belirleyen kuyruklara ilişkin hesap verilerini toplamayın.

MQMON_ON

ACCTQ kuyruk özniteliğinde QMGR değerini belirleyen kuyruklar için hesap verilerini toplayın.

Bu değerde yapılan değişiklikler yalnızca, öznitelikte yapılan değişiklikten sonra kuyruk yöneticisine yapılan bağlantılar için geçerlidir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_ACCOUNTING_Q seçicisini kullanın.

QueueMonitoring (MQLONG)

Bu, kuyrukların çevrimiçi izlenmesine ilişkin varsayılan ayarı belirtir.

QueueMonitoring kuyruk özniteliği MQMON_Q_MGR olarak ayarlanırsa, bu öznitelik kanal tarafından varsayılan değeri belirtir. Değer şöyle olabilir:

MQMON_OFF

Çevrimiçi izleme verilerinin toplanması kapalı. Bu, kuyruk yöneticisinin ilk varsayılan değeridir.

MQMON_YOK

Kuyruklar için çevrimiçi izleme verilerinin toplanması, **QueueMonitoring** özniteliklerinin ayarından bağımsız olarak kapatılmış olur.

MQMON_LOW

Çevrimiçi izleme verilerinin toplanması, veri toplama oranının düşük olduğu bir biçimde açık.

MQMON_ORTAMı

Çevrimiçi izleme verilerinin toplanması, veri toplama oranının orta düzeyde olduğu bir şekilde açıktır.

MQMON_YÜKSEKLİK

Çevrimiçi izleme verilerinin toplanması, veri toplamanın yüksek oranına sahip olarak açıktır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_MONITORING_Q seçicisini kullanın.

QueueStatistics (MQLONG)

Bu, kuyruklara ilişkin istatistik verilerinin toplanmasını denetler.

Aşağıdaki değerlerden biridir:

MQMON_YOK

QueueStatistics kuyruk özniteliğinin ayarından bağımsız olarak, kuyruklara ilişkin kuyruk istatistiklerini toplamayın. Bu varsayılan değerdir.

MQMON_OFF

QueueStatistics kuyruk özniteliğinde Kuyruk Yöneticisi 'ni belirten kuyruklara ilişkin istatistik verilerini toplamayın.

MQMON_ON

QueueStatistics kuyruk özniteliğinde Kuyruk Yöneticisi 'ni belirten kuyruklar için istatistik verilerini toplayın.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_STATISTICS_Q seçicisini kullanın.

ReceiveTimeout (MQLONG)

Bu, bir TCP/IP kanalının, etkin olmayan duruma dönmeden önce ortağından verileri (sağlıklı işletim bildirimleri de içinde olmak üzere) almak için ne kadar bekleyeceğini belirler. MQI kanalları için değil, yalnızca ileti kanalları için geçerlidir.

ReceiveTimeout değerinin tam anlamı, ReceiveTimeouttipinde belirtilen değerle değiştirilir.

ReceiveTimeoutTipi aşağıdakilerden birine ayarlanabilir:

- MQRCVTIME_EQUAL-bu değer, kanalın bekleyeceği saniye cinsinden sayıdır. 0-999999 aralığında bir değer belirtin.
- MQRCVTIME_ADD-Bu değer, kararlaştırılan HBINT ' ye eklenecek saniye cinsinden sayıdır ve bir kanalın ne kadar bekleyeceğini belirler. 1-999999 aralığında bir değer belirtin.
- MQRCVTIME_MULTIPLY-Bu değer, kararlaştırılan HBINT ' e uygulanacak bir çarpandır. 0 ya da 2-99 aralığında bir değer belirtin.

Varsayılan değer 0'dır.

Bir kanalın iş ortağından veri alma bekleme süresini zamanaşımına uğramasını önlemek için, ReceiveTimeouttipini MQRCVTIME_MULTIPLY ya da MQRCVTIME_EQUAL olarak ve ReceiveTimeout değerini 0 olarak ayarlayın.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_RECEIVE_TIMEOUT seçicisini kullanın.

ReceiveTimeoutAlt Sınır (MQLONG)

Bu, TCP/IP kanalının etkin olmayan duruma dönmeden önce ortağından sağlıklı işletim bildirimleri de içinde olmak üzere veri almak için bekleyeceği süre alt sınırıdır (saniye cinsinden).

MQI kanalları için değil, yalnızca ileti kanalları için geçerlidir. Değer 0 ile 999999 arasında olmalı ve varsayılan değer 0 olmalıdır.

ReceiveTimeoutkomutunu, TCP/IP kanal bekleme süresini HBINT ' nin kararlaştırılan değeriyle görelilik olarak hesaplanacağını ve sonuçta elde edilen değer bu parametrenin değerinden küçük olduğunu belirtmek için kullanırsanız, bunun yerine bu değer kullanılır.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_RECEIVE_TIMEOUT_MIN seçicisini kullanın.

ReceiveTimeoutTip (MQLONG)

Bu, bir TCP/IP kanalının ortağından etkin değil durumuna dönmeden önce sağlıklı işletim bildirimleri de içinde olmak üzere verileri ne kadar süreyle alacağını tanımlamak için ReceiveTimeout işlemine uygulanan niteleyicidir. MQI kanalları için değil, yalnızca ileti kanalları için geçerlidir.

Değer aşağıdakilerden biridir:

MQRCVTIME_MULTIPLY

ReceiveTimeout , bir kanalın ne kadar bekleyeceğini belirlemek için anlaşmalı HBINT değerine uygulanacak bir çarpan değeridir. Bu varsayılan değerdir.

MQRCVTIME_ADD

ReceiveTimeout , bir kanalın ne kadar süreyle bekleyeceğini belirlemek üzere anlaşmalı HBINT değerine eklenecek bir değerdir (saniye olarak).

MQRCVTIME_EQUAL

ReceiveTimeout , kanalın beklediği saniye cinsinden bir değerdir.

Bir kanal, ortağından veri almak üzere bekleme süresini durdurmak için ReceiveTimeouttipini MQRCVTIME_MULTIPLY ya da MQRCVTIME_EQUAL olarak ve ReceiveTimeout değerini 0 olarak ayarlayın.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_RECEIVE_TIMEOUT_TYPE seçicisini kullanın.

RemoteEvent (MQLONG)

Bu, uzak hata olaylarının oluşturulup oluşturulmayacağını denetler. Aşağıdaki değerlerden biridir:

MQEVR_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVR_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_REMOTE_EVENT seçicisini kullanın.

RepositoryName (MQCHAR48)

Bu, bu kuyruk yöneticisinin havuz yöneticisi hizmeti sağladığı bir kümenin adıdır. Kuyruk yöneticisi bu hizmeti birden çok küme için sağlıyorsa, *RepositoryNameList* kümeleri tanıtan bir ad listesi nesnesinin adını belirtir ve *RepositoryName* boş olur. *RepositoryName* ve *RepositoryNameList* ürünlerinden en az biri boş olmalıdır.

Bu özneliğin değerini belirlemek için MQINQ çağrısıyla MQCA_REPOSITORY_NAME seçicisini kullanın. Bu özneliğin uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir.

RepositoryNameList (MQCHAR48)

Bu, bu kuyruk yöneticisinin havuz yöneticisi hizmeti sağladığı kümelerin adlarını içeren bir ad listesi nesnesinin adıdır. Kuyruk yöneticisi bu hizmeti tek bir küme için sağlarsa, ad listesi nesnesi tek bir ad içerir. Diğer bir seçenek olarak, *RepositoryName* kümenin adını belirtmek için kullanılabilir; bu durumda *RepositoryNameList* boş olur. *RepositoryName* ve *RepositoryNameList* ürünlerinden en az biri boş olmalıdır.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_REPOSITORY_NAMELIST seçicisini kullanın. Bu özneliğin uzunluğu MQ_NAMELIST_NAME_LENGTH tarafından verilir.

ScyCase(MQCHAR8)

Kuyruk yöneticisinin güvenlik profili adlarını büyük harfli olarak mı, yoksa büyük harfli olarak mı destekleyip desteklemediğini belirtir.

Değer aşağıdakilerden biridir:

MQSCYC_UPPER

Güvenlik profili adları büyük harfli olmalıdır.

MQSCYC_MIXED

Güvenlik profili adları büyük harfli ya da büyük harfli olabilir.

Bu öznitelikte yapılan değişiklikler, *SecurityType* (MQSECTYPE_CLASSES) belirtilerek bir Güvenliği Yenile komutu çalıştırıldığında yürürlüğe girer.

z/OS Bu öznitelik yalnızca z/OSüzerinde desteklenir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_SECURITY_CASE seçicisini kullanın.

SharedQMgrAdı (MQLONG)

Bu, *ObjectQmgrName* , kuyruk paylaşım grubundaki başka bir kuyruk yöneticisine ait olduğunda, paylaşılan bir kuyruk için MQOPEN çağrısında *ObjectQmgrName* ' in kullanılıp kullanılmayacağını ya da yerel kuyruk yöneticisi olarak kabul edilip edilmeyeceğini belirtir.

Değer, aşağıdaki değerlerden herhangi biri olabilir:

MQSQM_USE

ObjectQmgrName kullanılır ve uygun iletim kuyruğu açılır.

MQSQM_IGNORE

Hedef kuyruk paylaşılıyorsa ve *ObjectQmgrName* aynı kuyruk paylaşım grubundaki bir kuyruk yöneticisiyse, açma işlemi yerel olarak gerçekleştirilir.

Bu öznitelik yalnızca z/OSüzerinde geçerlidir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_SHARED_Q_Q_MGR_NAME seçicisini kullanın.

SPLCAP

Advanced Message Security güvenlik yeteneklerinin bir kuyruk yöneticisi için kullanılabilir olup olmadığını belirtir.

MQCAP_DESTEKLI

Kuyruk yöneticisinin altında çalıştığı kuruluş için AMS bileşeni kuruluysa, bu varsayılan değerdir.

MQCAP_NOT_DESTEKLENMIYOR

SSLEvent (MQLONG)

Bu, TLS olaylarının oluşturulup oluşturulmayacağını belirtir.

Aşağıdaki değerlerden biridir:

MQEVR_ENABLED

TLS olaylarını şu şekilde oluştur:

MQRC_CHANNEL_SSL_HATA

MQEVR_DISABLED

TLS olayları oluşturmayın; bu varsayılan değerdir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_SSL_EVENT seçicisini kullanın.

SSLFIPSRequired (MQLONG)

Not: AIX, Linux, and Windows işletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerinde aranarak görüntülenebilir.

Bu, şifreleme, şifreleme donanımı yerine IBM MQ içinde yürütüldüğünde yalnızca FIPS onaylı algoritmaların kullanılacağını belirtmenizi sağlar. Şifreleme donanımı yapılandırıldıysa, kullanılan

şifreleme modülleri donanım ürünü tarafından sağlanan modüllerdir; bu modüller, kullanılmakta olan donanım ürününe bağlı olarak belirli bir düzeyde FIPS onaylı olabilir ya da olmayabilir.

Değer, aşağıdaki değerlerden biridir:

MQSSL_FIPS_NO

Kullanılmakta olan platformda desteklenen herhangi bir CipherSpec kullanın. Bu değer varsayılan değerdir.

MQSSL_FIPS_YES

Bu kuyruk yöneticisinden ve bu kuyruk yöneticisine tüm TLS bağlantılarında izin verilen CipherSpecs içinde yalnızca FIPS onaylı şifreleme algoritmalarını kullanın.

Bu parametre yalnızca z/OS, AIX, Linux, and Windows platformlarında geçerlidir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_SSL_FIPS_REQUIRED seçicisini kullanın.

İlgili görevler

MQI istemcisinde çalıştırma zamanında yalnızca FIPS onaylı CipherSpecs kullanılmasının belirtilmesi

İlgili başvurular

AIX, Linux, and Windows için Federal Bilgi İşleme Standartları (FIPS)

SSLKeyResetSayısı (MQLONG)

Bu, iletişimi başlatan TLS kanal ileti kanalı araçlarının (MCA ' lar) kanalda şifreleme için kullanılan gizli anahtarı sıfırladığını belirtir.

Değer, gizli anahtar yeniden anlaşılmadan önce kanalda gönderilen ve alınan toplam şifrenmemiş bayt sayısını temsil eder. Bayt sayısı, MCA tarafından gönderilen denetim bilgilerini içerir.

Değer, 0 ile 999 999 999 999 arasında bir sayıdır ve varsayılan değer 0 'dır. 1-32 KB aralığında bir TLS gizli anahtar sıfırlama sayısı belirtirseniz, TLS kanalları 32 KB gizli anahtar sıfırlama sayısını kullanır. Bu, küçük TLS gizli anahtar sıfırlama değerleri için ortaya çıkabilecek aşırı anahtar sıfırlamalarını işleme maliyetini önlemek içindir.

Başlatıcı kanal MCA tarafından gönderilen ve alınan şifrenmemiş baytların toplam sayısı belirtilen değeri aştığında gizli anahtar yeniden anlaşılır. Kanal sağlıklı işletim bildirimleri etkinleştirilirse, gizli anahtar, bir kanal sağlıklı işletim bildirim gönderilmeden ya da alınmadan önce ya da şifrenmemiş toplam bayt sayısı belirtilen değeri (hangisi önce gelirse) aştığında yeniden anlaşılır.

Yeniden anlaşma için gönderilen ve alınan bayt sayısı, kanal MCA tarafından gönderilen ve alınan denetim bilgilerini içerir ve her yeniden anlaşma oluştuğunda sıfırlanır.

Gizli anahtarların hiçbir zaman yeniden anlaşılmadığını belirtmek için 0 değerini kullanın.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_SSL_RESET_COUNT seçicisini kullanın.

StartStopOlayı (MQLONG)

Bu, başlangıç ve durdurma olaylarının oluşturulup oluşturulmayacağını denetler. Değer aşağıdakilerden biridir:

MQEVN_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVN_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_START_STOP_EVENT seçicisini kullanın.

StatisticsInterval (MQLONG)

Bu, istatistik izleme verilerinin izleme kuyruğuna ne sıklıkta (saniye cinsinden) yazılacağını belirtir.

Değer, varsayılan değeri 1800 (30 dakika) olan 0-604800 aralığında bir tamsayıdır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_STATISTICS_INTERVAL seçicisini kullanın.

SyncPoint (MQLONG)

Bu, yerel kuyruk yöneticisinin MQGET, MQPUT ve MQPUT1 çağrılarıyla iş birimlerini destekleyip desteklemediğini gösterir.

MQSP_KULLANILABILIR

İş birimleri ve eşitleme kullanılabilir.

MQSP_NOT_VAR

İş birimleri ve eşitleme kullanılamıyor.

- z/OS üzerinde bu değer hiçbir zaman döndürülmez.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_SYNCPOINT seçicisini kullanın.

TCPChannel (MQLONG)

Bu, TCP/IP iletim protokolünü kullanan, yürürlükteki ya da bağlanabilen istemci sayısı üst sınırdır.

Değer, varsayılan değeri 200 olan 0-9999 aralığında olmalıdır. 0 değerini belirterseniz, TCP/IP kullanılmaz.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_TCP_CHANNELS seçicisini kullanın.

TCPKeepAlive (MQLONG)

Bağlantının diğer ucunun hala kullanılabilir olup olmadığını denetlemek için TCP KEEPALIVE kullanılıp kullanılmayacağını belirler. Uygun değilse, kanal kapalıdır.

Değer aşağıdakilerden biridir:

MQTCPKEEP_YES

TCP profili yapılandırma veri kümesinde belirtildiği şekilde TCP KEEPALIVE değerini kullanın. KeepAliveInterval (KAINT) kanal özniteliğini belirtirseniz, ayarlandığı değer kullanılır.

MQTCPKEEP_NO

TCP KEEPALIVE özelliğini kullanmayın. Bu varsayılan değerdir.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_TCP_KEEP_ALIVE seçicisini kullanın.

TCPName (MQCHAR8)

Bu, TCPStackType değerine bağlı olarak kullanılacak tek ya da tercih edilen TCP/IP yığınının adıdır. Bu parametre yalnızca CINET birden çok yığın ortamında geçerlidir. Varsayılan değer TCPIP 'dir.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_TCP_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_TCP_NAME_LENGTH tarafından verilir.

TCPStackType (MQLONG)

Bu, kanal başlatıcının yalnızca TCPName 'de belirtilen TCP/IP yığını mı kullanacağını, yoksa isteğe bağlı olarak seçilen herhangi bir TCP/IP yığına mı bağlanabileceğini belirler. Bu parametre yalnızca CINET birden çok yığın ortamında geçerlidir.

Değer aşağıdakilerden biridir:

MQTCPSTACK_SINGLE

Kanal başlatıcı yalnızca TCPName 'de adı belirtilen TCP/IP adres alanlarını kullanabilir. Bu varsayılan değerdir.

MQTCPSTACK_MULTIPLE

Kanal başlatıcı, kullanabileceği herhangi bir TCP/IP adres alanını kullanabilir. Bir kanal ya da dinleyici için başka bir değer belirlenmemişse, TCPName parametresinde belirlenen değer kullanılır.

Bu öznitelik yalnızca z/OS üzerinde desteklenir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_TCP_STACK_TYPE seçicisini kullanın.

TraceRouteKaydı (MQLONG)

Bu, izleme rotası bilgilerinin kaydedilmesini denetler.

Değer aşağıdakilerden biridir:

MQRECORDING_DISABLED

İzleme rotası iletilerinin sonuna eklenmesine izin verilmez.

MQRECORDING_Q

İletileri sabit adlandırılmış kuyruğa yönlendirir.

MQRECORDING_MSG

İletin kendisini kullanarak iletilerin saptanan bir kuyruğa izlenmesini sağlar. Bu varsayılan değerdir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_TRACE_ROUTE_RECORDING seçicisini kullanın.

TriggerInterval (MQLONG)

Bu, tetikleyici iletilerin sayısını sınırlamak için kullanılan bir zaman aralığıdır (milisaniye cinsinden).

Bu yalnızca *TriggerType* MQTT_FIRST olduğunda geçerlidir. Bu durumda tetikleyici iletiler genellikle yalnızca kuyruğa uygun bir ileti geldiğinde ve kuyruk daha önce boş olduğunda üretilir. Ancak belirli koşullar altında, kuyruk boş olmasa da MQTT_FIRST tetikleyicisiyle ek bir tetikleyici iletileri oluşturulabilir. Bu ek tetikleyici iletileri her *TriggerInterval* milisaniyeden daha sık oluşturulmaz.

Tetikleme hakkında daha fazla bilgi için bkz. [Kanalları tetikleme](#).

Değer 0 'dan küçük ve 999 999 999 'dan büyük değil. Varsayılan değer 999 999 999 'dur.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_TRIGGER_INTERVAL seçicisini kullanın.

TriggerInterval (MQLONG)

Bu, tetikleyici iletilerin sayısını sınırlamak için kullanılan bir zaman aralığıdır (milisaniye cinsinden).

Bu yalnızca *TriggerType* MQTT_FIRST olduğunda geçerlidir. Bu durumda tetikleyici iletiler genellikle yalnızca kuyruğa uygun bir ileti geldiğinde ve kuyruk daha önce boş olduğunda üretilir. Ancak belirli koşullar altında, kuyruk boş olmasa da MQTT_FIRST tetikleyicisiyle ek bir tetikleyici iletileri oluşturulabilir. Bu ek tetikleyici iletileri her *TriggerInterval* milisaniyeden daha sık oluşturulmaz.

Tetikleme hakkında daha fazla bilgi için bkz. [Kanalları tetikleme](#).

Değer 0 'dan küçük ve 999 999 999 'dan büyük değil. Varsayılan değer 999 999 999 'dur.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_TRIGGER_INTERVAL seçicisini kullanın.

Sürüm (MQCFST)

Bu, VVRRMMFF olarak IBM MQ kodunun sürümüdür; burada:

VV-Sürüm

RR-Yayın Düzeyi

MM-Bakım düzeyi

FF-Düzeltilme düzeyi

XrCapability(MQLONG)

Bu, MQ Telemetry komutlarının kuyruk yöneticisi tarafından desteklenip desteklenmediğini denetler.

Değer aşağıdakilerden biridir:

MQCAP_DESTEKLI

MQ Telemetry bileşeni kurulu ve Telemetry komutları desteklenir.

MQCAP_NOT_DESTEKLENMIYOR

MQ Telemetry bileşeni kurulu değil.

Bu öznitelik yalnızca [Çoklu platformlar](#) üzerinde desteklenir.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_XR_CAPABILITY seçicisini kullanın.

Kuyruklara ilişkin öznitelikler

Beş tip kuyruk tanımlaması vardır. Bazı kuyruk öznitelikleri tüm kuyruk tipleri için geçerlidir; diğer kuyruk öznitelikleri yalnızca belirli kuyruk tipleri için geçerlidir.

Kuyruk tipleri

Kuyruk yöneticisi aşağıdaki kuyruk tanımlaması tiplerini destekler:

Yerel kuyruk

İletileri yerel bir kuyrukta saklayabilirsiniz.

z/OS z/OS ' da bunu paylaşılan ya da özel bir kuyruk yapabilirsiniz.

Programın bağlı olduğu kuyruk yöneticisine aitse, bir program kuyruğa *yerel* olarak bilinir. Yerel kuyruklardan ileti alabilir ve yerel kuyruklara ileti yerleştirebilirsiniz.

Kuyruk tanımlaması nesnesi, kuyruğa konan fiziksel iletilerin yanı sıra, kuyruğun tanımlama bilgilerini de içerir.

Yerel kuyruk yöneticisi kuyruğu

Kuyruk yerel kuyruk yöneticisinde var.

z/OS Kuyruk, z/OS üzerinde özel kuyruk olarak bilinir.

z/OS Paylaşılan kuyruk (yalnızca z/OS)

Kuyruk, paylaşılan havuza sahip olan kuyruk paylaşım grubuna ait olan tüm kuyruk yöneticilerinin erişebileceği paylaşılan bir havuzda var.

Kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisine bağlı uygulamalar, bu tipteki kuyruklara ileti yerleştirebilir ve kuyruklardan ileti kaldırabilir. Bu tür kuyruklar, yerel kuyruklarla etkili bir şekilde aynıdır. **QType** kuyruk özneliğinin değeri MQQT_LOCAL.

Yerel kuyruk yöneticisine bağlı uygulamalar, bu tipteki kuyruklara ileti yerleştirebilir ve kuyruklardan ileti kaldırabilir. **QType** kuyruk özneliğinin değeri MQQT_LOCAL.

Küme kuyruğu

İletileri, tanımlandığı kuyruk yöneticisindeki bir küme kuyruğunda saklayabilirsiniz. Küme kuyruğu, bir küme kuyruk yöneticisi tarafından barındırılan ve kümedeki diğer kuyruk yöneticilerinin kullanımına sunulan bir kuyruktur. **QType** kuyruk özneliğinin değeri MQQT_CLUSTER.

Kümedeki diğer kuyruk yöneticilerine bir küme kuyruğu tanımlaması tanıtılır. Kümedeki diğer kuyruk yöneticileri, iletileri karşılık gelen uzak kuyruk tanımlamasına gerek kalmadan bir küme kuyruğuna yerleştirebilirler. Küme kuyruğu, küme adı kullanılarak birden çok kümede duyurulabilir.

Bir kuyruk duyurulduğunda, kümedeki herhangi bir kuyruk yöneticisi bu kuyruğa ileti yerleştirebilir. Bir ileti koymak için kuyruk yöneticisinin, kuyruğun bulunduğu tam havuzlardan bunu öğrenmesi gerekir. Daha sonra iletiye bazı yöneltme bilgileri ekler ve iletiyi bir küme iletim kuyruğuna yerleştirir.

Kuyruk yöneticisi, birden çok iletim kuyruğundaki bir kümedeki diğer kuyruk yöneticilerine ilişkin iletileri saklayabilir. Bir kuyruk yöneticisini, iletileri birden çok küme iletim kuyruğunda iki farklı şekilde saklaması için yapılandırabilirsiniz. **DEFCLXQ** kuyruk yöneticisi özneliğini CHANNEL olarak ayarlarsanız, her küme gönderen kanalı için SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE ' den otomatik olarak farklı bir küme iletim kuyruğu yaratılır. CLCHNAME iletim kuyruğu seçeneğini bir

ya da daha çok küme gönderen kanalla eşleşecek şekilde ayarlarsanız, kuyruk yöneticisi, o iletim kuyruğundaki eşleşen kanallara ilişkin iletileri saklayabilir.



Uyarı: IBM WebSphere MQ 7.5'un önceki bir sürümünden yükseltilmiş bir kuyruk yöneticisiyle özel olarak ayrılmış SYSTEM.CLUSTER.TRANSMIT.QUEUES kullanıyorsanız, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE ' de SHARE/NOSHARE seçeneğinin **SHARE** olarak ayarlandığından emin olun.

z/OS Küme kuyruğu, IBM MQ for z/OS' daki bir kuyruk paylaşım grubunun üyeleri tarafından paylaşılan bir kuyruk olabilir.

Uzak kuyruk

Uzak kuyruk fiziksel bir kuyruk değildir; uzak kuyruk yöneticisinde varolan bir kuyruğun yerel tanımıdır. Uzak kuyruğun yerel tanımlaması, yerel kuyruk yöneticisine iletilerin uzak kuyruk yöneticisine nasıl yönlendirileceğini bildiren bilgiler içerir.

Yerel kuyruk yöneticisine bağlı uygulamalar bu tipteki kuyruklara ileti yerleştirebilirler; iletiler, iletileri uzak kuyruk yöneticisine yöneltmek için kullanılan yerel iletim kuyruğuna yerleştirilir. Uygulamalar uzak kuyruklardan ileti kaldıramaz. **QType** kuyruk özniteliğinin değeri MQQT_REMOTE.

Aşağıdakiler için bir uzak kuyruk tanımı da kullanabilirsiniz:

- Yanıt kuyruğu diğer ad alma

Bu durumda, tanımlamanın adı bir yanıt kuyruğunun adıdır. Daha fazla bilgi için bkz. [Yanıt kuyruğu diğer adları ve kümeleri](#).

- Kuyruk yöneticisi diğer adını alma

Bu durumda, tanımlamanın adı bir kuyruk yöneticisinin adı değil, diğer adıdır. Daha fazla bilgi için bkz. [Kuyruk yöneticisi diğer adları ve kümeleri](#).

Diğer ad kuyruğu

Bu bir fiziksel kuyruk değildir; yerel kuyruk, paylaşılan kuyruk, küme kuyruğu ya da uzak kuyruk için alternatif bir addir. Diğer adın çözüldüğü kuyruğun adı, diğer ad kuyruğunun tanımının bir parçasıdır.

Yerel kuyruk yöneticisine bağlı uygulamalar bu tipteki kuyruklara ileti yerleştirebilirler; iletiler, diğer adın çözüldüğü kuyruğa yerleştirilir. Diğer ad yerel kuyruğa, paylaşılan kuyruğa ya da yerel yönetim ortamı olan bir küme kuyruğuna çözümlenirse, uygulamalar bu tipteki kuyruklardan iletileri kaldırabilir. **QType** kuyruk özniteliğinin değeri: MQQT_ALIAS.

Model kuyruğu

Bu bir fiziksel kuyruk değildir; yerel bir kuyruğun yaratılabileceği bir kuyruk öznitelikleri kümeidir.

İletiler bu tipteki kuyruklarda saklanamaz.

kuyruk sınırları

IBM MQ 9.2.0'den, IBM MQ' un önceki yayınlarında kullanılan iki terabayt varsayılan sınırından daha fazlasını destekleyecek kuyrukları yapılandırma ve izleme seçeneğiniz vardır. Ayrıca, bir kuyruk dosyasının büyüyebileceği boyutu azaltma seçeneğiniz de vardır.

Kuyrukları yapılandırmanızı sağlamak için, yerel ve model kuyruklarında **MAXFSIZE** özniteliğini kullanabilir ve kuyrukları izlemek için kuyruk durumu özniteliklerini **CURFSIZE** ve **CURMAXFS** kullanabilirsiniz.

Daha fazla bilgi için [IBM MQ kuyruk dosyalarını değiştirme](#) başlıklı konuya bakın.

Kuyruk öznitelikleri

Bazı kuyruk öznitelikleri tüm kuyruk tipleri için geçerlidir; diğer kuyruk öznitelikleri yalnızca belirli kuyruk tipleri için geçerlidir. Bir özniteliğin uygulandığı kuyruk tipleri [Çizelge 561 sayfa 828](#) ve sonraki tablolarda gösterilir.

[Çizelge 561 sayfa 828](#) , kuyruklara özgü öznitelikleri özetler. Öznitelikler alfabetik sırayla açıklanır.

Not: Bu bölümde gösterilen özniteliklerin adları, MQINQ ve MQSET çağrılarıyla kullanılan açıklayıcı adlardır; adlar PCF komutlarıyla aynıdır. Öznitelikleri tanımlamak, değiştirmek ya da görüntülemek için MQSC komutları kullanıldığında, diğer kısa adlar kullanılır; ayrıntılar için MQSC komutları konusuna bakın.

Aşağıdaki tabloda sütunlar aşağıdaki gibi geçerlidir:

- Yerel kuyruklara ilişkin kolon, paylaşılan kuyruklara da uygulanır.
- Model kuyruklarına ilişkin kolon, model kuyruğundan yaratılan yerel kuyruk tarafından devralınan öznitelikleri gösterir.
- Küme kuyruklarına ilişkin kolon, küme kuyruğu tek başına sorgu için ya da sorgu ve çıkış için açıldığında sorulabilecek öznitelikleri gösterir. Başka öznitelikler sorulursa, çağrı tamamlanma kodu MQCC_WARNING ve neden kodu MQRC_SELEC_CTOR_NOT_FOR_TYPE (2068) döndürür.

Küme kuyruğu sorgunun yanı sıra bir ya da daha fazla giriş, göz atma ya da küme için açılırsa, bunun yerine yerel kuyruklara ilişkin sütun geçerli olur.

Küme kuyruğu tek başına sorgu için ya da sorgu ve çıkış için açılırsa ve temel kuyruk yöneticisi adını belirtirse, bunun yerine yerel kuyruklara ilişkin sütun geçerli olur.

Çizelge 561. Kuyruklara ilişkin öznitelikler						
Öznitelik	Açıklama	Yerel	Modeli	Diğer Ad	Uzak	Küme
<u>AlterationDate</u>	Tanımının en son değiştirildiği tarih	X		X	X	
<u>AlterationTime</u>	Tanımlamanın en son değiştirildiği zaman	X		X	X	
<u>BackoutRequeueQName</u>	Aşırı geriletme yeniden kuyruğa alma kuyruk adı	X	X			
<u>BackoutThreshold</u>	Geriletme eşiği	X	X			
<u>BaseQName</u>	Diğer adın çözümleneceği kuyruk adı			X		
<u>CFStrucName</u>	Bağlaşım-tesis yapısı adı	X	X			
<u>CLCHNAME</u>	Küme gönderen kanal adları	✓	✓			
<u>ClusterName</u>	Kuyruğun ait olduğu kümenin adı	X		X	X	X
<u>ClusterNameList</u>	Kuyruğun ait olduğu kümelerin adlarını içeren ad listesi nesnesinin adı	X		X	X	
<u>CLWLQueuePriority</u>	Küme iş yükü kuyruğu önceliği	X		X	X	X
<u>CLWLQueueRank</u>	Küme iş yükü kuyruğu sırası	X		X	X	X
<u>CLWLUseQ</u>	Uzak kuyruk kullan	X				
<u>CreationDate</u>	Kuyruğun yaratıldığı tarih	X				
<u>CreationTime</u>	Kuyruğun yaratıldığı saat	X				
<u>CurrentQDepth</u>	Yürürlükteki kuyruk derinliği	X				
<u>DefaultPutResponse</u>	Varsayılan koyma yanıtı	✓	✓	✓	✓	
<u>DefBind</u>	Varsayılan bağ tanımı	X		X	X	X
<u>DefinitionType attribute</u>	Kuyruk tanımlaması tipi	X	X			
<u>DefInputOpenOption</u>	Varsayılan giriş açma seçeneği	X	X			
<u>DefPersistence</u>	Varsayılan ileti kalıcılığı	X	X	X	X	X
<u>DefPriority</u>	Varsayılan ileti önceliği	✓	✓	✓	✓	✓
<u>DefReadAhead</u>	Varsayılan önden okuma	X	X	X		
<u>DistLists</u>	Dağıtım listesi desteği	X	X			

Çizelge 561. Kuyruklara ilişkin öznitelikler (devamı var)

Öznitelik	Açıklama	Yerel	Modeli	Diğer Ad	Uzak	Küme
HardenGetBackout	Doğru bir geriletme sayısının korunup korunmayacağını belirler	X	X			
IndexType	Dizin tipi	X	X			
InhibitGet	Kuyruk için alma işlemlerine izin verilip verilmediğini belirler	X	X	X		
InhibitPut	Kuyruğa ilişkin koyma işlemlerine izin verilip verilmediğini belirler	X	X	X	X	X
InitiationQName	Başlatma kuyruğunun adı	X	X			
MaxMsgLength	İleti uzunluğu üst sınırı (byte)	X	X			
MaxQDepth	Kuyruk derinliği üst sınırı	X	X			
MsgDeliverySequence attribute	İleti teslim sırası	X	X			
NonPersistentMessage Class	Kalıcı olmayan iletiler için güvenilirlik hedefi	X	X			
OpenInputCount	Giriş için açma sayısı	X				
OpenOutputCount	Çıkış için açma sayısı	X				
PropertyControl	Özellik denetimi	✓	✓	✓		
ProcessName	İşlem adı	X	X			
QDepthHighEvent attribute	Kuyruk Derinliği Yüksek olaylarının oluşturulup oluşturulmayacağını belirler	X	X			
QDepthHighLimit	Kuyruk derinliği için üst sınır	X	X			
QDepthLowEvent attribute	Kuyruk Derinliği Düşük olaylarının oluşturulup oluşturulmayacağını belirler	X	X			
QDepthLowLimit attribute	Kuyruk derinliği için alt sınır	X	X			
QDepthMaxEvent	Tüm Kuyruk olaylarının oluşturulup oluşturulmayacağını belirler	X	X			
QDesc	Kuyruk tanımlaması	X	X	X	X	X
QName	Kuyruk adı	X		X	X	X
QServiceInterval	Kuyruk hizmeti aralığı için hedef	X	X			
QServiceIntervalEvent attribute	Hizmet Aralığı Yüksek ya da Hizmet Aralığı Tamam olaylarının oluşturulup oluşturulmayacağını belirler	X	X			
QSGDisp attribute	Kuyruk paylaşım grubu yok etme	X		X	X	
QueueAccounting	Kuyruk muhasebesi veri toplaması	X	X	X	X	X
QueueMonitoring	Kuyruklar için çevrimiçi izleme verileri	X	✓			
QueueStatistics	Kuyruk istatistikleri veri toplama	X	X	X	X	X
QType	Kuyruk tipi	X		X	X	X
RemoteQMgrName	Uzak kuyruk yöneticisinin adı				X	
RemoteQName	Uzak kuyruğun adı				X	
RetentionInterval	Alıkoyma aralığı	X	X			

Çizelge 561. Kuyruklara ilişkin öznitelikler (devamı var)						
Öznitelik	Açıklama	Yerel	Modeli	Diğer Ad	Uzak	Küme
<u>Scope</u>	Kuyruk için bir girişin hücre dizininde de bulunup bulunmadığını belirler	X		X	X	
<u>Shareability</u>	Kuyruk paylaşılabilir	X	X			
<u>StorageClass</u>	Kuyruk için depolama sınıfı	X	X			
<u>TriggerControl</u>	Tetikleme denetimi	X	X			
<u>TriggerData</u>	Tetikleyici verileri	X	X			
<u>TriggerDepth</u>	Tetikleme derinliği	X	X			
<u>TriggerMsgPriority</u>	Tetikleyiciler için eşik iletisi önceliği	X	X			
<u>TriggerType</u>	Tetikleme Tipi	X	X			
<u>Usage attribute</u>	Kuyruk kullanımı	X	X			
<u>XmitQName</u>	İletim kuyruğu adı				X	

İlgili kavramlar

Küme kuyrukları

Yerel kuyruklar

Hangi küme iletim kuyruğunun kullanılacağını seçme

AlterationDate (MQCHAR12)

Tanımının en son değiştirildiği tarih.

Çizelge 562. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Bu, tanımın en son değiştirildiği tarihtir. Tarihin biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için iki sondaki boşlukla doldurulur (örneğin, 1992-09-23--), burada -- iki boş karakteri temsil eder).

Kuyruk yöneticisi çalışırken, belirli özniteliklerin değerleri (örneğin, *CurrentQDepth*) değişir. Bu özniteliklerde yapılan değişiklikler *AlterationDate* ögesini etkilemez.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_ALTERATION_DATE seçicisini kullanın. Bu özneliğin uzunluğu MQ_DATE_LENGTH tarafından verilir.

AlterationTime (MQCHAR8)

Tanımlamanın en son değiştirildiği zaman.

Çizelge 563. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Bu, tanımın en son değiştirildiği zamandır. Saatin biçimi HH.MM.SS 24 saatlik zaman biçimini kullanır; saat 10 'dan küçükse (örneğin, 09.10.20), başında sıfır vardır.

- z/OS işletim sisteminde, Greenwich Ortalama Saati (GMT), sistem saatinin GMT ' ye doğru olarak ayarlanmasına tabidir.
- Diğer ortamlarda, saat yerel saattir.

Kuyruk yöneticisi çalışırken, belirli özniteliklerin değerleri (örneğin, *CurrentQDepth*) değişir. Bu özniteliklerde yapılan değişiklikler *AlterationTime* ögesini etkilemez.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_ALTERATION_TIME seçicisini kullanın. Bu özneliğin uzunluğu MQ_TIME_LENGTH tarafından verilir.

BackoutRequeueQName (MQCHAR48)

Bu, aşırı geriletme yeniden kuyruğa alma kuyruğu adıdır. Kuyruk yöneticisi, değerinin sorgulanmasına izin vermenin yanı sıra, bu özneliğin değerine dayalı olarak herhangi bir işlem gerçekleştirmez.

Çizelge 564. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

WebSphere Application Server içinde çalışan uygulamalar ve IBM MQ Application Server Facilities kullanan uygulamalar, geriletmiş iletilerin nereye gitmesi gerektiğini belirlemek için bu özneliği kullanır. Diğer tüm uygulamalar için kuyruk yöneticisi, özneliğin değerine dayalı olarak herhangi bir işlem gerçekleştirmez.

IBM MQ classes for JMS , *BackoutThreshold* özneliğinde belirttiği gibi, zaten geriletmiş olan bir iletinin en fazla kaç kez aktarılacağını belirlemek için bu özneliği kullanır.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_BACKOUT_REQ_Q_NAME seçicisini kullanın. Bu özneliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

BackoutThreshold (MQLONG)

Bu geriletme eşiği. Kuyruk yöneticisi, değerinin sorgulanmasına izin vermenin yanı sıra, bu özneliğin değerine dayalı olarak herhangi bir işlem gerçekleştirmez.

Çizelge 565. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

WebSphere Application Server içinde çalışan uygulamalar ve IBM MQ Application Server Facilities kullanan uygulamalar, bir iletinin geriletilmesinin gerekip gerekmediğini belirlemek için bu özneliği kullanır. Diğer tüm uygulamalar için kuyruk yöneticisi, özneliğin değerine dayalı olarak herhangi bir işlem gerçekleştirmez.

IBM MQ classes for JMS , bir iletinin *BackoutRequeueQName* özneliğiyle belirtilen kuyruğa aktarılmadan önce kaç kez geriletilmesine izin verileceğini saptamak için bu özneliği kullanır.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_BACKOUT_THRESHOLD seçicisini kullanın.

BaseQName (MQCHAR48)

Bu, yerel kuyruk yöneticisine tanımlanan bir kuyruğun adıdır.

Çizelge 566. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
		X		

(Kuyruk adlarıyla ilgili ek bilgi için [MQOD- ObjectName alanınabakın.](#)) Kuyruk aşağıdaki tiplerden biridir:

MQQT_LOCAL

Yerel kuyruk.

MQQT_REMOTE

Uzak kuyruğun yerel tanımlaması.

MQOT_CLUSTER

Küme kuyruğu.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_BASE_Q_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

BaseType (MQCFIN)

Diğer adın çözümleneceği nesnenin tipi.

Çizelge 567. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
		X		

Aşağıdaki değerlerden biridir:

MQOT_Q

Temel nesne tipi bir kuyruk

MQOT_KONU

Temel nesne tipi bir konudur

CFStrucName (MQCHAR12)

Bu, kuyruktaki iletilerin saklandığı bağlaşım olanağı yapısının adıdır. Adın ilk karakteri A-Z aralığındadır ve diğer karakterler A-Z, 0-9 ya da boşluk aralığındadır.

Çizelge 568. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bağlaşım olanağındaki yapının tam adını almak için, **QSGName** kuyruk yöneticisi özniteliğinin değerini **CFStrucName** kuyruk özniteliğinin değeriyle birlikte sonek.

Bu öznitelik yalnızca paylaşılan kuyruklar için geçerlidir; *QSGDiSp*, MQQSGD_SHARED değerine sahip değilse yoksayılır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_CF_STRUC_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_CF_STRUC_NAME_LENGTH tarafından verilir.

z/OS Bu öznitelik yalnızca z/OS üzerinde desteklenir.

ClusterChannelAdı (MQCHAR20)

ClusterChannelName, bu kuyruğu bir iletim kuyruğu olarak kullanan küme gönderen kanallarının sosyal adıdır. Öznitelik, hangi küme-gönderen kanallarının bu küme iletim kuyruğundan bir küme-alıcı kanalına ileti gönderdiğini belirtir.

Çizelge 569. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Varsayılan kuyruk yöneticisi yapılandırması, tek bir iletim kuyruğundan (SYSTEM.CLUSTER.TRANSMIT.QUEUE) ileti göndermek için tüm küme gönderen kanalları içindir. Varsayılan yapılanış, **DefClusterXmitQueueType** kuyruk yöneticisi özniteliği değiştirilerek değiştirilebilir. Özniteliğin varsayılan değeri SCTQ' dur. Değeri CHANNEL olarak değiştirebilirsiniz. **DefClusterXmitQueueType** özniteliğini CHANNEL olarak ayarlarsanız, her küme gönderen kanalı

varsayılan olarak belirli bir küme iletim kuyruğunu (SYSTEM . CLUSTER . TRANSMIT . *ChannelName*) kullanır.

İletim kuyruğu özniteliğini *ClusterChannelName* özniteliğini kümeyi gönderen bir kanala el ile ayarlayabilirsiniz. Kümeyi gönderen kanal tarafından bağlanan kuyruk yöneticisine yönlendirilen iletiler, küme gönderen kanalı tanımlayan iletim kuyruğunda saklanır. Bunlar varsayılan küme iletim kuyruğunda saklanmaz. *ClusterChannelName* özniteliğini boş olarak ayarlarsanız, kanal yeniden başlatıldığında kanal varsayılan küme iletim kuyruğuna geçer. Varsayılan kuyruk, kuyruk yöneticisinin değerine *DefClusterXmitQueueType* (Tip) özniteliğine bağlı olarak SYSTEM . CLUSTER . TRANSMIT . *ChannelName* ya da SYSTEM . CLUSTER . TRANSMIT . QUEUE şeklinde olur.

ClusterChannelName içinde yıldız işaretlerini ("*") belirterek, bir iletim kuyruğunu bir küme gönderen kanalları kümesiyle ilişkilendirebilirsiniz. Yıldız işaretleri, kanal adı dizgisinin başında, sonunda ya da ortasında herhangi bir sayıda yer olabilir. **ClusterChannelName** 20 karakterlik bir uzunlukla sınırlıdır: MQ_CHANNEL_NAME_LENGTH.

ClusterName (MQCHAR48)

Bu, kuyruğun ait olduğu kümenin adıdır.

Çizelge 570. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Kuyruk birden çok kümeye aitse *ClusterNameList* , kümeleri tanımlayan bir ad listesi nesnesinin adını belirtir ve *ClusterName* boş olur. *ClusterName* ve *ClusterNameList* ürünlerinden en az biri boş olmalıdır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_CLUSTER_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_CLUSTER_NAME_LENGTH tarafından verilir.

ClusterNameList (MQCHAR48)

Bu, bu kuyruğun ait olduğu kümelerin adlarını içeren bir ad listesi nesnesinin adıdır.

Çizelge 571. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Kuyruk tek bir kümeye aitse, ad listesi nesnesi tek bir ad içerir. Diğer bir seçenek olarak, *ClusterName* kümenin adını belirtmek için kullanılabilir; bu durumda *ClusterNameList* boş olur. *ClusterName* ve *ClusterNameList* ürünlerinden en az biri boş olmalıdır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_CLUSTER_NAMELIST seçicisini kullanın. Bu özniteliğin uzunluğu MQ_NAMELIST_NAME_LENGTH tarafından verilir.

CLWLQueuePriority (MQLONG)

Bu, küme iş yükü kuyruğu önceliğidir; kuyruğun önceliğini gösteren 0-9 aralığındaki bir değerdir.

Çizelge 572. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Daha fazla bilgi için bkz. [Küme kuyrukları](#).

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_CLWL_Q_PRIORITY seçicisini kullanın.

CLWLQueueRank (MQLONG)

Bu, küme iş yükü kuyruğu sırasıdır; kuyruğun sırasını gösteren, 0-9 aralığındaki bir değerdir.

Çizelge 573. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Daha fazla bilgi için bkz. [Küme kuyrukları](#).

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_CLWL_Q_RANK seçicisini kullanın.

CLWLUseQ (MQLONG)

Bu, hedef kuyrukta hem yerel yönetim ortamı, hem de en az bir uzak küme yönetim ortamı olduğunda MQPUT ' nin davranışını tanımlar. Koyma değeri bir küme kanalından kaynaklandysa, bu öznitelik geçerli değildir.

Çizelge 574. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Değer aşağıdakilerden biridir:

MQCLWL_USEQ_ANY

Uzak ve yerel kuyrukları kullanın.

MQCLWL_USEQ_LOCAL

Uzak kuyrukları kullanmayın.

MQCLWL_USEQ_AS_Q_MGR

Tanımlı kuyruk yöneticisinin MQIA_CLWL_USEQ ögesinden edin.

Daha fazla bilgi için bkz. [Küme kuyrukları](#).

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_CLWL_USEQ seçicisini kullanın. Bu özniteliğin uzunluğu MQ_CLWL_USEQ_LENGTH tarafından verilir.

CreationDate (MQCHAR12)

Bu, kuyruğun yaratıldığı tarihtir.

Çizelge 575. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Tarihin biçimi şöyledir: YYYY-MM-DD, uzunluğu 12 bayt (örneğin, 2013-09-23--), burada -- 2 boş karakteri gösterir) yapmak için iki sondaki boşlukla doldurulur.

- IBM sistemlerinde, bir kuyruğun yaratılma tarihi, kuyruğu temsil eden temel işletim sistemi varlığından (dosya ya da kullanıcı alanı) farklı olabilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_CREATION_DATE seçicisini kullanın. Bu özniteliğin uzunluğu MQ_CREATION_DATE_LENGTH tarafından verilir.

CreationTime (MQCHAR8)

Bu, kuyruğun yaratıldığı zamandır.

Çizelge 576. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Saatin biçimi HH.MM.SS 24 saatlik zaman biçimini kullanır; saat 10 'dan küçükse (örneğin, 09.10.20), başında sıfır vardır.

- z/OS işletim sisteminde, Greenwich Ortalama Saati (GMT), sistem saatinin GMT ' ye doğru olarak ayarlanmasına tabidir.
- Diğer ortamlarda, saat yerel saattir.
- IBM sistemlerinde, bir kuyruğun yaratılma zamanı, kuyruğu temsil eden temel işletim sistemi varlığından (dosya ya da kullanıcı alanı) farklı olabilir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_CREATION_TIME seçicisini kullanın. Bu özneliğin uzunluğu MQ_CREATION_TIME_LENGTH tarafından verilir.

CurrentQDepth (MQLONG)

Bu, kuyrukta bulunan iletilerin sayısıdır.

Çizelge 577. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

MQPUT çağrısı sırasında ve MQGET çağrısının geriletme sırasında artırılır. Göz atmayan bir MQGET çağrısı sırasında ve bir MQPUT çağrısının geriletme sırasında azaltılır. Bunun sonucu, sayımın bir iş birimi içinde kuyruğa konan, ancak henüz kesinleştirilmemiş iletileri içermesi, ancak bunların MQGET çağrısı tarafından alınmaya uygun olmamaları olabilir. Benzer şekilde, MQGET çağrısı kullanılarak bir iş birimi içinde alınan, ancak henüz kesinleştirilmemiş iletileri dışlar.

Sayım, süre bitimini geçmiş, ancak henüz atılmamış olan iletileri de içerir; ancak, bu iletiler alınmaya uygun değildir. Ek bilgi için [MQMD-Expiry alanı](#) başlıklı konuya bakın.

İş birimi işleme ve iletilerin bölümlenmesi *CurrentQDepth* ' un *MaxQDepth* değerini aşmasına neden olabilir. Ancak, bu durum iletilerin alınabilirliğini etkilemez; kuyruktaki tüm iletiler, MQGET çağrısı kullanılarak olağan şekilde alınabilir.

Kuyruk yöneticisi çalışırken bu özneliğin değeri dalgalandı.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_CURRENT_Q_DEPTH seçicisini kullanın.

DefaultPutYanıt (MQLONG)

Bir uygulama MQPMO_RESPONSE_AS_Q_DEF belirtilirken kuyruğa koyma işlemleri için kullanılacak yanıt tipini belirtir.

Çizelge 578. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	

Aşağıdaki değerlerden biridir:

MQPRT_SYNC_RESPONSE

Koyma işlemi zamanuyumlu olarak yayınlandı ve bir yanıt döndürüldü.

MQPRT_ASYNC_RESPONSE (mqprt_async_yanıtı)

Koyma işlemi zamanuyumsuz olarak yayınlanır ve MQMD alanlarının bir alt kümesi döndürülüyor.

DefBind (MQLONG)

MQOPEN çağrısında MQOO_BIND_AS_Q_DEF belirtildiğinde ve kuyruk bir küme kuyruğu olduğunda kullanılan varsayılan bağ tanımdır.

Çizelge 579. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Değer aşağıdakilerden biridir:

MQBND_BIND_ON_OPEN

Bağ tanımı MQOPEN çağrısı tarafından düzeltildi.

MQBND_BIND_NOT_FIXED

Bağlama düzeltilmedi.

MQBND_BIND_ON_GROUP

Bir uygulamanın, bir grup iletinin aynı hedef örneğe ayrılmasını istemesini sağlar.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_DEF_BIND seçicisini kullanın.

DefinitionType (MQLONG)

Bu, kuyruğun nasıl tanımlandığını gösterir.

Çizelge 580. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Değer aşağıdakilerden biridir:

MQQDT_ÖNCEDEN TANIMLI

Kuyruk, sistem denetimcisi tarafından yaratılan kalıcı bir kuyruktur; kuyruğu yalnızca sistem denetimcisi silebilir.

Önceden tanımlanmış kuyruklar DEFINE MQSC komutu kullanılarak yaratılır ve yalnızca DELETE MQSC komutu kullanılarak silinebilir. Model kuyruklarından önceden tanımlanmış kuyruklar yaratılamaz.

Komutlar bir işletmen tarafından ya da yetkili bir kullanıcı tarafından komut giriş kuyruğuna komut iletisi göndererek verilebilir (ek bilgi için [CommandInputQName](#) özniteliğine bakın).

MQQDT_PERMANENT_DYNAMIC

Kuyruk, nesne tanımlayıcı MQOD ' de belirtilen bir model kuyruğu adıyla MQOPEN çağrısı yayınlayan bir uygulama tarafından yaratılan kalıcı bir kuyruktur. Model kuyruğu tanımlaması, **DefinitionType** özniteliği için MQQDT_PERMANENT_DYNAMIC değerine sahiptir.

Bu kuyruk tipi, MQCLOSE çağrısı kullanılarak silinebilir. Daha fazla ayrıntı için bkz. "[MQCLOSE-Nesneyi kapat](#)" sayfa 642 .

Kalıcı dinamik kuyruk için **QSGDisp** özniteliğinin değeri MQQSGD_Q_MGR 'dir.

MQQDT_TEMPORARY_DYNAMIC

Kuyruk, MQOD nesne tanımlayıcısında belirtilen bir model kuyruğu adıyla MQOPEN çağrısı yayınlayan bir uygulama tarafından yaratılan geçici bir kuyruktur. Model kuyruğu tanımlaması, **DefinitionType** özniteliği için MQQDT_TEMPORARY_DYNAMIC değerine sahiptir.

Bu kuyruk tipi, yaratan uygulama tarafından kapatıldığında MQCLOSE çağrısı tarafından otomatik olarak silinir.

Geçici bir dinamik kuyruk için **QSGDisp** özniteliğinin değeri MQQSGD_Q_MGR 'dir.

MQQDT_SHARED_DYNAMIC

Kuyruk, nesne tanımlayıcı MQOD ' de belirtilen bir model kuyruğu adıyla MQOPEN çağrısı yayınlayan bir uygulama tarafından yaratılan paylaşılan bir kalıcı kuyruktur. Model kuyruğu tanımlaması, **DefinitionType** özniteliği için MQQDT_SHARED_DYNAMIC değerine sahiptir.

Bu kuyruk tipi, MQCLOSE çağrısı kullanılarak silinebilir. Daha fazla ayrıntı için bkz. “MQCLOSE-Nesneyi kapat” sayfa 642 .

Paylaşılan bir dinamik kuyruk için **QSGDisp** özniteliğinin değeri MQQSGD_SHARED 'dir.

Model kuyruğu tanımlamasındaki bu öznitelik, model kuyruklarının her zaman önceden tanımlanmış olması nedeniyle, model kuyruğunun nasıl tanımlandığını göstermez. Bunun yerine, model kuyruğundaki bu özniteliğin değeri, MQOPEN çağrısı kullanılarak model kuyruğu tanımlamasından yaratılan her dinamik kuyruğun *DefinitionType* değerini saptamak için kullanılır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_DEFINITION_TYPE seçicisini kullanın.

DefInputOpenOption (MQLONG)

Bu, giriş için kuyruğun açılacağı varsayılan yoldur.

Çizelge 581. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Kuyruk açıldığında MQOPEN çağrısında MQOO_INPUT_AS_Q_DEF seçeneği belirtilirse bu geçerlidir. Değer aşağıdakilerden biridir:

MQOO_INPUT_EXCLUSIVE

Dışlayıcı erişimi olan iletileri almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Kuyruk şu anda herhangi bir tip giriş için bu ya da başka bir uygulama tarafından açıksa (MQOO_INPUT_SHARED ya da MQOO_INPUT_EXCLUSIVE), çağrı MQRC_OBJECT_IN_USE neden koduyla başarısız olur.

MQOO_INPUT_SHARED

Paylaşılan erişimi olan iletileri almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Kuyruk şu anda MQOO_INPUT_SHARED olan bu uygulama ya da başka bir uygulama tarafından açıksa, ancak kuyruk şu anda MQOO_INPUT_EXCLUSIVE ile açıksa, MQRC_OBJECT_IN_USE neden koduyla başarısız olursa çağrı başarılı olabilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_DEF_INPUT_OPEN_OPTION seçicisini kullanın.

DefPersistence (MQLONG)

Bu, kuyruktaki iletilerin varsayılan kalıcılığıdır. İleti konduğunda ileti tanımlayıcısında MQPER_PERSISTENCE_AS_Q_DEF belirtilirse bu geçerlidir.

Çizelge 582. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Kuyruk adı çözüm yolunda birden çok tanımlama varsa, varsayılan kalıcılık, MQPUT ya da MQPUT1 çağrısı sırasında yoldaki *ilk* tanımlamasında bu özniteliğin değerinden alınır. Bu şöyle olabilir:

- Diğer ad kuyruğu
- Yerel bir kuyruk
- Uzak kuyruğun yerel tanımlaması
- Kuyruk yöneticisi diğer adı
- Bir iletim kuyruğu (örneğin, *DefXmitQName* kuyruğu)

Değer aşağıdakilerden biridir:

MQPER_KALICI

İleti, sistem hatalarından kurtulur ve kuyruk yöneticisi yeniden başlatılır. Kalıcı iletilerin yerleştirilemediği yer:

- Geçici dinamik kuyruklar
- CFLEVEL (2) ya da altındaki bir CFSTRUCT nesnesiyle eşleşen ya da CFSTRUCT nesnesinin RECOVER (NO) olarak tanımlandığı paylaşılan kuyruklar.

Kalıcı iletiler kalıcı dinamik kuyruklara ve önceden tanımlanmış kuyruklara yerleştirilebilir.

MQPER_NOT_KALICI

İleti, olağan durumda sistem hatalarından ya da kuyruk yöneticisi yeniden başlatmadan kurtulmaz. Kuyruk yöneticisi yeniden başlatılırken yardımcı bellekte iletinin bozulmamış bir kopyası bulursa da bu durum geçerlidir.

Paylaşılan kuyruklar söz konusu olduğunda, *yapma* kalıcı olmayan iletiler kuyruk paylaşım grubundaki kuyruk yöneticilerinin yeniden başlatılmasına devam ediyor, ancak paylaşılan kuyruklarda ileti saklamak için kullanılan bağlaşım olanağının başarısızlıklarına dayanmıyor.

Kalıcı ve kalıcı olmayan iletiler aynı kuyruksa bulunabilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_DEF_PERSISTENCE seçicisini kullanın.

DefPriority (MQLONG)

Bu, kuyruksaki iletiler için varsayılan önceliklerdir. İleti kuyruğa konduğunda ileti tanımlayıcısında MQPRI_PRIORITY_AS_Q_DEF belirtilirse bu geçerlidir.

Çizelge 583. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Kuyruk adı çözme yolunda birden çok tanımlama varsa, iletinin varsayılan önceliği, koyma işlemi sırasında yoldaki *ilk* tanımlamasında bu özniteliğin değerinden alınır. Bu şöyle olabilir:

- Diğer ad kuyruğu
- Yerel bir kuyruk
- Uzak kuyruğun yerel tanımlaması
- Kuyruk yöneticisi diğer adı
- Bir iletim kuyruğu (örneğin, *DefXmitQName* kuyruğu)

Bir iletinin kuyruğa yerleştirilme şekli, kuyruğun **MsgDeliverySequence** özniteliğinin değerine bağlıdır:

- **MsgDeliverySequence** özniteliği MQMDS_PRIORITY ise, bir iletinin kuyruğa yerleştirileceği mantıksal konum, ileti tanımlayıcısındaki *Priority* alanının değerine bağlıdır.
- **MsgDeliverySequence** özniteliği MQMDS_FIFO ise, iletiler, ileti tanımlayıcısındaki *Priority* alanının değerine bakılmaksızın, çözülmüş kuyruğun *DefPriority* değerine eşit bir önceliğe sahip gibi kuyruğa yerleştirilir. Ancak, *Priority* alanı iletiyi koyan uygulama tarafından belirtilen değeri korur. Ek bilgi için [MsgDeliverySequence attribute](#) başlıklı konuya bakın.

Öncelikler sıfır (en düşük)- *MaxPriority* (en yüksek) aralığındadır; bkz. [MaxPriority özniteliği](#).

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_DEF_PRIORITY seçicisini kullanın.

DefRead(MQLONG)

İstemciye teslim edilen kalıcı olmayan iletiler için varsayılan önden okuma davranışını belirtir.

Çizelge 584. Bu özneliğin geçerli olduđu kuyruk tipleri				
Yerel	Modeli	Diđer Ad	Uzak	Küme
X	X	X		

DefReadÖnden ařağıdaki deđerlerden birine ayarlanabilir:

MQREADA_NO

Kalıcı olmayan iletiler, bir uygulama istemeden önce istemciye gönderilmez. İstemci olağıandışı bir şekilde sona ererse, en çok bir kalıcı olmayan ileti kaybolabilir.

MQREADA_YES

Kalıcı olmayan iletiler, bir uygulama istemeden önce istemciye gönderilir. İstemci olağıandışı bir şekilde sona ererse ya da istemci gönderildiğı tüm iletileri tüketmezse, kalıcı olmayan iletiler kaybolabilir.

MQREADA_DISABLED

Bu kuyruk için etkinleştirilmemiş, kalıcı olmayan iletilerin önden okunması. İletiler, istemci uygulaması tarafından önden okuma istenip istenmemesinden bağımsız olarak istemciye gönderilmez.

Bu özneliğin deđerini saptamak için MQINQ çağrısıyla MQIA_DEF_READ_ÖNDEN seçiciyi kullanın.

DefPResp (MQLONG)

Varsayılan koyma yanıtı tipi (DEFPRESP) özneliğı, MQPMO içindeki PutResponsetipi MQPMO_RESPONSE_AS_Q_DEF olarak ayarlandığında uygulamalar tarafından kullanılan deđer tanımlar. Bu öznelik tüm kuyruk tipleri için geçerlidir.

Çizelge 585. Bu özneliğin geçerli olduđu kuyruk tipleri				
Yerel	Modeli	Diđer Ad	Uzak	Küme
X	X	X	X	X

Deđer ařağıdakilerden biridir:

Sync

Koyma işlemi zamanuyumlu olarak bir yanıt döndürerek yayınlanır.

Async

Koyma işlemi zamanuyumsuz olarak yayınlanır ve MQMD alanlarının bir alt kümesi döndürülüyor.

Bu özneliğin deđerini saptamak için MQINQ çağrısıyla MQIA_DEF_PUT_RESPONSE_TYPE seçicisini kullanın.

DistLists (MQLONG)

Bu, dağıtım listesi iletilerinin kuyruğa yerleştirilip yerleştirilemeyeceğini gösterir.

Çizelge 586. Bu özneliğin geçerli olduđu kuyruk tipleri				
Yerel	Modeli	Diđer Ad	Uzak	Küme
X	X			

Bir ileti kanalı aracısı (MCA), yerel kuyruk yöneticisine, kanalın diđer ucundaki kuyruk yöneticisinin dağıtım listelerini destekleyip desteklemediğini bildirmek için özneliğı ayarlar. Bu ikinci kuyruk yöneticisi (ortaklık kuyruk yöneticisi olarak adlandırılır), iletiyi gönderen bir MCA tarafından yerel iletim kuyruğundan kaldırıldıktan sonra iletiyi alacak olan kuyruk yöneticisidir.

Gönderen MCA, ortak kuyruk yöneticisinde alan MCA ile bağlantı kurduğunda özneliğı ayarlar. Bu şekilde, gönderen MCA, yerel kuyruk yöneticisinin yalnızca ortak kuyruk yöneticisinin doğru olarak işleyebileceğı iletileri iletim kuyruğuna yerleştirmesine neden olabilir.

Bu öznelik birincil olarak iletim kuyruklarıyla kullanılır, ancak tanımlanan işlem, kuyruk için tanımlanan kullanımdan bağımsız olarak gerçekleştirilir (bkz. Kullanım özneliğı).

Değer aşağıdakilerden biridir:

MQDL_SUPPORTED

Dağıtım listesi iletileri kuyrukta saklanabilir ve bu formda ortak kuyruk yöneticisine iletilebilir. Bu, iletiyi birden çok hedefe göndermek için gereken işleme miktarını azaltır.

MQDL_NOT_DESTEKLENMİYOR

Dağıtım listesi iletileri, ortak kuyruk yöneticisi dağıtım listelerini desteklemediği için kuyrukta saklanamıyor. Bir uygulama bir dağıtım listesi iletiyi koyarsa ve bu ileti bu kuyruğa yerleştirilirse, kuyruk yöneticisi dağıtım listesi iletiyi böler ve tek tek iletileri kuyruğa yerleştirir. Bu, iletiyi birden çok hedefe göndermek için gerekli işleme miktarını artırır, ancak iletilerin ortak kuyruk yöneticisi tarafından doğru şekilde işlenmesini sağlar.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQIA_DIST_LIST seçicisini kullanın. Bu özneliğin değerini değiştirmek için MQSET çağrısına bakın.

Bu öznelik z/OS üzerinde desteklenmez.

HardenGetGeri Alma (MQLONG)

Her ileti için, bir iş birimi içindeki bir MQGET çağrısıyla iletinin kaç kez alındığını ve o iş biriminin kaç kez geriletildiğini belirten bir sayı tutulur.

<i>Çizelge 587. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu sayı, MQGET çağrısı tamamlandıktan sonra ileti tanımlayıcısındaki *BackoutCount* alanında kullanılabilir.

İleti geriletme sayısı, kuyruk yöneticisini yeniden başlatmaya devam eder. Ancak, sayımın doğru olduğundan emin olmak için, MQGET çağrısı bu kuyruğa ilişkin bir iş birimi içindeki bir iletiyi her aldığı anda bilgilerin *sertleştirilmiş* olması (diske ya da diğer bir kalıcı depolama aygıtına kaydedilmesi) gerekir. Bu yapılmazsa, kuyruk yöneticisi başarısız olur ve MQGET çağrısı geri çekilirse, sayı artırılabilir ya da artırılamaz.

Bir iş birimindeki her MQGET çağrısına ilişkin bilgilerin güçlendirilmesi ek işleme maliyeti getirir; bu nedenle, **HardenGetBackout** özneliğini MQQA_BACKOUT_HARDENED olarak ayarlayın; ancak, saymanın doğru olması çok önemliyse.

Çoklu platformlar üzerinde, bu özneliğin ayarından bağımsız olarak, ileti geriletme sayısı her zaman sertleşmiştir.

Aşağıdaki değerler kullanılabilir:

MQQA_BACKOUT_HARDENED

Bu kuyruktaki iletilere ilişkin geriletme sayısının doğru olduğundan emin olmak için sertleme kullanılır.

MQQA_BACKOUT_NOT_SERTLEŞTİRİLMİŞ

Bu kuyruktaki iletilere ilişkin geriletme sayısının doğru olduğundan emin olmak için sertleme kullanılmaz. Bu nedenle sayım olması gerekenden daha düşük olabilir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_HARDEN_GET_BACKOUT seçicisini kullanın.

IndexType (MQLONG)

Kuyruk yöneticisinin kuyruktaki iletiler için tutmakta olduğu dizin tipini belirtir.

<i>Çizelge 588. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Gereken izin tipi, uygulamanın iletileri nasıl aldığına ve kuyruğun paylaşılan bir kuyruk mu, yoksa paylaşılmayan bir kuyruk mu olduğuna bağlıdır (bkz. [QSGDisp özneliği](#)). *IndexType* için aşağıdaki değerler kullanılabilir:

MQIT_NONE

Bu kuyruk için kuyruk yöneticisi herhangi bir dizini tutmaz. MQGET çağrısında seçim ölçütü kullanılmadan, genellikle sıralı olarak işlenen kuyruklar için bu değeri kullanın.

MQIT_MSG_ID

Kuyruk yöneticisi, kuyruktaki iletilerin ileti tanıtıcılarını kullanan bir izin sağlar. Uygulamanın genellikle ileti tanıtıcısını MQGET çağrısında seçim ölçütü olarak kullanarak iletileri aldığı bu değer kuyruklarını kullanın.

MQIT_CORREL_ID

Kuyruk yöneticisi, kuyruktaki iletilerin ilinti tanıtıcılarını kullanan bir izin sağlar. Bu değeri, uygulamanın genellikle MQGET çağrısında seçim ölçütü olarak ilinti tanıtıcısını kullanan iletileri aldığı kuyruklar için kullanın.

MQIT_MSG_TOKEN

Önemli: Bu izin tipi yalnızca, IBM MQ Workflow for z/OS ürünüyle kullanılan kuyruklar için kullanılmalıdır.

Kuyruk yöneticisi, z/OS iş yükü yöneticisi (WLM) işlemleriyle kullanılmak üzere kuyruktaki iletilerin ileti belirteçlerini kullanan bir izin sağlar.

WLM tarafından yönetilen kuyruklar için bu seçeneği belirtmeniz *gerekir* ; bunu başka bir kuyruk tipi için belirtmeyin. Ayrıca, bir uygulamanın z/OS iş yükü yöneticisi işlemlerini kullanmadığı, ancak ileti simgesini MQGET çağrısında seçim ölçütü olarak kullanan iletileri almakta olduğu bir kuyruk için de bu değeri kullanmayın.

MQIT_GRP_Tnt

Kuyruk yöneticisi, kuyruktaki iletilerin grup tanıtıcılarını kullanan bir izin sağlar. Bu değer, uygulamanın MQGET çağrısında MQGMO_LOGICAL_ORDER seçeneğini kullanarak ileti aldığı kuyruklar için kullanılmalıdır.

Bu izin tipine sahip bir kuyruk, bir iletim kuyruğu olamaz. CFLEVEL (3) ya da üstü bir CFSTRUCT nesnesiyle eşlemek için bu izin tipine sahip paylaşılan bir kuyruk tanımlanmalıdır.

Not:

1. Dizin tipi MQIT_GROUP_ID olan bir kuyruktaki iletilerin fiziksel sırası tanımlanmadı; kuyruk, MQGET çağrısında MQGMO_LOGICAL_ORDER seçeneğini kullanarak iletilerin verimli bir şekilde alınması için eniyilendi. Bu, iletilerin fiziksel sırasının genellikle iletilerin kuyruğa geldiği sıra olmadığı anlamına gelir.
2. Bir MQIT_GROUP_ID kuyruğunun *MsgDeliverySequence* değeri MQMDS_PRIORITY ise, kuyruk yöneticisi iletilerin mantıksal sırayla alınmasını eniyilemek için 0 ve 1 ileti önceliklerini kullanır. Sonuç olarak, bir gruptaki ilk iletinin önceliği sıfır ya da bir olmamalıdır; varsa, ileti iki önceliğe sahip gibi işlenir. MQMD yapısındaki *Priority* alanı değiştirilmez.

İleti gruplarıyla ilgili daha fazla bilgi için, [MQGMO-Options alanındaki](#) grup ve bölüm seçeneklerinin tanımına bakın.

Çeşitli durumlarda kullanılması gereken izin tipi [Çizelge 589 sayfa 841](#) ve [Çizelge 590 sayfa 842](#) içinde gösterilir.

<i>Çizelge 589. MQGMO_LOGICAL_ORDER belirtilmediğinde kuyruk dizini tipinin önerilen ya da gerekli değerleri</i>		
MQGET çağrısında seçim ölçütleri	Paylaşılmayan kuyruk için izin tipi	Paylaşılan kuyruk için izin tipi
Yok	Herhangi Biri	Herhangi Biri
Tek bir tanıtıcı kullanılarak seçim:		

Çizelge 589. MQGMO_LOGICAL_ORDER belirtilmediğinde kuyruk dizini tipinin önerilen ya da gerekli değerleri (devamı var)

MQGET çağrısında seçim ölçütleri	Paylaşılmayan kuyruk için dizin tipi	Paylaşılan kuyruk için dizin tipi
İleti Tanıtıcısı	MQIT_MSG_ID önerilir	MQIT_NONE ya da MQIT_MSG_ID gereklidir; MQIT_MSG_ID önerilir
İlinti tanıtıcısı	MQIT_CORREL_ID önerilir	MQIT_CORREL_ID gereklidir
Grup tanıtıcısı	MQIT_GROUP_ID önerilir	MQIT_GROUP_ID gerekli
İki tanıtıcı kullanan seçim:		
İleti tanıtıcısı artı ilinti tanıtıcısı	MQIT_MSG_ID ya da MQIT_CORREL_ID önerilir	MQIT_NONE ya da MQIT_MSG_ID ya da MQIT_CORREL_ID gereklidir (Verimlilik için, dizin tipinin en farklı anahtarlara sahip MQMD alanıyla eşleşecek şekilde seçilmesi önerilir)
İleti tanıtıcısı artı grup tanıtıcısı	MQIT_MSG_ID ya da MQIT_GROUP_ID önerilir	Desteklenmiyor
İlinti tanıtıcısı artı grup tanıtıcısı	MQIT_CORREL_ID ya da MQIT_GROUP_ID önerilir	Desteklenmiyor
Üç tanıtıcı kullanarak seçim:		
İleti tanıtıcısı artı ilinti tanıtıcısı artı grup tanıtıcısı	MQIT_MSG_ID ya da MQIT_CORREL_ID ya da MQIT_GROUP_ID önerilir	Desteklenmiyor
Grupla ilgili ölçütler kullanılarak seçim:		
Grup tanıtıcısı ve ileti sıra numarası	MQIT_GROUP_ID gerekli	MQIT_GROUP_ID gerekli
İleti sıra numarası (1 olmalı)	MQIT_GROUP_ID gerekli	MQIT_GROUP_ID gerekli
İleti simgesi kullanılarak seçim:		
Uygulama kullanımı için ileti simgesi	MQIT_MSG_TOKEN kullanma	
WLM kullanımı için ileti simgesi	MQIT_MSG_TOKEN gerekli	Desteklenmiyor


Çizelge 590. MQGMO_LOGICAL_ORDER belirtildiğinde kuyruk dizini tipinin önerilen ya da gerekli değerleri

MQGET çağrısında seçim ölçütleri	Paylaşılmayan kuyruk için dizin tipi	Paylaşılan kuyruk için dizin tipi
Yok	MQIT_GROUP_ID gerekli	MQIT_GROUP_ID gerekli
Tek bir tanıtıcı kullanılarak seçim:		
İleti Tanıtıcısı	MQIT_GROUP_ID gerekli	Desteklenmiyor
İlinti tanıtıcısı	MQIT_GROUP_ID gerekli	Desteklenmiyor

Çizelge 590. MQGMO_LOGICAL_ORDER belirtildiğinde kuyruk dizini tipinin önerilen ya da gerekli değerleri (devamı var)

MQGET çağrısında seçim ölçütleri	Paylaşılmayan kuyruk için dizin tipi	Paylaşılan kuyruk için dizin tipi
Grup tanıtıcısı	MQIT_GROUP_ID gerekli	MQIT_GROUP_ID gerekli
İki tanıtıcı kullanan seçim:		
İleti tanıtıcısı artı ilinti tanıtıcısı	MQIT_GROUP_ID gerekli	Desteklenmiyor
İleti tanıtıcısı artı grup tanıtıcısı	MQIT_GROUP_ID gerekli	Desteklenmiyor
İlinti tanıtıcısı artı grup tanıtıcısı	MQIT_GROUP_ID gerekli	Desteklenmiyor
Üç tanıtıcı kullanarak seçim:		
İleti tanıtıcısı artı ilinti tanıtıcısı artı grup tanıtıcısı	MQIT_GROUP_ID gerekli	Desteklenmiyor

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_INDEX_TYPE seçicisini kullanın.

 Bu öznitelik yalnızca z/OS üzerinde desteklenir.

InhibitGet (MQLONG)

Bu, bu kuyruk için alma işlemlerine izin verilip verilmeyeceğini denetler.

Çizelge 591. Bu özniteliğin geçerli olduğu kuyruk tipleri

Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X		

Kuyruk bir diğer ad kuyruğuysa, MQGET çağrısının başarılı olması için, alma işlemi sırasında hem diğer ad hem de temel kuyruk için alma işlemlerine izin verilmelidir. Değer aşağıdakilerden biridir:

MQQA_GET_INENGELLEME

Alma işlemleri engellenmiş.

MQGET çağrıları başarısız oldu, neden kodu MQRC_GET_INDÜ. Bu, MQGMO_BROWSE_FIRST ya da MQGMO_BROWSE_NEXT belirten MQGET çağrılarını içerir.

Not: Bir iş birimi içinde çalışan bir MQGET çağrısı başarıyla tamamlanırsa, **InhibitGet** özniteliğinin değerinin daha sonra MQQA_GET_INENGELLENDI olarak değiştirilmesi, iş biriminin kesinleştirilmesini engellemez.

MQQA_GET_ALLOWED

Alma işlemlerine izin verilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_INHIBIT_GET seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

InhibitPut (MQLONG)

Bu, bu kuyruk için koyma işlemlerine izin verilip verilmeyeceğini denetler.

Çizelge 592. Bu özniteliğin geçerli olduğu kuyruk tipleri

Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Kuyruk adı çözüme yolunda birden çok tanımlama varsa, koyma işlemi sırasında yoldaki *her* tanımlaması (kuyruk yöneticisi diğer adı tanımlamaları da içinde olmak üzere) için, MQPUT ya da MQPUT1 çağrısının başarılı olması için koyma işlemlerine izin verilmelidir. Değer aşağıdakilerden biridir:

MQQA_PUT_INENGELLEME

Koyma işlemleri engellenmiş.

MQPUT ve MQPUT1 çağrıları başarısız oldu; neden kodu MQRC_PUT_INN.

Not: Bir iş birimi içinde çalışan bir MQPUT çağrısı başarıyla tamamlanırsa, **InhibitPut** özniteliğinin değerinin daha sonra MQQA_PUT_INN olarak değiştirilmesi, iş biriminin kesinleştirilmesini engellemez.

MQQA_PUT_ALLOWED

Koyma işlemlerine izin verilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_INHIBIT_PUT seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

InitiationQName (MQCHAR48)

Bu, yerel kuyruk yöneticisinde tanımlı bir kuyruğun adıdır; kuyruk MQQT_LOCAL tipinde olmalıdır.

Çizelge 593. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Kuyruk yöneticisi, bu özniteliğin ait olduğu kuyruğa gelen bir iletinin sonucu olarak uygulama başlatma gerektiğinde başlatma kuyruğuna bir tetikleyici iletisi gönderir. Başlatma kuyruğu, tetikleme iletisi alındıktan sonra uygun uygulamayı başlatan bir tetikleyici izleme uygulaması tarafından izlenmelidir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQCA_INITIATION_Q_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

MaxMsgUzunluğu (MQLONG)

Bu, kuyruğa yerleştirilebilecek en uzun *fiziksel* iletinin uzunluğu için bir üst sınırdır.

Çizelge 594. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Ancak, **MaxMsgLength** kuyruk özniteliği **MaxMsgLength** kuyruk yöneticisi özniteliğinden bağımsız olarak ayarlanabildiğinden, kuyruğa yerleştirilebilecek en uzun fiziksel iletinin uzunluğu için gerçek üst sınır, bu iki değerden daha azdır.

Kuyruk yöneticisi bölümlenmeyi destekliyorsa, bir uygulamanın iki **MaxMsgLength** özniteliğinin daha küçük olanından daha uzun bir *mantıksal* ileti koyması mümkündür; ancak, uygulama MQMD 'de MQMF_SEGMENTATION_ALLOWED işaretini belirtiyorsa. Bu işaret belirtilirse, mantıksal ileti uzunluğu üst sınırı 999 999 999 bayttır, ancak genellikle işletim sistemi ya da uygulamanın çalıştığı ortam tarafından zorunlu kılınan kaynak kısıtlamaları alt sınırla sonuçlanır.

Çok uzun bir iletiyi kuyruğa yerleştirme girişimi, aşağıdaki neden kodlarından biriyle başarısız olur:

- İleti kuyruk için çok büyükse, MQRC_MSG_TOO_BIG_FOR_Q
- İleti kuyruk yöneticisi için çok büyükse, ancak kuyruk için çok büyük değilse, MQRC_MSG_TOO_BIG_FOR_Q_MGR

MaxMsgLength özniteliğinin alt sınırı sıfırdır; üst sınır 100 MB 'dir (104 857 600 bayt).

Ek bilgi için [MQPUT- BufferLength değiştirilmesine](#) bakın.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_MAX_MSG_LENGTH seçicisini kullanın.

MaxQDepth (MQLONG)

Bu, herhangi bir zamanda kuyrukta bulunabilecek fiziksel ileti sayısı için tanımlanan üst sınırdır.

Çizelge 595. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

MaxQDepth ileti içeren bir kuyruğa ileti koyma girişimi başarısız oldu. Neden kodu: MQRC_Q_FULL.

İş birimi işleme ve iletilerin bölümlenmesi, kuyruktaki gerçek fiziksel ileti sayısının **MaxQDepth** değerini aşmasına neden olabilir. Ancak, kuyruktaki tüm iletiler MQGET çağrısı kullanılarak alınabildiğinden, bu durum iletinin alınabilirliğini etkilemez.

Bu özneliğin değeri sıfır ya da daha büyük. Üst sınır ortam tarafından belirlenir:

- Aşağıdaki altyapılarda, değer 999 999 999 'u aşamaz:

- ▶ **AIX** AIX
- ▶ **Linux** Linux
- ▶ **Windows** Windows
- ▶ **z/OS** z/OS

- ▶ **IBM i** IBM i' de değer 640 000 'i aşamaz.

Not: Kuyrukta **MaxQDepth** iletiden az ileti olsa bile, kuyruğun kullanabileceği saklama alanı tükenebilir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_MAX_Q_DEPTH seçicisini kullanın.

MsgDeliverySırası (MQLONG)

Çizelge 596. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, MQGET çağrısına ilişkin iletilerin uygulamaya döndürüleceği sırayı belirler:

MQMDS_FIFO

İletiler FIFO sırasıyla döndürülür (ilk giren ilk çıkar).

MQGET çağrısı, iletinin önceliğine bakılmaksızın, çağrıda belirtilen seçim ölçütlerini karşılayan *ilk* iletiyi döndürür.

MQMDS_PRIORITY

İletiler öncelik sırasına göre döndürülür.

Bir MQGET çağrısı, çağrıda belirtilen seçim ölçütlerini karşılayan *en yüksek öncelik* iletisini döndürür. Her öncelik düzeyinde, iletiler FIFO sırasıyla döndürülür (ilk giren ilk çıkar).

- z/OS sistemlerinde, kuyrukta bir *IndexType* (MQIT_GROUP_ID) varsa, **MsgDeliverySequence** özneliği, ileti gruplarının uygulamaya döndürüleceği sırayı belirtir. Grupların döndürüleceği sıra, her gruptaki ilk iletinin konumuna ya da önceliğine göre belirlenir. Kuyruktaki iletilerin fiziksel sırası tanımlanmadı; kuyruk, MQGET çağrısında MQGMO_LOGICAL_ORDER seçeneğini kullanarak iletilerin verimli bir şekilde alınması için eniyilendi.
- z/OS sistemlerinde *IndexType* MQIT_GROUP_ID ve *MsgDeliverySequence* MQMDS_PRIORITY ise, kuyruk yöneticisi, iletilerin mantıksal sırayla alınmasını eniyilemek için sıfır ve bir ileti önceliklerini kullanır. Sonuç olarak, bir gruptaki ilk iletinin önceliği sıfır ya da bir olmamalıdır; varsa, ileti iki önceliğe sahip gibi işlenir. MQMD yapısındaki *Priority* alanı değiştirilmez.

Kuyrukta iletiler varken ilgili öznelikler değiştirilirse, teslim sırası aşağıdaki gibidir:

- İletilerin MQGET çağrısı tarafından döndürülme sırası, ileti kuyruğa geldiğinde kuyruk için yürürlükte olan **MsgDeliverySequence** ve **DefPriority** özniteliklerinin değerlerine göre belirlenir:
 - İleti geldiğinde *MsgDeliverySequence* MQMDS_FIFO ise, ileti, önceliği *DefPriority* gibi kuyruğa yerleştirilir. Bu, iletinin ileti tanımlayıcısındaki *Priority* alanının değerini etkilemez; bu alan, ileti ilk konduğunda sahip olduğu değeri korur.
 - İleti geldiğinde *MsgDeliverySequence* MQMDS_PRIORITY ise, ileti, ileti tanımlayıcısında *Priority* alanı tarafından belirtilen önceliğe uygun yerde kuyruğa yerleştirilir.

Kuyruktaki iletiler varken **MsgDeliverySequence** özniteliğinin değeri değiştirilirse, kuyruktaki iletilerin sırası değişmez.

Kuyrukta iletiler varken **DefPriority** özniteliğinin değeri değiştirilirse, **MsgDeliverySequence** özniteliği MQMDS_FIFO; kuyruğa daha yüksek öncelikte konanlar önce teslim edilse de, iletilerin FIFO sırasıyla teslim edilmesi gerekmez.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_MSG_DELIVERY_SEQUENCE seçicisini kullanın.

NonPersistentMessageClass (MQLONG)

Kalıcı olmayan iletiler için güvenilirlik hedefi.

Çizelge 597. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, bu kuyruğa konan kalıcı olmayan iletilerin atıldığı durumları belirtir:

MQNPM_CLASS_NORMAL

Kalıcı olmayan iletiler, kuyruk yöneticisi oturumunun geçerlik süreleriyle sınırlıdır; iletiler, kuyruk yöneticisinin yeniden başlatılması durumunda atılır. Bu, yalnızca paylaşılmayan kuyruklar için geçerlidir ve varsayılan değerdir.

MQNPM_CLASS_HIGH

Kuyruk yöneticisi, kuyruk ömrü boyunca kalıcı olmayan iletileri korumayı dener. Bir hata durumunda kalıcı olmayan iletiler kaybolmaya devam edebilir. Bu değer, paylaşılan kuyruklar için uygulanır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_NPM_CLASS seçicisini kullanın.

OpenInputSayısı (MQLONG)

Bu, MQGET çağrısıyla kuyruktan ileti kaldırmak için geçerli olan tanıtıcı sayısıdır.

Çizelge 598. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Bu, *yerel* kuyruk yöneticisi tarafından bilinen bu tür tanıtıcıların toplam sayısıdır. Kuyruk paylaşılan bir kuyruksa, bu sayı, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubundaki diğer kuyruk yöneticilerindeki kuyruk için gerçekleştirilen giriş açılarını içermez.

Sayı, bu kuyruğa çözülen diğer ad kuyruğunun giriş için açıldığı tanıtıcıları içerir. Sayı, giriş içermeyen işlemler (örneğin, yalnızca göz atmak için açılan bir kuyruk) için kuyruğun açıldığı tanıtıcıları içermez.

Kuyruk yöneticisi çalışırken bu özniteliğin değeri dalgalandı.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_OPEN_INPUT_COUNT seçicisini kullanın.

OpenOutputSayısı (MQLONG)

Bu, MQPUT çağrısıyla kuyruğa ileti eklemek için geçerli olan tanıtıcı sayısıdır.

<i>Çizelge 599. Bu özneliğin geçerli olduđu kuyruk tipleri</i>				
Yerel	Modeli	Diđer Ad	Uzak	Küme
X				

Bu, yerel kuyruk yöneticisi tarafından bilinen bu tür tanıtıcıların toplam sayısıdır; uzak kuyruk yöneticisinde bu kuyruk için gerçekleştirilen çıkışlara ilişkin açma işlemlerini içermez. Kuyruk paylaşılan bir kuyruksa, bu sayı, yerel kuyruk yöneticisinin ait olduđu kuyruk paylaşım grubundaki diđer kuyruk yöneticilerindeki kuyruk için gerçekleştirilen çıkışa ilişkin açmaları içermez.

Sayı, bu kuyruğa çözülen diđer ad kuyruğunun çıkış için açıldıđı tanıtıcıları içerir. Sayı, çıkış içermeyen işlemler (örneğin, yalnızca sorgu için açılan bir kuyruk) için kuyruğun açıldıđı tanıtıcıları içermez.

Kuyruk yöneticisi çalışırken bu özneliğin değeri dalgalandı.

Bu özneliğin değeri saptamak için MQINQ çağrısıyla MQIA_OPEN_OUTPUT_COUNT seçicisini kullanın.

ProcessName (MQCHAR48)

Bu, yerel kuyruk yöneticisinde tanımlanan bir süreç nesnesinin adıdır. Süreç nesnesi, kuyruğa hizmet verebilen bir programı tanıtır.

<i>Çizelge 600. Bu özneliğin geçerli olduđu kuyruk tipleri</i>				
Yerel	Modeli	Diđer Ad	Uzak	Küme
X	X			

Bu özneliğin değeri saptamak için MQINQ çağrısıyla MQCA_PROCESS_NAME seçicisini kullanın. Bu özneliğin uzunluđu MQ_PROCESS_NAME_LENGTH tarafından verilir.

PropertyControl (MQLONG)

MQGMO_PROPERTIES_AS_Q_DEF seçeneđiyle MQGET çağrısı kullanılarak kuyruklardan alınan iletiler için ileti özelliklerinin nasıl işleneceđini belirtir.

<i>Çizelge 601. Bu özneliğin geçerli olduđu kuyruk tipleri</i>				
Yerel	Modeli	Diđer Ad	Uzak	Küme
X	X	X		

Deđer aşağıdakilerden biridir:

MQPROP_Tümü

İletin tüm özellikleri, uygulamaya teslim edildiğinde iletiye eklenir. İleti tanımlayıcısı (ya da uzantısı) dışındaki özellikler, ileti verilerindeki bir ya da daha çok MQRFH2 üstbilgisine yerleştirilir. Bir ileti tanıtıcısı sağlanırsa, davranış, ileti tanıtıcısındaki özellikleri döndürmektir.

MQPROP_COMPATIBILITY

İleti, mcd. öneğine sahip bir özellik içeriyorsa, Jm., -USR. ya da mqext., Tüm ileti özellikleri bir MQRFH2 üstbilgisinde uygulamaya teslim edilir. Ters durumda, ileti tanımlayıcısında (ya da uzantıda) bulunanlar dışında, iletin tüm özellikleri atılır ve artık uygulama bunlara erişemez. Bu varsayılan değerdur; JMS ile ilgili özelliklerin, ileti verilerindeki bir MQRFH2 üstbilgisinde olmasını beklenen uygulamaların deđiştirilmeden çalışmaya devam etmesini sağlar. Bir ileti tanıtıcısı sağlanırsa, davranış ileti tanıtıcısındaki özellikleri döndürmektir.

MQPROP_FORCE_MQRFH2

Uygulamanın bir ileti tanıtıcısı belirtip belirtmediđine bakılmaksızın, MQRFH2 üstbilgisindeki ileti verilerinde her zaman özellikler döndürülür. MQGET çağrısındaki MQGMO yapısının MsgHandle alanında sağlanan geçerli bir ileti tanıtıcısı yoksa, İletin özelliklerine ileti tanıtıcısıyla erişilemez.

MQPROP_NONE

İleti tanımlayıcısı (ya da uzantısı) dışında, iletinin tüm özellikleri, ileti uygulamaya teslim edilmeden önce iletiden kaldırılır. Bir ileti tanıtıcısı sağlanırsa, davranış, ileti tanıtıcısındaki özellikleri döndürmektir.

Bu parametre Yerel, Diğer Ad ve Model kuyrukları için geçerlidir. Değerini saptamak için MQINQ çağrısıyla MQIA_PROPERTY_CONTROL seçicisini kullanın.

QDepthHighOlayı (MQLONG)

Bu, Kuyruk Derinliği Yüksek olaylarının oluşturulup oluşturulmayacağını denetler.

<i>Çizelge 602. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Kuyruk Derinliği Yüksek olayı, bir uygulamanın kuyruğa bir ileti koyduğunu ve bu, kuyruktaki iletilerin sayısının kuyruk derinliği yüksek eşikinden büyük ya da ona eşit olmasına neden olduğunu gösterir (bkz. **QDepthHighLimit** özneliği).

Not: Bu özneliğin değeri dinamik olarak değişebilir.

Değer aşağıdakilerden biridir:

MQEVR_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVR_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_Q_DEPTH_HIGH_EVENT seçicisini kullanın.

Bu öznelik z/OS üzerinde desteklenir, ancak değerini saptamak için MQINQ çağrısı kullanılamaz.

QDepthHighSınırı (MQLONG)

Bu, bir Kuyruk Derinliği Yüksek olayı oluşturmak için kuyruk derinliğinin karşılaştırıldığı eşiktir.

<i>Çizelge 603. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu olay, bir uygulamanın kuyruğa bir ileti koyduğunu ve kuyruktaki ileti sayısının kuyruk derinliği yüksek eşikinden büyük ya da ona eşit olmasına neden olduğunu gösterir. Bkz. [QDepthHighOlay](#) özneliği.

Değer, kuyruk derinliği üst sınırı (**MaxQDepth** özneliği) yüzdesi olarak ifade edilir ve 0 'dan büyük ya da 0 'a eşittir ve 100 'den küçük ya da 100 'e eşittir. Varsayılan değer 80 'dir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_Q_DEPTH_HIGH_LIMIT seçicisini kullanın.

Bu öznelik z/OS üzerinde desteklenir, ancak değerini saptamak için MQINQ çağrısı kullanılamaz.

QDepthLowOlayı (MQLONG)

Bu, Kuyruk Derinliği Düşük olaylarının oluşturulup oluşturulmayacağını denetler.

<i>Çizelge 604. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Kuyruk Derinliği Düşük olayı, bir uygulamanın kuyruktan bir ileti aldığını ve kuyruktaki ileti sayısının kuyruk derinliği alt eşliğinden az ya da ona eşit olmasına neden olduğunu gösterir (bkz. [QDepthLowLimit](#) özneliği).

Not: Bu özneliğin değeri dinamik olarak değışebilir.

Değer aşağıdakilerden biridir:

MQEVN_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVN_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin değeri saptamak için MQINQ çağrısıyla MQIA_Q_DEPTH_LOW_EVENT seçicisini kullanın.

Bu öznelik z/OSüzerinde desteklenir, ancak değeri saptamak için MQINQ çağrısı kullanılamaz.

QDepthLowSınırı (MQLONG)

Bu, bir Kuyruk Derinliği Düşük olayı oluşturmak için kuyruk derinliğinin karşılaştırıldığı eşiktir.

<i>Çizelge 605. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu olay, bir uygulamanın kuyruktan bir ileti aldığını ve kuyruktaki ileti sayısının kuyruk derinliği alt eşliğinden az ya da ona eşit olmasına neden olduğunu gösterir. Bkz. [QDepthLowOlay](#) özneliği.

Değer, kuyruk derinliği üst sınırı (**MaxQDepth** özneliği) yüzdesi olarak ifade edilir ve 0 'dan büyük ya da 0 'a eşittir ve 100 'den küçük ya da 100 'e eşittir. Varsayılan değeri 20'dir.

Bu özneliğin değeri saptamak için MQINQ çağrısıyla MQIA_Q_DEPTH_LOW_LIMIT seçicisini kullanın.

Bu öznelik z/OSüzerinde desteklenir, ancak değeri saptamak için MQINQ çağrısı kullanılamaz.

QDepthMaxOlayı (MQLONG)

Bu, Kuyruk Tam olaylarının oluşturulup oluşturulmayacağını denetler. Tam Kuyruk olayı, kuyruk dolu olduğu için bir kuyruğa koyma işleminin reddedildiğini, yani kuyruk derinliğinin zaten üst sınır değerine ulaştığını gösterir.

<i>Çizelge 606. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Not: Bu özneliğin değeri dinamik olarak değışebilir.

Değer aşağıdakilerden biridir:

MQEVN_DISABLED

Olay raporlama devre dışı bırakıldı.

MQEVN_ENABLED

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin değeri saptamak için, MQINQ çağrısıyla MQIA_Q_DEPTH_MAX_EVENT seçicisini kullanın.

Bu öznelik z/OSüzerinde desteklenir, ancak değeri saptamak için MQINQ çağrısı kullanılamaz.

QDesc (MQCHAR64)

Açıklayıcı açıklama için bu alanı kullanın.

Çizelge 607. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Alanın içeriği kuyruk yöneticisi için önemli değil, ancak kuyruk yöneticisi alanın yalnızca görüntülenebilecek karakterleri içermesini gerektirebilir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 bayttır).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özneliğinde tanımlandığı gibi) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_Q_DESC seçicisini kullanın. Bu özneliğin uzunluğu MQ_Q_DESC_LENGTH tarafından verilir.

QName (MQCHAR48)

Bu, yerel kuyruk yöneticisinde tanımlı bir kuyruğun adıdır.

Çizelge 608. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Bir kuyruk yöneticisinde tanımlanan tüm kuyruklar aynı kuyruk ad alanını paylaşır. Bu nedenle, bir MQQT_LOCAL kuyruğu ve bir MQQT_ALIAS kuyruğu aynı ada sahip olamaz.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_Q_NAME seçicisini kullanın. Bu özneliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

QServiceInterval (MQLONG)

Bu, Hizmet Aralığı Yüksek ve Hizmet Aralığı Tamam olaylarını oluşturmak için kullanılan hizmet aralığıdır.

Çizelge 609. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bkz. [QServiceIntervalOlay özneliği](#).

Değer milisaniye cinsinden, sıfırdan büyük ya da sıfıra eşit ve 999 999 999 'dan küçük ya da bu değere eşit.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_Q_SERVICE_INTERVAL seçicisini kullanın.

Bu öznelik z/OS üzerinde desteklenir, ancak değerini saptamak için MQINQ çağrısı kullanılamaz.

QServiceIntervalOlayı (MQLONG)

Bu, Hizmet Aralığı Yüksek ya da Hizmet Aralığı Tamam olaylarının oluşturulup oluşturulmayacağını denetler.

Çizelge 610. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

- Bir denetim, kuyruktan en az **QServiceInterval** özniteliğiyle belirtilen süre boyunca herhangi bir ileti alınmadığını belirttiğinde bir Hizmet Aralığı Yüksek Olayı oluşturulur.
- Bir denetim, iletilerin **QServiceInterval** özniteliğiyle belirtilen süre içinde kuyruktan alındığını gösterdiğinde bir Hizmet Aralığı Tamam olayı oluşturulur.

Not: Bu özniteliğin değeri dinamik olarak değişebilir.

Değer aşağıdakilerden biridir:

MQQSIE_HIGH

Kuyruk Hizmeti Aralığı Yüksek olayları etkinleştirildi.

- Kuyruk Hizmeti Aralığı Yüksek olayları **etkin** ve
- Kuyruk Hizmeti Aralığı Tamam olayları **devre dışı**.

MQQSIE_OK

Kuyruk Hizmeti Aralığı Tamam olayları etkinleştirildi.

- Kuyruk Hizmeti Aralığı Yüksek olayları **devre dışı** ve
- Kuyruk Hizmeti Aralığı Tamam olayları **etkin**.

MQQSIE_NONE

Herhangi bir kuyruk hizmeti aralığı olayı etkinleştirilmedi.

- Kuyruk Hizmeti Aralığı Yüksek olayları **devre dışı** ve
- Kuyruk Hizmeti Aralığı Tamam olayları da **devre dışı** bırakılır.

Paylaşılan kuyruklar için bu özniteliğin değeri yoksayılr; MQQSIE_NONE değeri varsayılr.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_Q_SERVICE_INTERVAL_EVENT seçicisini kullanın.

z/OSüzerinde, bu özniteliğin değerini saptamak için MQINQ çağrısı kullanamazsınız.

QSGDisp (MQLONG)

Bu, kuyruğun atılacağını belirtir.

<i>Çizelge 611. Bu özniteliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Değer aşağıdakilerden biridir:

MQQSGD_Q_MGR

Nesnenin kuyruk yöneticisi yok etme işlemi var. Bu, nesne tanımlamasının yalnızca yerel kuyruk yöneticisi tarafından tanındığı; tanımın kuyruk paylaşım grubundaki diğer kuyruk yöneticileri tarafından bilinmediği anlamına gelir.

Kuyruk paylaşım grubundaki her kuyruk yöneticisinin yürürlükteki nesneyle aynı ada ve tipe sahip bir nesnesi olabilir; ancak, bunlar ayrı nesnelere ve bunlar arasında ilinti yoktur. Öznitelikleri birbiriyle aynı olacak şekilde kısıtlanmaz.

MQQSGD_COPY

Nesne, paylaşılan havuzda var olan bir ana nesne tanımlamasının yerel bir kopyasıdır. Kuyruk paylaşım grubundaki her kuyruk yöneticisinin nesnenin kendi kopyası olabilir. Başlangıçta, tüm kopyaların öznitelikleri aynıdır, ancak MQSC komutlarını kullanarak, her kopyanın özniteliklerini diğer kopyalardan farklı olacak şekilde değiştirebilirsiniz. Paylaşılan havuzdaki ana tanımlama değiştirildiğinde, kopyaların öznitelikleri yeniden eşitlenir.

MQOSGD_SHARED

Nesne paylaşılan yok etme içeriyor. Bu, paylaşılan havuzda, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından bilinen nesnenin tek bir eşgörünümünün bulunduğu anlamına gelir. Gruptaki bir kuyruk yöneticisi nesneye eriştiğinde, nesnenin paylaşılan tek eşgörünümüne erişir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_QSG_DISP seçicisini kullanın.

z/OS Bu öznelik yalnızca z/OS üzerinde desteklenir.

QueueAccounting (MQLONG)

<i>Çizelge 612. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	

Bu, kuyruğa ilişkin hesap verilerinin toplanmasını denetler. Bu kuyruk için toplanacak hesap verileri için, MQCONNX çağrısındaki MQCNO yapısında QMGR özneliği ACCTQ ya da Seçenekler alanı kullanılarak bu bağlantıya ilişkin muhasebe verileri de etkinleştirilmelidir.

Bu öznelik aşağıdaki değerlerden birini içeriyor:

MQMON_Q_MGR

Bu kuyruğa ilişkin hesap verileri, QMGR özneliği ACCTQ ayarına dayalı olarak toplanır. Bu varsayılan ayardır.

MQMON_OFF

Bu kuyruk için hesap verilerini toplamayın.

MQMON_ON

Bu kuyruk için hesap verilerini toplar.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_ACCOUNTING_Q seçicisini kullanın.

QueueMonitoring (MQLONG)

Kuyruklara ilişkin çevrimiçi izleme verilerinin toplanmasını denetler.

<i>Çizelge 613. Bu özneliğin geçerli olduğu kuyruk tipleri</i>				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Değer aşağıdakilerden biridir:

MQMON_Q_MGR

QueueMonitoring kuyruk yöneticisi özneliğinin ayarına göre izleme verilerini toplayın. Bu varsayılan değerdir.

MQMON_OFF

Bu kuyruk için çevrimiçi izleme verilerinin toplanması kapalı.

MQMON_LOW

QueueMonitoring kuyruk yöneticisi özneliğinin değeri MQMON_NONE değilse, çevrimiçi izleme verilerinin toplanması bu kuyruk için düşük veri toplama hızıyla açılır.

MQMON_ORTAMı

QueueMonitoring kuyruk yöneticisi özneliğinin değeri MQMON_NONE değilse, çevrimiçi izleme verilerinin toplanması bu kuyruk için orta hızda veri toplama ile açık olur.

MQMON_YükSEKLIK

QueueMonitoring kuyruk yöneticisi özneliğinin değeri MQMON_NONE değilse, çevrimiçi izleme verilerinin toplanması bu kuyruk için yüksek veri toplama hızıyla açılır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_MONITORING_Q seçicisini kullanın.

QueueStatistics (MQCHAR12)

Çizelge 614. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	

Bu, kuyruğa ilişkin istatistik verilerinin toplanmasını denetler.

Bu öznitelik aşağıdaki değerlerden birini içeriyor:

MQMON_Q_MGR

Bu kuyruğa ilişkin muhasebe verileri, QMGR özniteliği STATQ ayarına dayalı olarak toplanır. Bu varsayılan ayardır.

MQMON_OFF

Bu kuyruk için istatistik verilerini toplama işlemini kapatır.

MQMON_ON

Bu kuyruk için istatistik verilerinin toplanmasını etkinleştirir.

QType (MQLONG)

Çizelge 615. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Bu, kuyruğun tipidir; aşağıdaki değerlerden birine sahiptir:

MQQT_ALIAS

Diğer ad kuyruğu tanımlaması.

MQQT_CLUSTER

Küme kuyruğu.

MQQT_LOCAL

Yerel kuyruk.

MQQT_REMOTE

Uzak kuyruğun yerel tanımlaması.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_Q_TYPE seçicisini kullanın.

RemoteQMgrAdı (MQCHAR48)

Çizelge 616. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
			X	

Bu, **RemoteQName** kuyruğunun tanımlandığı uzak kuyruk yöneticisinin adıdır. **RemoteQName** kuyruğunun **QSGDisp** değeri MQQSGD_COPY ya da MQQSGD_SHARED ise, **RemoteQMgrName**, **RemoteQName**' un sahibi olan kuyruk paylaşım grubunun adı olabilir.

Bir uygulama uzak kuyruğun yerel tanımlamasını açarsa, **RemoteQMgrName** boş olmamalı ve yerel kuyruk yöneticisinin adı olmamalıdır. **XmitQName** boşsa, iletim kuyruğu olarak **RemoteQMgrName** ile aynı ada sahip yerel kuyruk kullanılır. **RemoteQMgrName** adlı bir kuyruk yoksa, **DefXmitQName** kuyruk yöneticisi özniteliğiyle tanımlanan kuyruk kullanılır.

Bu tanımlama bir kuyruk yöneticisi diğer adı için kullanılıyorsa, **RemoteQMgrName** , diğer adı kullanılmakta olan kuyruk yöneticisinin adıdır. Yerel kuyruk yöneticisinin adı olabilir. Ters durumda, **XmitQName** açıldığında boşsa, adı **RemoteQMgrName** ile aynı olan bir yerel kuyruk olmalıdır; bu kuyruk iletim kuyruğu olarak kullanılır.

Bu tanımlama bir yanıt diğer adı için kullanılıyorsa, bu ad, **ReplyToQMgr** olacak kuyruk yöneticisinin adıdır.

Not: Kuyruk tanımlaması yaratıldığında ya da değiştirildiğinde, bu öznitelik için belirtilen değer üzerinde geçerlilik denetimi gerçekleştirilmez.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_REMOTE_Q_MGR_NAME seçicisini kullanın. Bu özneliğin uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir.

RemoteQName (MQCHAR48)

Çizelge 617. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
			X	

Bu, **RemoteQMgrName** uzak kuyruk yöneticisinde bilinen kuyruğun adıdır.

Bir uygulama uzak kuyruğun yerel tanımını açarsa, **RemoteQName** açıldığında boş olmamalıdır.

Bu tanımlama bir kuyruk yöneticisi diğer adı tanımlaması için kullanılıyorsa, açma işlemi gerçekleştiğinde **RemoteQName** boş olmalıdır.

Tanım bir yanıt diğer adı için kullanılıyorsa, bu ad, **ReplyToQ** olacak kuyruğun adıdır.

Not: Kuyruk tanımlaması yaratıldığında ya da değiştirildiğinde, bu öznitelik için belirtilen değer üzerinde geçerlilik denetimi gerçekleştirilmez.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla MQCA_REMOTE_Q_NAME seçicisini kullanın. Bu özneliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

RetentionInterval (MQLONG)

Bu, kuyruğun alıkonacağı zaman dönemidir. Bu süre geçtikten sonra, kuyruk silinmeye uygun olur.

Çizelge 618. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Saat, kuyruğun yaratıldığı tarih ve saatten itibaren saat cinsinden ölçülür. Kuyruğun yaratılma tarihi ve saati, **CreationDate** ve **CreationTime** özniteliklerine kaydedilir.

Bu bilgiler, bir bakım uygulamasının ya da işletmenin artık gerekli olmayan kuyrukları tanımlamasını ve silmesini sağlamak için sağlanır.

Not: Kuyruk yöneticisi, bu özneliğe dayalı olarak kuyrukları silmek ya da süresi dolmamış bir alıkoyma aralığına sahip kuyrukların silinmesini önlemek için hiçbir işlem gerçekleştirmez; gerekli herhangi bir işlemi gerçekleştirmek kullanıcının sorumluluğundadır.

Kalıcı dinamik kuyrukların birikmesini önlemek için gerçekçi bir alıkoyma aralığı kullanın (bkz. [DefinitionType](#) özneliği). Ancak, bu öznitelik önceden tanımlanmış kuyruklarla da kullanılabilir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_RETENTION_INTERVAL seçicisini kullanın.

Kapsam (MQLONG)

Bu, bu kuyruğa ilişkin bir girişin bir hücre dizininde de bulunup bulunmadığını denetler.

Çizelge 619. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Kurulabilir bir Ad hizmeti tarafından bir hücre dizini sağlanır. Değer aşağıdakilerden biridir:

MQSCO_Q_MGR

Kuyruk tanımlamasının kuyruk yöneticisi kapsamı var: Kuyruğun tanımı, kuyruğun iyisi olan kuyruk yöneticisini aşmıyor. Kuyruğun başka bir kuyruk yöneticisinden çıkış için açılması için, sahip olan kuyruk yöneticisinin adı belirtilmelidir ya da diğer kuyruk yöneticisinin yerel bir kuyruk tanımlaması olmalıdır.

MQSCO_CELL

Kuyruk tanımlamasının hücre kapsamı vardır: Kuyruk tanımlaması, hücredeki tüm kuyruk yöneticilerinin kullanabileceği bir hücre dizinine de yerleştirilir. Kuyruk, kuyruğun adı belirtilerek, hücredeki kuyruk yöneticilerinden herhangi birinden çıkış için açılabilir; kuyruğun sahibi olan kuyruk yöneticisinin adı belirtilmesi gerekmez. Ancak, yerel tanımlama öncelikli olduğu için, bu adı taşıyan bir kuyruğun yerel tanımına da sahip olan, hücredeki kuyruk yöneticisi tarafından kuyruk tanımlaması kullanılamaz.

Kurulabilir bir Ad hizmeti tarafından bir hücre dizini sağlanır.

Model ve dinamik kuyrukların hücre kapsamı olamaz.

Bu değer, yalnızca bir hücre dizinini destekleyen bir ad hizmetinin konfigürasyonu tanımlandıysa geçerlidir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_SCOPE seçicisini kullanın.

Bu öznitelik için destek aşağıdaki kısıtlamalara tabidir:

- IBM işletim sisteminde öznitelik desteklenir, ancak yalnızca MQSCO_Q_MGR geçerlidir.
- z/OS işletim sisteminde öznitelik desteklenmez.

Paylaşılabilir (MQLONG)

Kuyruğun aynı anda birden çok kez giriş için açılıp açılmayacağını gösterir.

Çizelge 620. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Değer aşağıdakilerden biridir:

MQQA_SHAREABLE

Kuyruk paylaşılabilir.

MQOO_INPUT_SHARED seçeneğiyle birden çok açılmasına izin verilir.

MQQA_NOT_SHAREABLE

Kuyruk paylaşılabilir değil.

MQOO_INPUT_SHARED seçeneğiyle bir MQOPEN çağrısı MQOO_INPUT_EXCLUSIVE olarak kabul edilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_SHAREABILITY seçicisini kullanın.

StorageClass (MQCHAR8)

Bu, kuyruğu tutmak için kullanılan fiziksel saklama alanını tanımlayan kullanıcı tanımlı bir addır. Uygulamada, bir ileti yalnızca bellek arabelleğinin dışına sayfalanması gerektiğinde diske yazılır.

Çizelge 621. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQCA_STORAGE_CLASS seçicisini kullanın. Bu özniteliğin uzunluğu MQ_STORAGE_CLASS_LENGTH tarafından verilir.

z/OS Bu öznitelik yalnızca z/OS üzerinde desteklenir.

TriggerControl (MQLONG)

Bu, kuyruğa hizmet vermek üzere bir uygulama başlatmak için tetikleyici iletilerin bir başlatma kuyruğuna yazılıp yazılmayacağını denetler.

Çizelge 622. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, aşağıdakilerden biridir:

MQTC_OFF

Bu kuyruk için yazılacak tetikleyici ileti yok. *TriggerType* değeri bu durumda alakasız.

MQTC_ON

Tetikleyici iletiler, uygun tetikleyici olayları oluştuğunda bu kuyruk için yazılır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_TRIGGER_CONTROL seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

TriggerData (MQCHAR64)

Kuyruk yöneticisinin, bu kuyruğa gelen bir ileti, başlatma kuyruğuna bir tetikleyici iletilerinin yazılmasına neden olduğunda tetikleyici iletiye eklediği serbest biçimli verilerdir.

Çizelge 623. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu verilerin içeriği kuyruk yöneticisi için önemli değildir. Bu, başlatma kuyruğunu işleyen tetikleyici uygulaması ya da tetikleyici izleme programının başlatıldığı uygulama için anlamlıdır.

Karakter dizisi boş değer içermemelidir. Gerekirse sağa doğru boşluklarla doldurulur.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_TRIGGER_DATA seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın. Bu özniteliğin uzunluğu MQ_TRIGGER_DATA_LENGTH tarafından verilir.

TriggerDepth (MQLONG)

Çizelge 624. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, bir tetikleyici iletilerinin yazılmadan önce kuyruğa olması gereken *TriggerMsgPriority* ya da daha yüksek öncelikli iletilerinin sayısıdır. Bu, *TriggerType* MQTT_DEPTH olarak ayarlandığında geçerlidir. *TriggerDepth* değeri bir ya da daha büyük. Aksi halde bu öznitelik kullanılmaz.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_TRIGGER_DEPTH seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

TriggerMsgÖnceliği (MQLONG)

Bu, iletilerin tetikleyici iletilerin oluşturulmasına katkıda bulunmadığı ileti önceliğidir (yani, kuyruk yöneticisi bir tetikleyici ileti oluşturulup oluşturulmayacağına karar verirken bu iletileri yoksayar).

Çizelge 625. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

TriggerMsgPriority, sıfır (en düşük)-*MaxPriority* aralığında olabilir (yüksek; bkz. *MaxPriority* özniteliği); sıfır değeri, tüm iletilerin tetikleyici iletileri oluşturulmasına katkıda bulunmasına neden olur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla MQIA_TRIGGER_MSG_PRIORITY seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

TriggerType (MQLONG)

Bu, bu kuyruğa gelen iletilerin sonucu olarak tetikleyici iletilerin yazılacağı koşulları denetler.

Çizelge 626. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Aşağıdaki değerlerden birini içerir:

MQTT_NONE

Bu kuyruktaki iletilerin sonucu olarak tetikleyici ileti yazılmaz. Bu, *TriggerControl* 'un MQTC_OFF olarak ayarlanmasıyla aynı etkiye sahiptir.

MQTT_FIRST

Kuyruktaki *TriggerMsgPriority* ya da daha yüksek öncelikli iletilerin sayısı 0 'dan 1 'e değiştiğinde bir tetikleyici iletisi yazılır.

MQTT_HER

Kuyruğa *TriggerMsgPriority* ya da daha yüksek öncelikli bir ileti geldiğinde tetikleyici ileti yazılır.

MQTT_DERINLIĞI

Kuyruktaki *TriggerMsgPriority* ya da daha yüksek öncelikli iletilerin sayısı *TriggerDepth* değerine eşit ya da değerini aşarsa bir tetikleyici iletisi yazılır. Tetikleyici iletisi yazıldıktan sonra, *TriggerControl* belirttik olarak yeniden açılıncaya kadar daha fazla tetikleme için MQTC_OFF olarak ayarlanır.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_TRIGGER_TYPE seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

Kullanım (MQLONG)

Bu, kuyruğun ne için kullanıldığını gösterir.

Çizelge 627. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Değer aşağıdakilerden biridir:

MQUS_NORMAL

Bu, uygulamaların ileti yerleştirirken ve alınırken kullandığı bir kuyruktur; kuyruk bir iletim kuyruğu değildir.

MQUS_ILETIMI

Bu, uzak kuyruk yöneticilerine gönderilen iletileri tutmak için kullanılan bir kuyruktur. Bir uygulama uzak bir kuyruğa ileti gönderdiğinde, yerel kuyruk yöneticisi iletiyi geçici olarak uygun iletim kuyruğuna özel bir biçimde saklar. Bir ileti kanalı aracısı, iletiyi iletim kuyruğundan okur ve iletiyi uzak kuyruk yöneticisine iletir. Uzak yönetim yapılandırılmasına ilişkin ek bilgi için [Uzak denetim için kuyruk yöneticilerinin yapılandırılması](#) başlıklı konuya bakın.

İletileri doğrudan üzerine koymak için MQOO_OUTPUT için bir iletim kuyruğunu yalnızca ayrıcalıklı uygulamalar açabilir. Genellikle, bunu yalnızca yardımcı uygulamalar yapar. İleti veri biçiminin doğru olduğundan emin olun (bkz. "MQXQH-İletim kuyruğu üstbilgisi" sayfa 612) ya da iletim işlemi sırasında hatalar ortaya çıkabilir. MQPMO_*_CONTEXT bağlam seçeneklerinden biri belirtilmedikçe bağlam geçirilmez ya da ayarlanmaz.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_USAGE seçicisini kullanın.

XmitQName (MQCHAR48)

Bu, iletim kuyruğu adıdır. Bir uzak kuyruk ya da kuyruk yöneticisi diğer adı tanımlaması açıldığında bu öznitelik boş değilse, iletiyi iletmek için kullanılacak yerel iletim kuyruğunun adını belirler.

Çizelge 628. Bu özniteliğin geçerli olduğu kuyruk tipleri

Yerel	Modeli	Diğer Ad	Uzak	Küme
			X	

XmitQName alanı boşsa, iletim kuyruğu olarak **RemoteQMgrName** ile aynı adı taşıyan yerel kuyruk kullanılır. **RemoteQMgrName** adlı bir kuyruk yoksa, **DefXmitQName** kuyruk yöneticisi özniteliğiyle tanımlanan kuyruk kullanılır.

Tanımlama bir kuyruk yöneticisi diğer adı olarak kullanılıyorsa ve **RemoteQMgrName** yerel kuyruk yöneticisinin adıysa, bu öznitelik yoksayılr. Tanım, yanıtlanacak kuyruk diğer adı tanımlaması olarak kullanılıyorsa da yoksayılr.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_XMIT_Q_NAME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

Ad listesi öznitelikleri

Aşağıdaki çizelge, ad çizelgelerine özgü öznitelikleri özetler. Öznitelikler alfabetik sırayla açıklanır.

Namelistler, bu sistemlere bağlı IBM MQ MQI clients ve tüm IBM MQ sistemlerinde desteklenir.

Not: Bu kısımda gösterilen özniteliklerin adları, MQINQ ve MQSET çağrılarıyla kullanılan açıklayıcı adlardır; adlar PCF komutlarıyla aynıdır. Öznitelikleri tanımlamak, değiştirmek ya da görüntülemek için MQSC komutları kullanıldığında, diğer kısa adlar kullanılır; ek bilgi için [MQSC komutları](#) konusuna bakın.

Çizelge 629. Ad listesi öznitelikleri

Öznitelik	Açıklama
AlterationDate	Tanımlamanın en son değiştirildiği tarih
AlterationTime	Tanımlamanın en son değiştirildiği zaman
NameCount	Ad listesindeki ad sayısı
NamelistDesc	Ad listesi açıklaması
NamelistName	Ad listesi adı
Adlar	<i>NameCount</i> adlarının bir listesi
NamelistType	Ad listesi tipi
QSGDisp	Kuyruk paylaşım grubu yok etme

AlterationDate (MQCHAR12)

Bu, tanımın en son değiştirildiği tarihtir. Tarih biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için sondaki iki boşlukla doldurulur.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_ALTERATION_DATE seçicisini kullanın. Bu özniteliğin uzunluğu MQ_DATE_LENGTH tarafından verilir.

AlterationTime (MQCHAR8)

Bu, tanımın en son değiştirildiği zamandır. Saatin biçimi şöyledir: HH.MM.SS.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_ALTERATION_TIME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_TIME_LENGTH tarafından verilir.

NameCount (MQLONG)

Bu, ad listesindeki isimlerin sayısıdır. Sıfırdan büyük ya da sıfıra eşittir. Aşağıdaki değer tanımlandı:

MQNC_MAX_NAMELIST_NAME_COUNT

Ad listelerinde ad sayısı üst sınırı.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_NAME_COUNT seçicisini kullanın.

NamelistDesc (MQCHAR64)

Açıklayıcı açıklama için bu alanı kullanın; değeri tanımlama işlemi tarafından oluşturulur. Alanın içeriği kuyruk yöneticisi için önemli değil, ancak kuyruk yöneticisi alanın yalnızca görüntülenebilecek karakterleri içermesini gerektirebilir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, bu alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 byte olabilir).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özniteliğinde tanımlandığı gibi) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_NAMELIST_DESC seçicisini kullanın.

Bu özniteliğin uzunluğu MQ_NAMELIST_DESC_LENGTH tarafından verilir.

NamelistName (MQCHAR48)

Bu, yerel kuyruk yöneticisinde tanımlı bir ad listesi adıdır. Ad listesi adlarıyla ilgili daha fazla bilgi için [Diğer nesne adları](#) bölümüne bakın.

Her ad listesi, kuyruk yöneticisine ait diğer ad listelerinin adlarından farklı bir ada sahiptir, ancak farklı tiplerdeki (örneğin, kuyruklar) diğer kuyruk yöneticisi nesnelerinin adlarını yineleyebilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_NAMELIST_NAME seçicisini kullanın.

Bu özniteliğin uzunluğu MQ_NAMELIST_NAME_LENGTH tarafından verilir.

NamelistType (MQLONG)

Bu, ad listesindeki adların niteliğini belirtir ve ad değerinin nasıl kullanıldığını gösterir. Aşağıdaki değerlerden biridir:

MQNT_NONE

Atanmış tipi olmayan ad listesi.

MQNT_Q

Kuyrukların adlarını içeren ad listesi.

MQNT_CLUSTER

Küme adlarını içeren ad listesi.

MQNT_AUTH_INFO

Kimlik doğrulama bilgisi nesnelerinin adlarını içeren ad listesi.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_NAMELIST_TYPE seçicisini kullanın.

z/OS Bu öznitelik yalnızca z/OSüzerinde desteklenir.

Adlar (MQCHAR48xNameCount)

Bu, *NameCount* adlarının bir listesidir; burada her ad, yerel kuyruk yöneticisine tanımlanan bir nesnenin adıdır. Nesne adlarıyla ilgili ek bilgi için [IBM MQ nesnelere adlandırma kuralları](#) başlıklı konuya bakın.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQCA_NAMES seçicisini kullanın.

Listedeki her adın uzunluğu MQ_OBJECT_NAME_LENGTH tarafından verilir.

QSGDisp (MQLONG)

Bu, ad listesi 'nin atılacağını belirtir. Değer aşağıdakilerden biridir:

MQQSGD_Q_MGR

Nesnenin kuyruk yöneticisi atma işlemi var: Nesne tanımlaması yalnızca yerel kuyruk yöneticisi tarafından bilinir; tanımlama, kuyruk paylaşım grubundaki diğer kuyruk yöneticileri tarafından bilinmez.

Kuyruk paylaşım grubundaki her kuyruk yöneticisinin yürürlükteki nesneyle aynı ada ve tipe sahip bir nesnesi olabilir; ancak, bunlar ayrı nesnelere ve bunlar arasında ilinti yoktur. Öznitelikleri birbiriyle aynı olacak şekilde kısıtlanmaz.

MQQSGD_COPY

Nesne, paylaşılan havuzda var olan bir ana nesne tanımlamasının yerel bir kopyasıdır. Kuyruk paylaşım grubundaki her kuyruk yöneticisinin nesnenin kendi kopyası olabilir. Başlangıçta, tüm kopyalar aynı özniteliklere sahiptir, ancak her kopyayı MQSC komutlarını kullanarak diğer kopyaların özniteliklerinden farklı olacak şekilde değiştirebilirsiniz. Paylaşılan havuzdaki ana tanımlama değiştirildiğinde, kopyaların öznitelikleri yeniden eşitlenir.

Bu özneliğin değerini saptamak için MQINQ çağrısıyla MQIA_QSG_DISP seçicisini kullanın.

z/OS Bu öznitelik yalnızca z/OSüzerinde desteklenir.

Süreç tanımlamalarına ilişkin öznitelikler

Aşağıdaki çizelge, süreç tanımlamalarına özgü öznitelikleri özetler. Öznitelikler alfabetik sırayla açıklanır.

Not: Bu kısımdaki özniteliklerin adları, MQINQ ve MQSET çağrılılarıyla kullanılan açıklayıcı adlardır; adlar PCF komutlarıyla aynıdır. Öznitelikleri tanımlamak, değiştirmek ya da görüntülemek için MQSC komutları kullanıldığında, diğer kısa adlar kullanılır; ek bilgi için [MQSC komutları](#) konusuna bakın.

Öznitelik	Açıklama
AlterationDate	Tanımlamanın en son değiştirildiği tarih
AlterationTime	Tanımlamanın en son değiştirildiği zaman
AppId	Uygulama tanıtıcısı
AppType	Uygulama tipi
EnvData	Ortam verileri
ProcessDesc	Süreç Tanımı
ProcessName	İşlem adı
QSGDisp	Kuyruk paylaşım grubu yok etme
UserData	Kullanıcı verileri

AlterationDate (MQCHAR12)

Bu, tanımlamanın en son değiştirildiği tarihtir. Tarih biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için sondaki iki boşlukla doldurulur.

Bu özniteliğin deęerini saptamak için MQINQ çağırısıyla MQCA_ALTERATION_DATE seçicisini kullanın. Bu özniteliğin uzunluğu MQ_DATE_LENGTH tarafından verilir.

AlterationTime (MQCHAR8)

Bu, tanımın en son deęiştirildięi zamandır. Saatin biçimi şöyledir: HH.MM.SS.

Bu özniteliğin deęerini saptamak için MQINQ çağırısıyla MQCA_ALTERATION_TIME seçicisini kullanın. Bu özniteliğin uzunluğu MQ_TIME_LENGTH tarafından verilir.

AppId (MQCHAR256)

Bu, başlatılacak uygulamayı tanıtan bir karakter dizilimidir. Bu bilgiler, başlatma kuyruğundaki iletileri işleyen bir tetikleyici uygulaması tarafından kullanılır; bilgiler, tetikleme iletilerinin bir parçası olarak başlatma kuyruğuna gönderilir.

AppId 'in anlamı, tetikleyici-izleyici uygulaması tarafından belirlenir. IBM MQ tarafından sağlanan tetikleyici, *AppId* 'in yürütülebilir bir programın adı olmasını gerektirir. Aşağıdaki notlar, belirtilen ortamlar için geçerlidir:

- z/OS üzerinde, *AppId* aşağıdaki gibi olmalıdır:
 - CICS tetikleyici izleme işlemi CKTI kullanılarak başlatılan uygulamalar için bir CICS işlem tanıtıcısı
 - IMS tetikleyici izleme programı CSQQTRMN kullanılarak başlatılan uygulamalara ilişkin IMS hareket tanıtıcısı
- Windows' da, program adının başına bir sürücü ve izin yolu öneki konabilir.
- AIX and Linux üzerinde, program adının başına bir izin yolu öneki konabilir.

Karakter dizgisi boş deęer içeremez. Gerekirse sağa doğru boşluklarla doldurulur.

Bu özniteliğin deęerini saptamak için MQINQ çağırısıyla MQCA_APPL_ID seçicisini kullanın. Bu özniteliğin uzunluğu MQ_PROCESS_APPL_ID_LENGTH tarafından verilir.

AppType (MQLONG)

Bu, bir tetikleyici iletilerinin alınmasına yanıt olarak başlatılacak programın niteliğini tanımlar. Bu bilgiler, başlatma kuyruğundaki iletileri işleyen bir tetikleyici uygulaması tarafından kullanılır; bilgiler, tetikleme iletilerinin bir parçası olarak başlatma kuyruğuna gönderilir.

AppType 'in herhangi bir deęeri olabilir, ancak standart tipler için şu deęerler önerilir; kullanıcı tanımlı uygulama tiplerini MQAT_USER_LAST ile MQAT_USER_FIRST aralığındaki deęerlerle sınırlandırın:

MQAT_AIX

AIX uygulaması (MQAT_UNIX ile aynı deęer).

MQAT_BATCH

Toplu iş uygulaması

MQAT_CICS

CICS hareketi.

MQAT_IMS

IMS uygulaması.

MQAT_IMS_BRIDGE

IMS köprü uygulaması.

MQAT_JAVA

Java uygulaması.

MQAT_MVS

MVS ya da TSO uygulaması (MQAT_ZOS ile aynı deęer).

MQAT_OS390

OS/390 uygulaması (MQAT_ZOS ile aynı deęer).

MQAT_OS400

IBM i uygulaması.

MQAT_UNIX

UNIX uygulaması.

MQAT_BILINMIYOR

Bilinmeyen tipte uygulama.

MQAT_KULLANICISI

Kullanıcı uygulaması.

MQAT_WINDOWS

64 bit Windows uygulaması.

MQAT_WINDOWS_NT

32 bit Windows uygulaması.

MQAT_WLM

z/OS iş yükü yöneticisi uygulaması.

MQAT_ZOS

z/OS uygulaması.

MQAT_USER_FIRST

Kullanıcı tanımlı uygulama tipi için en düşük değer.

MQAT_USER_LAST

Kullanıcı tanımlı uygulama tipi için en yüksek değer.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_APPL_TYPE seçicisini kullanın.

EnvData (MQCHAR128)

Bu, başlatılacak uygulamayla ilgili ortamla ilgili bilgileri içeren bir karakter dizilimidir. Bu bilgiler, başlatma kuyruğundaki iletileri işleyen bir tetikleyici uygulaması tarafından kullanılır; bilgiler, tetikleme iletilerinin bir parçası olarak başlatma kuyruğuna gönderilir.

EnvData 'in anlamı, tetikleyici-izleyici uygulaması tarafından belirlenir. IBM MQ tarafından sağlanan tetikleyici izleme programı, başlatılan uygulamaya geçirilen parametre listesinin sonuna *EnvData* eklenir. Değiştirge listesi, MQTMC2 yapısından oluşur. Ardından bir boşluk ve ardından sondaki boşluklar kaldırılmış *EnvData* gelir. Aşağıdaki notlar, belirtilen ortamlar için geçerlidir:

- z/OS'ta:
 - *EnvData* , IBM MQ tarafından sağlanan tetikleyici izleyici uygulamaları tarafından kullanılmaz.
 - ApplType MQAT_WLM ise, iş bilgileri üstbilgisindeki (MQWIH) ServiceName ve ServiceStep alanları için EnvData içindeki varsayılan değerleri belirleyebilirsiniz.
- AIX and Linux sistemlerinde *EnvData* , arka planda başlatılan uygulamayı çalıştırmak için & karakterine ayarlanabilir.

Karakter dizgisi boş değer içeremez. Gerekirse sağa doğru boşluklarla doldurulur.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_ENV_DATA seçicisini kullanın. Bu özniteliğin uzunluğu MQ_PROCESS_ENV_DATA_LENGTH tarafından verilir.

ProcessDesc (MQCHAR64)

Açıklayıcı açıklama için bu alanı kullanın. Alanın içeriği kuyruk yöneticisi için önemli değil, ancak kuyruk yöneticisi alanın yalnızca görüntülenebilecek karakterleri içermesini gerektirebilir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 bayttır).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özniteliğinde tanımlandığı gibi) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_PROCESS_DESC seçicisini kullanın.

Bu özniteliğin uzunluğu MQ_PROCESS_DESC_LENGTH tarafından verilir.

ProcessName (MQCHAR48)

Bu, yerel kuyruk yöneticisinde tanımlanan bir süreç tanımlamasının adıdır.

Her süreç tanımlamasının adı, kuyruk yöneticisine ilişkin diğer süreç tanımlamalarının adlarından farklıdır. Ancak süreç tanımlamasının adı, farklı tiplerdeki (örneğin, kuyruklar) diğer kuyruk yöneticisi nesnelere aynı olabilir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_PROCESS_NAME seçicisini kullanın.

Bu özniteliğin uzunluğu MQ_PROCESS_NAME_LENGTH tarafından verilir.

QSGDisp (MQLONG)

Bu, süreç tanımlamasının yok edileceğini belirtir. Değer aşağıdakilerden biridir:

MQQSGD_Q_MGR


Nesnenin kuyruk yöneticisi atma işlemi var: Nesne tanımlaması yalnızca yerel kuyruk yöneticisi tarafından bilinir; tanımlama, kuyruk paylaşım grubundaki diğer kuyruk yöneticileri tarafından bilinmez.

Kuyruk paylaşım grubundaki her kuyruk yöneticisinin yürürlükteki nesneyle aynı ada ve tipe sahip nesnesi olabilir; ancak, bunlar ayrı nesnelere ve bunlar arasında ilinti yoktur. Öznitelikleri birbiriyle aynı olacak şekilde kısıtlanmaz.

MQQSGD_COPY

Nesne, paylaşılan havuzda var olan bir ana nesne tanımlamasının yerel bir kopyasıdır. Kuyruk paylaşım grubundaki her kuyruk yöneticisinin nesnenin kendi kopyası olabilir. Başlangıçta, tüm kopyalar aynı özniteliklere sahiptir, ancak her kopyayı MQSC komutlarını kullanarak diğer kopyaların özniteliklerinden farklı olacak şekilde değiştirebilirsiniz. Paylaşılan havuzdaki ana tanımlama değiştirildiğinde, kopyaların öznitelikleri yeniden eşitlenir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQIA_QSG_DISP seçicisini kullanın.

 Bu öznitelik yalnızca z/OS üzerinde desteklenir.

UserData (MQCHAR128)

UserData , başlatılacak uygulamayla ilgili kullanıcı bilgilerini içeren bir karakter dizisidir. Bu bilgiler, başlatma kuyruğundaki iletileri ya da tetikleyici izleme programı tarafından başlatılan uygulamayı işleyen bir tetikleyici uygulaması tarafından kullanılır. Bilgiler, tetikleme iletilerinin bir parçası olarak başlatma kuyruğuna gönderilir.

UserData ' in anlamı, tetikleyici-izleyici uygulaması tarafından belirlenir. IBM MQ tarafından sağlanan tetikleyici izleme programı, parametre listesinin bir parçası olarak başlatılan uygulamaya UserData iletir. Değiştirge listesi, MQTMC2 yapısından (UserData içeren) oluşur; bunu izleyen bir boşluk ve ardından boşluk kaldırılmış EnvData izler.

Karakter dizisi boş değer içeremez. Gerekirse sağa doğru boşluklarla doldurulur. Microsoft Windows için, süreç tanımlaması runmqtrm' e geçirilecekse, karakter dizisi çift tırnak işareti içermemelidir.

Bu özniteliğin değerini saptamak için MQINQ çağrısıyla MQCA_USER_DATA seçicisini kullanın. Bu özniteliğin uzunluğu MQ_PROCESS_USER_DATA_LENGTH tarafından verilir.

Dönüş kodları

Her IBM MQ Message Queue Interface (MQI) ve IBM MQ Administration Interface (MQAI) çağrısı için, kuyruk yöneticisi ya da bir çıkış yordamı tarafından bir **tamamlanma** kodu ve bir **neden** kodu döndürülür.

Uygulamalar, özellikle belirtildiği durumlar dışında, belirli bir sırada denetlenen hatalara bağlı olmamalıdır. Bir çağrıdan birden fazla tamamlanma kodu ya da neden kodu oluşabiliyorsa, bildirilen hata uygulamaya bağlıdır.

Bir IBM MQ API çağrısını izleyen başarılı tamamlama olup olmadığını denetleyen uygulamaların her zaman tamamlanma kodunu denetlemesi gerekir. Neden kodunun değerine dayalı olarak, tamamlanma kodu değerini varsaymayın.

Tamamlanma kodları

Tamamlama kodu parametresi (*CompCode*), çağırının, çağırının başarıyla tamamlanıp tamamlanmadığını, kısmen tamamlandığını ya da başarısız olup olmadığını hızlı bir şekilde görmesini sağlar. Aşağıda, çağrı açıklamalarında belirtilenden daha fazla ayrıntıya sahip tamamlanma kodlarının bir listesi yer almaktadır:

MQCC_OK

Çağrı tam olarak tamamlandı; tüm çıkış parametreleri ayarlandı. Bu durumda **Reason** parametresi her zaman MQRC_NONE değerine sahiptir.

MQCC_UYARISI

Arama kısmen tamamlandı. *CompCode* ve *Reason* çıkış parametrelerine ek olarak bazı çıkış parametreleri ayarlanmış olabilir. **Reason** parametresi, kısmi tamamlama hakkında ek bilgi verir.

MQCC_FAILED

Aramanın işlenmesi tamamlanmadı. Özellikle belirtildiği durumlar dışında, kuyruk yöneticisinin durumu değişmez. *CompCode* ve *Reason* çıkış parametreleri ayarlandı; belirtilenler dışında diğer parametreler değiştirilmez.

Bunun nedeni, uygulama programındaki bir hata olabilir ya da programın dışındaki bir durumun (örneğin, kullanıcının yetkisi iptal edilmiş olabilir) sonucu olabilir. **Reason** parametresi, hatayla ilgili ek bilgi verir.

Neden kodları

Neden kodu parametresi (*Reason*), tamamlanma kodu parametresini (*CompCode*) niteler.

Raporlamak için özel bir neden yoksa, MQRC_NONE döndürülür. Başarılı bir çağrı MQCC_OK ve MQRC_NONE değerini döndürür.

Tamamlanma kodu MQCC_WARNING ya da MQCC_FAILED ise, kuyruk yöneticisi her zaman uygun bir neden bildirir; ayrıntılar her çağrı açıklaması altında verilir.

Kullanıcı çıkışı yordamlarının tamamlanma kodlarını ve nedenlerini belirlediği durumlarda, bu kurallara uymaları gerekir. Ayrıca, kullanıcı çıkışları tarafından tanımlanan özel neden değerlerinin, kuyruk yöneticisi tarafından tanımlanan değerlerle çakışmamasını sağlamak için sıfırdan küçük olması gerekir. Çıkışlar, kuyruk yöneticisi tarafından önceden tanımlanmış nedenleri (uygun olduğu durumlarda) belirleyebilir.

Neden kodları aşağıdaki durumlarda da oluşur:

- MQDLH yapısının *Reason* alanı
- MQMD yapısının *Feedback* alanı

Neden kodlarının tam açıklamaları için bkz. [İletiler ve neden kodları](#).

MQI seçeneklerinin geçerliliğini denetleme kuralları

Bu kısımda, bir MQOPEN, MQPUT, MQPUT1, MQGET, MQCLOSE ya da MQSUB çağrısından MQRC_OPTIONS_ERROR neden kodu üreten durumlar listelenir.

MQOPEN çağrısı

MQOPEN çağrısının seçenekleri için:

- Aşağıdakilerden en az *biri* belirtilmelidir:
 - MQOO_GÖZ AT
 - MQOO_INPUT_EXCLUSIVE ¹
 - MQOO_INPUT_SHARED ¹

- MQOO_INPUT_AS_Q_DEF ¹
- MQOO_INQUIRE
- MQOO_ÇıKışı
- MQOO_SET
- MQOO_BIND_ON_OPEN ²
- MQOO_BIND_NOT_FIXED ²
- MQOO_BIND_ON_GROUP ²
- MQOO_BIND_AS_Q_DEF ²
- Aşağıdakilerden yalnızca *birine* izin verilir:
 - MQOO_READ_ÖNDEN
 - MQOO_NO_READ_ÖNDEN
 - MQOO_READ_AHEAD_AS_Q_DEF

1. Aşağıdakilerden yalnızca *birine* izin verilir:

- MQOO_INPUT_EXCLUSIVE
- MQOO_INPUT_SHARED
- MQOO_INPUT_AS_Q_DEF

2. Aşağıdakilerden yalnızca *birine* izin verilir:

- MQOO_BIND_ON_OPEN
- MQOO_BIND_NOT_FIXED
- MQOO_BIND_ON_GROUP
- MQOO_BIND_AS_Q_DEF

Not: Daha önce listelenen seçenekler karşılıklı olarak birbirini dışlar. Ancak, MQOO_BIND_AS_Q_DEF değeri sıfır olduğundan, bu değer diğer iki bağ tanımlama seçeneğiyle birlikte belirtilmesi MQRC_OPTIONS_ERROR neden koduyla sonuçlanmaz. Program belgelerine yardımcı olmak için MQOO_BIND_AS_Q_DEF sağlanır.

- MQOO_SAVE_ALL_CONTEXT belirtilirse, MQOO_INPUT_* seçeneklerinden biri de belirtilmelidir.
- MQOO_SET_*_CONTEXT ya da MQOO_PASS_*_CONTEXT seçeneklerinden biri belirtilirse, MQOO_OUTPUT da belirtilmelidir.
- MQOO_CO_OP belirtilirse, MQOO_BROWSE de belirtilmelidir
- MQOO_NO_MULTICAST belirtilirse, MQOO_OUTPUT da belirtilmelidir.

MQPUT çağırısı

Koyma iletisi seçenekleri için:

- MQPMO_SYNCPOINT ve MQPMO_NO_SYNCPOINT birleşimine izin verilmez.
- Aşağıdakilerden yalnızca *birine* izin verilir:
 - MQPMO_DEFAULT_CONTEXT
 - MQPMO_NO_CONTEXT
 - MQPMO_PASS_ALL_CONTEXT
 - MQPMO_PASS_IDENTITY_CONTEXT
 - MQPMO_SET_ALL_CONTEXT
 - MQPMO_SET_IDENTITY_CONTEXT
- Aşağıdakilerden yalnızca *birine* izin verilir:
 - MQPMO_ASYNC_RESPONSE (mqpmo_async_yanıtı)

- MQPMO_SYNC_RESPONSE
- MQPMO_RESPONSE_AS_TOPIC_DEF
- MQPMO_RESPONSE_AS_Q_DEF
- MQPMO_ALTERNATE_USER_AUTHORITY ögesine izin verilmez (yalnızca MQPUT1 çağrısında geçerlidir).

MQPUT1 çağrısı

Koyma iletisi seçenekleri için kurallar, aşağıdakiler dışında MQPUT çağrısıyla aynıdır:

- MQPMO_ALTERNATE_USER_AUTHORITY ögesine izin verilir.
- MQPMO_LOGICAL_ORDER ögesine izin verilmiyor.

MQGET çağrısı

İleti alma seçenekleri için:

- Aşağıdakilerden yalnızca *birine* izin verilir:
 - MQGMO_NO_SYNCPOINT
 - MQGMO_SYNCPOINT
 - MQGMO_SYNCPOINT_IF_PERSISTENT
- Aşağıdakilerden yalnızca *birine* izin verilir:
 - MQGMO_BROWSE_FIRST
 - MQGMO_BROWSE_MSG_UNDER_CURSOR
 - MQGMO_BROWSE_NEXT
 - MQGMO_MSG_UNDER_CURSOR
- MQGMO_SYNCPOINT aşağıdaki öğelerle birlikte kullanılamaz:
 - MQGMO_BROWSE_FIRST
 - MQGMO_BROWSE_MSG_UNDER_CURSOR
 - MQGMO_BROWSE_NEXT
 - MQGMO_KİLİK
 - MQGMO_UNLOCK
- MQGMO_SYNCPOINT_IF_PERSISTENT, aşağıdakilerden herhangi biriyle birlikte kullanılamaz:
 - MQGMO_BROWSE_FIRST
 - MQGMO_BROWSE_MSG_UNDER_CURSOR
 - MQGMO_BROWSE_NEXT
 - MQGMO_COMPLETE_MSG
 - MQGMO_UNLOCK
- MQGMO_MARK_SKIP_BACKOUT için MQGMO_SYNCPOINT belirtilmelidir.
- MQGMO_WAIT ve MQGMO_SET_SIGNAL birleşimi kullanılamaz.
- MQGMO_LOCK belirtilirse, aşağıdakilerden biri de belirtilmelidir:
 - MQGMO_BROWSE_FIRST
 - MQGMO_BROWSE_MSG_UNDER_CURSOR
 - MQGMO_BROWSE_NEXT
- MQGMO_UNLOCK belirtilirse, yalnızca şu değerlere izin verilir:
 - MQGMO_NO_SYNCPOINT
 - MQGMO_NO_WAIT

MQCLOSE çağrısı

MQCLOSE çağrısının seçenekleri için:

- MQCO_DELETE ve MQCO_DELETE_PURGE birleşimine izin verilmez.
- Aşağıdakilerden yalnızca birine izin verilir:
 - MQCO_KEEP_SUB
 - MQCO_REMOVE_SUB

MQSUB çağrısı

MQSUB çağrısının seçenekleri için:

- Aşağıdakilerden en az biri belirtilmelidir:
 - MQSO ALTER
 - MQSO RESUME
 - MQSO OLUŞTURMA
- Aşağıdakilerden yalnızca birine izin verilir:
 - MQSO DURABLE
 - MQSO_NON_DURABLE

Not: Daha önce listelenen seçenekler karşılıklı olarak birbirini dışlar. Ancak, MQSO_NON_DURABLE değeri sıfır olduğundan, MQSO_DURABLE ile belirtilmesi MQRC_OPTIONS_ERROR neden koduyla sonuçlanmaz. Program belgelerine yardımcı olmak için MQSO_NON_DURABLE sağlanmıştır.

- MQSO_GROUP_SUB ve MQSO_MANAGED birleşimine izin verilmez.
- MQSO_GROUP_SUB için MQSO_SET_CORREL_ID belirtilmelidir.
- Aşağıdakilerden yalnızca birine izin verilir:
 - MQSO_ANY_USERID
 - MQSO_FIXED_USERID
- MQSO_NEW_PUBLICATIONS_ONLY şununla birlikte kullanılabilir:
 - MQSO OLUŞTURMA
 - MQSO ALTER, özgün abonelikte MQSO_NEW_PUBLICATIONS_ONLY ayarlandıysa
- 1'den büyük MQSO_PUBLICATIONS_ON_REQUEST ve SubLevel birleşimine izin verilmez.
- Aşağıdakilerden yalnızca birine izin verilir:
 - MQSO_WILDCARD_CHAR
 - MQSO_WILDCARD_TOPIC
- MQSO_NO_MULTICAST, MQSO_MANAGED özelliğinin belirtilmesini gerektirir.

Kuyruğa alınan yayınlama/abone olma komut iletileri

Bir uygulama, kuyruğa alınmış bir yayınlama/abone olma uygulamasını denetlemek için MQRFH2 komut iletilerini kullanabilir.

Yayınlama/abone olma için MQRFH2 kullanan bir uygulama, SYSTEM.BROKER.CONTROL.QUEUE:

- [“Yayın iletisini sil” sayfa 868](#)
- [“Abone Kaydını Kaldır iletisi” sayfa 869](#)
- [“İleti yayınla” sayfa 873](#)
- [“Abone iletisini kaydet” sayfa 875](#)

- [“Güncelleme İste iletisi” sayfa 879](#)

Kuyruğa alınmış yayınlama/abone olma uygulamaları yazıyorsanız, bu iletileri, kuyruk yöneticisi yanıt iletisini ve ileti tanımlayıcısını (MQMD) anlamanız gerekir; aşağıdaki bilgilere bakın:

- [“Kuyruk Yöneticisi Yanıt iletisi” sayfa 881](#)
- [“Kuyruk yöneticisi tarafından iletilen yayınlara ilişkin MQMD ayarları” sayfa 886](#)
- [“Kuyruk yöneticisi yanıt iletilerindeki MQMD ayarları” sayfa 887](#)
- [“Yayınlama/abone olma neden kodları” sayfa 883](#)

Komutlar, MQRFH2 üstbilgisinin **NameValueData** alanındaki bir psc klasöründe bulunur. Bir komut iletisine yanıt olarak aracı tarafından gönderilebilecek ileti bir pscr klasöründe bulunur.

Her komutun tanımlamaları, bir klasörde bulunabilecek özellikleri listeler. Aksi belirtilmedikçe, özellikler isteğe bağlıdır ve yalnızca bir kez oluşabilir.

Özelliklerin adları <Command>olarak gösterilir.

Değerler dizgi biçiminde olmalıdır, örneğin: Publish.

Bir özelliğin değerini gösteren bir dizgi değişmezi parantez içinde gösterilir, örneğin: (MQPSC_PUBLISH).

Dizgi değişmezleri, kuyruk yöneticisiyle birlikte sağlanan cmqpsc . h üstbilgi dosyasında tanımlanır.

Yayın iletisini sil

Delete Publication komut iletisi, kuyruk yöneticisine belirtilen konulara ilişkin alıkonan yayınları silmesini söylemek için bir yayınlayıcıdan ya da başka bir kuyruk yöneticisinden bir kuyruk yöneticisine gönderilir.

Bu ileti, kuyruk yöneticisinin kuyruğa alınmış yayınlama/abone olma arabirimi tarafından izlenen bir kuyruğa gönderilir.

Giriş kuyruğu, özgün yayının gönderildiği kuyruk olmalıdır.

Delete Publication komut iletisinde belirtilen konuların tümü için değil, bazıları için yetkiniz varsa, yalnızca bu konular silinir. **Broker Response** iletisi, hangi konuların silinmediğini gösterir.

Benzer şekilde, bir **Publish** komutu birden çok konu içeriyorsa, bu konuların bazılarıyla eşleşen ancak tümüyle eşleşmeyen bir **Delete Publication** komutu, yalnızca **Delete Publication** komutunda belirtilen konulara ilişkin yayınları siler.

Kuyruk yöneticisine bir komut iletisi gönderilirken gereken ileti tanımlayıcı (MQMD) değiştirgelerinin ayrıntıları için bkz. [“Kuyruk yöneticisi tarafından iletilen yayınlara ilişkin MQMD ayarları” sayfa 886](#) .

Özellikler

Komut (MQPSC_COMMAND)

Değer: DeletePub (MQPSC_DELETE_PUBLICATION).

Bu özellik belirtilmelidir.

Konu > (MQPSC_TOPIC)

Değer, alıkonan yayınların silineceği bir konuyu içeren bir dizedir. Birden çok konudaki yayınları silmek için dizgiye genel arama karakterleri eklenebilir.

Bu özellik belirtilmelidir; gerektiği kadar konu için yinelenebilir.

DelOpt (MQPSC_DELETE_OPTION)

Silme seçenekleri özelliği aşağıdaki değerlerden birini alabilir:

Yerel (MQPSC_LOCAL)

Belirtilen konulara ilişkin alıkonan tüm yayınlar, Yerel seçeneğiyle yayınlanmış olsun ya da olmasın, yerel kuyruk yöneticisinden (yani, iletinin gönderildiği kuyruk yöneticisinden) silinir.

Diğer kuyruk yöneticilerindeki yayınlar etkilenmez.

Yok (MQPSC_NONE)

Tüm seçenekler varsayılan değerlerini alır. Bu, De10pt özelliğinin atlansıyla aynı etkiye sahiptir. Aynı anda başka seçenekler belirtilirse, Yok yoksayılır.

Bu özellik atlanırsa, varsayılan değer, belirtilen konulara ilişkin alıkonan tüm yayınların, Yere1 seçeneğiyle yayınlanmış olup olmadıklarından bağımsız olarak, ağdaki tüm kuyruk yöneticilerinden silinmesidir.

Örnek

Burada bir **Delete Publication** komut iletisi için NameValueVerileri örneği verilmiştir. Bu, örnek uygulama tarafından yerel kuyruk yöneticisinde, Team1 ile Team2arasındaki eşleşmede en son puanı içeren alıkonan yayını silmek için kullanılır.

```
<psc>
  <Command>DeletePub</Command>
  <Topic>Sport/Soccer/State/LatestScore/Team1 Team2</Topic>
  <De10pt>Local</De10pt>
</psc>
```

Abone Kaydını Kaldır iletisi

Deregister Subscriber komut iletisi, bir abone tarafından ya da bir abone adına başka bir uygulama tarafından, artık belirtilen parametrelerle eşleşen iletileri almak istemediğini belirtmek için bir kuyruk yöneticisine gönderilir.

Bu ileti SYSTEM.BROKER.CONTROL.QUEUE, kuyruk yöneticisinin denetim kuyruğu. Kullanıcı, bu kuyruğa ileti koymak için gerekli yetkiye sahip olmalıdır.

Kuyruk yöneticisine bir komut iletisi gönderilirken gereken ileti tanımlayıcı (MQMD) değiştirgelerinin ayrıntıları için [Kuyruk yöneticisi tarafından iletilen yayınlar için MQMD ayarları](#) konusuna bakın.

Tek bir aboneliğin kaydı, ilgili konu, abonelik noktası ve özgün aboneliğin süzgeç değerleri belirtilerek kaldırılabilir. Özgün abonelikte herhangi bir değer belirtilmediyse (yani varsayılan değerleri aldıysa), abonelik kaydı kaldırıldığında bunlar atlanmalıdır.

DeregAll seçeneği kullanılarak bir aboneye ya da bir abone grubuna ilişkin tüm aboneliklerin kaydı silinebilir. Örneğin, DeregAll bir abonelik noktasıyla (ancak konu ya da süzgeç olmadan) birlikte belirtilirse, konu ve süzgeç dikkate alınmaksızın, belirtilen abonelik noktasında aboneye ilişkin tüm aboneliklerin kaydı silinir. Herhangi bir konu, süzgeç ve abonelik noktası birleşimine izin verilir; üçü de belirtilirse yalnızca bir abonelik eşleşebilir ve DeregAll seçeneği yoksayılır.

İleti, aboneliği kaydeden abone tarafından gönderilmelidir; bu, abonenin kullanıcı kimliği denetlenerek onaylanır.

Aboneliklerin kaydı, MQSC ya da PCF komutları kullanılarak bir sistem denetimcisi tarafından da silinebilir. Ancak, geçici bir dinamik kuyrukla kaydedilen abonelikler, yalnızca kuyruk adıyla değil, kuyrukla ilişkilendirilir. Kuyruk belirttik olarak ya da kuyruk yöneticisiyle bağlantısını kesen uygulama tarafından silinirse, o kuyruğa ilişkin aboneliklerin kaydını iptal etmek için **Deregister Subscriber** komutunu kullanmanız artık mümkün değildir. Aboneliklerin kaydı, geliştirici çalışma ortamı kullanılarak silinebilir ve abonelikte bir sonraki eşleşmesi ya da kuyruk yöneticisi yeniden başlatıldığında kuyruk yöneticisi tarafından otomatik olarak kaldırılır. Olağan koşullarda, uygulamalar kuyruğu silmeden ya da kuyruk yöneticisiyle bağlantıyı kesmeden önce aboneliklerinin kaydını kaldırmalıdır.

Bir abone bir aboneliği kayıttan çıkarmak için bir ileti gönderirse ve bunun başarıyla işlendiğini bildiren bir yanıt iletisi alırsa, aboneliğin kaydı kaldırıldığı sırada kuyruk yöneticisi tarafından işleniyorsa, bazı yayınlar abone kuyruğuna erişmeye devam edebilir. İletiler kuyruktan kaldırılmazsa, abone kuyruğunda işlenmemiş iletiler birikmiş olabilir. Uygulama, bir süre uyuduktan sonra uygun CorrelId ile MQGET çağırısı içeren bir döngü yürütür ve bu iletiler kuyruktan temizlenir.

Benzer şekilde, abone kalıcı bir dinamik kuyruk kullanıyorsa ve kaydı kaldırıp MQCLOSE çağrısında MQCO_DELETE_PURGE seçeneğiyle kuyruğu kapatırsa, kuyruk boş olmayabilir. Kuyruk silindiğinde kuyruk yöneticisinden herhangi bir yayın henüz kesinleştirilmediyse, MQCLOSE çağrısı tarafından bir MQRC_Q_NOT_EMPTY dönüş kodu yayınlanır. Uygulama, zaman zaman MQCLOSE çağrısına uyuyarak ve yeniden girerek bu sorunu önleyebilir.

Özellikler

Komut (MQPSC_COMMAND)

Değer: DeregSub (MQPSC_DEREGISTER_SUBSCRIBER).

Bu özellik belirtilmelidir.

Konu (MQPSC_TOPIC)

Değer, kayıttan kaldırılacak konuyu içeren bir dizedir.

Birden çok konunun kaydı kaldırılacaksa, isteğe bağlı olarak bu özellik yinelenebilir. <RegOpt> içinde DeregAll belirtilirse bu atlanabilir.

Abonenin diğer konulara ilişkin abonelikleri korumak istemesi durumunda, belirtilen konular kayıtlı olanların bir alt kümesi olabilir. Genel arama karakterlerine izin verilir, ancak genel arama karakterleri içeren bir konu dizgisi, **Deregister Subscriber** komut iletilmesinde belirtilen ilgili dizgiyle tam olarak eşleşmelidir.

SubPoint (MQPSC_SUBSCRIPTION_POINT)

Değer, aboneliğin ayrılacağı abonelik noktasını belirten bir dizedir.

Bu özellik yinelenmemelidir. Bir < Topic> belirtilirse ya da <RegOpt> içinde DeregAll belirtilirse bu atlanabilir. Bu özelliği atlarsanız, aşağıdaki durum oluşur:

- **belirtmezseniz**, DeregAll, < Topic> özelliğiyle eşleşen abonelikler (ve varsa, < Süzgeç > özelliği) varsayılan abonelik noktasından kayıttan çıkar.
- DeregAll belirtirseniz, tüm aboneliklerin (varsa, < Konular> ve < Süzgeç > özellikleriyle eşleşen) kaydı tüm abonelik noktalarından kaldırılır.

Varsayılan abonelik noktasını belirttik olarak belirtmeyeceğinizi unutmayın. Bu nedenle, yalnızca bu abonelik noktasından tüm aboneliklerin kaydını kaldırma yolu yoktur; konuları belirtmeniz gerekir.

SubIdentity (MQPSC_SUBSCRIPTION_IDENTITY)

Bu, uzunluk üst sınırı 64 karakter olan değişken uzunluklu bir dizedir. Bir uygulamayı bir abonelik ile ilgilendiren bir şekilde temsil etmek için kullanılır. Kuyruk yöneticisi, her abonelik için bir abone kimlikleri kümesi sağlar. Her abonelik, kimliğinin yalnızca tek bir kimliği ya da sınırsız sayıda kimliği bulundurmasına izin verebilir.

SubIdentity abonelik için kimlik kümesinde yer alıyorsa, kümeden kaldırılır. Bunun sonucu olarak kimlik kümesi boş olursa, RegOpt özelliğinin değeri olarak LeaveOnly belirtilmedikçe, abonelik kuyruk yöneticisinden kaldırılır. Kimlik kümesi yine de başka kimlikler içeriyorsa, abonelik kuyruk yöneticisinden kaldırılamaz ve yayın akışı kesintiye uğramaz.

SubIdentity belirtilirse, ancak SubIdentity abonelik için ayarlanan kimlikte değilse, **Deregister Subscriber** komutu MQRC_CF_SUB_IDENTITY_ERROR dönüş koduyla başarısız olur.

Süzgeç (MQPSC_FILTER)

Değer, kayıttan kaldırılacak süzgeci belirten bir dizedir. Büyük/küçük harf ve boşluklar da içinde olmak üzere, daha önce kaydedilmiş bir abonelik süzgeciyle tam olarak eşleşmelidir.

Birden çok süzgecin kaydı kaldırılacaksa, isteğe bağlı olarak bu özellik yinelenabilir. Bir < Topic> belirtilirse ya da <RegOpt> içinde DeregAll belirtilirse bu atlanabilir.

Belirtilen süzgeçler, abone diğer süzgeçler için abonelikleri korumak isterse, kayıtlı süzgeçlerin bir alt kümesi olabilir.

RegOpt (MQPSC_REGISTRATION_OPTION)

Kayıt seçenekleri özelliği aşağıdaki değerleri alabilir:

DeregAll

(MQPSC_DEREGISTER_ALL)

Bu abone için kaydedilen tüm eşleşen aboneliklerin kaydı silinecek.

DeregAllseçeneğini belirlerseniz:

- < Konular>, <SubPoint>ve < Süzgeç > atlanabilir.
- Gerekirse, < Konular> ve < Süzgeç > yinelenebilir.
- <SubPoint> yinelenmemelidir.

belirtmezseniz DeregAll:

- < Koni> belirtilmelidir ve gerekirse yinelenebilir.
- <SubPoint> ve < Süzgeç > atlanabilir.
- <SubPoint> yinelenmemelidir.
- Gerekirse, < Süzgeç > yinelenabilir.

Konular ve süzgeçlerin her ikisi de yinelenirse, ikisinin tüm birleşimleriyle eşleşen tüm abonelikler kaldırılır. Örneğin, üç konu ve üç süzgeç belirten bir **Deregister Subscriber** komutu dokuz aboneliği kaldırmayı dener.

CorrelAsTanıtıcısı

(MQPSC_CORREL_ID_AS_IDENTITY)

Sfır olmaması gereken ileti tanımlayıcısındaki (MQMD) CorrelId aboneyi tanımlamak için kullanılır. Özgün abonelikte kullanılan CorrelId ile eşleşmelidir.

FullResp

(MQPSC_FULL_RESPONSE)

FullResp belirtildiğinde, komut başarısız olmazsa, yanıt iletisinde aboneliğin tüm öznitelikleri döndürülür.

FullResp belirtildiğinde DeregAll ögesine **Deregister Subscriber** komutunda izin verilmez. Birden çok konu da belirtilamaz. Komut, her iki durumda da *MQRCCF_REG_OPTIONS_ERROR*dönüş koduyla başarısız olur.

LeaveOnly

(YALNIZCA MQPSC_LEAVE_ONLY)

Bunu, abonelik için kimlik kümesinde bulunan bir SubIdentity ile belirttiğinizde SubIdentity , abonelik için kimlik kümesinden kaldırılır. Sonuç kimlik kümesi boş olsa da, abonelik kuyruk yöneticisinden kaldırılmaz. SubIdentity değeri kimlik kümesinde yoksa, komut *MQRCCF_SUB_IDENTITY_ERROR*dönüş koduyla başarısız olur.

LeaveOnly SubIdentity olmadan belirtilirse, komut *MQRCCF_REG_OPTIONS_ERROR*dönüş koduyla başarısız olur.

LeaveOnly ya da SubIdentity belirtilmezse, abonelik, abonelik için ayarlanan kimlik içeriğinden bağımsız olarak kaldırılır.

Yok

(MQPSC_NONE)

Tüm seçenekler varsayılan değerlerini alır. Bu, kayıt seçenekleri özelliğinin atlansıyla aynı etkiye sahiptir. Aynı anda başka seçenekler belirtilirse, Yok yoksayılar.

VariableUserKimliği

(MQPSC_VARIABLE_USER_ID)

Belirtildiğinde, abonenin kimliği (kuyruk, kuyruk yöneticisi ve correlid) tek bir kullanıcı kimliğiyle sınırlı değildir. Bu, özgün kayıt iletisinin kullanıcı kimliğini abonenin kimliğiyle ilişkilendiren kuyruk yöneticisinin var olan davranışından farklıdır ve bundan sonra bu kimliği

kullanan başka bir kullanıcıyı engeller. Yeni bir abone aynı kimliği kullanmayı denerse, *MQRCCF_DUPLICATE_SUBSCRIPTION* dönüş kodu döndürülür.

Herhangi bir kullanıcı, uygun yetkiye sahip olduğunda aboneliği değiştirebilir ya da kayıttan çıkartabilir; kullanıcı kimliğinin özgün abonenin yetkisiyle eşleşmesi gerektiğinden, var olan denetimin önlenmesi gerekir.

Bu seçeneği var olan bir aboneliğe eklemek için, komutun özgün abonelikte aynı kullanıcı kimliğinden gelmesi gerekir.

Kayıttan kaldırılacak aboneliğin *VariableUserId* ayarı varsa, hangi aboneliğin kaydının kaldırılacağını belirtmek için kayıt kaldırma sırasında bu ayarlanmalıdır. Tersi durumda, aboneliği tanımlamak için **Deregister Subscriber** komutunun kullanıcı kimliği kullanılır. Bir abonelik adı sağlanırsa, bu, diğer abone tanıtıcılarıyla birlikte geçersiz kılınır.

Varsayılan değer, bu özellik atlanırsa, hiçbir kayıt seçeneği ayarlanmamaktır.

QMgrName (MQPSC_Q_MGR_NAME)

Değer, abone kuyruğunun kuyruk yöneticisi adıdır. Özgün abonelikte kullanılan *QMgrName* ile eşleşmelidir.

Bu özellik atlanırsa, varsayılan değer, ileti tanımlayıcısındaki (MQMD) *ReplyToQMgr* adıdır. Sonuçtaki ad boşsa, varsayılan olarak kuyruk yöneticisinin adını kullanır.

QName (MQPSC_Q_NAME)

Değer, abone kuyruğunun adıdır. Özgün abonelikte kullanılan *QName* ile eşleşmelidir.

Bu özellik atlanırsa, varsayılan değer, boş olmaması gereken ileti tanımlayıcısındaki (MQMD) *ReplyToQ* adıdır.

SubName (MQPSC_SUBSCRIPTION_NAME)

Bir **Deregister Subscriber** komutunda *SubName* belirtirseniz, aboneliğin kendisinde *VariableUserId* ayarlanmadıkça, *SubName* değeri kullanıcı kimliği dışındaki diğer tüm tanıtıcı alanlarından önceliklidir. *VariableUserId* ayarlanmazsa, **Deregister Subscriber** komutu ancak komut iletisinin kullanıcı kimliği abonelikte eşleşirse başarılı olur; eşleşmezse, komut *MQRCCF_DUPLICATE_IDENTITY* dönüş koduyla başarısız olur.

Bu komutun geleneksel kimliğiyle eşleşen ancak *SubName* olmayan bir abonelik varsa, **Deregister Subscriber** komutu *MQRCCF_SUB_NAME_ERROR* dönüş koduyla başarısız olur. Geleneksel kimlikle eşleşen, ancak *SubName* olmayan bir komut iletisi kullanılarak *SubName* olan bir aboneliğin kaydını kaldırma girişiminde bulunulursa, komut başarılı olur.

SubUserVerileri (MQPSC_SUBSCRIPTION_USER_DATA)

Bu, değişken uzunluklu bir metin dizgisidir. Değer, kuyruk yöneticisi tarafından abonelikte birlikte saklanır, ancak yayının aboneye teslim edilmesi üzerinde herhangi bir etkisi yoktur. Değer, aynı aboneliğe yeni bir değerle yeniden kaydedilerek değiştirilebilir. Bu öznelik, uygulamanın kullanımı içindir.

SubUserVerileri, *SubUserVerileri* varsa, bir aboneliğe ilişkin Meta Konu bilgilerinde (*MQCACF_REG_SUB_USER_DATA*) döndürülür.

Örnek

Burada bir **Deregister Subscriber** komut iletisi için *NameValueVerileri* örneği verilmiştir. Bu örnekte, örnek uygulama tüm eşleşmeler için en son puanı içeren konulara aboneliğinin kaydını kaldırıyor. *CorrelId*de içinde olmak üzere abonenin kimliği, MQMD 'deki varsayılan değerlerden alınır.

```
<psc>
  <Command>DeregSub</Command>
  <RegOpt>CorrelAsId</RegOpt>
  <Topic>Sport/Soccer/State/LatestScore/#</Topic>
</psc>
```


İleti yayınla

Publish komut iletisi, belirli bir konuya ya da konulara ilişkin bilgileri yayınlamak için kuyruğa ya da kuyruk yöneticisinden aboneye yerleştirilir.

Bir iletiyi kuyruğa koyma yetkisi ve belirli bir konu ya da konu hakkındaki bilgileri yayınlama yetkisi gereklidir.

Kullanıcının bazı konularda bilgi yayınlama yetkisi varsa, ancak tümü değilse, yayınlamak için yalnızca bu konular kullanılır; uyarı yanıtı, yayınlamak için kullanılmayan konuları gösterir.

Bir abonenin eşleşen abonelikleri varsa, kuyruk yöneticisi **Publish** iletisini ilgili **Register Subscriber** komut iletilerinde tanımlanan abone kuyruklarına iletir.

Kuyruk yöneticisine bir komut iletisi gönderilirken gereken ileti tanımlayıcı (MQMD) parametrelerinin ayrıntıları için [Kuyruk Yöneticisi Yanıt iletisi](#) konusuna bakın ve kuyruk yöneticisi bir yayını aboneye iletildiğinde kullanılır.

Kuyruk yöneticisi, **Publish** iletisini, yerel bir yayın değilse, ağdaki eşleşen abonelikleri olan diğer kuyruk yöneticilerine iletir.

Varsa, yayın verileri iletinin gövdesine eklenir. Veriler, MQRFH2 üstbilgisinin NameValueData alanındaki bir <mcd> klasöründe açıklanabilir.

Özellikler

Komut (**MQPSC_COMMAND**)

Değer: Yayınla (**MQPSC_PUBLISH**).

Bu özellik belirtilmelidir.

Konu (**MQPSC_TOPIC**)

Değer, bu yayını sınıflandıran bir konuyu içeren bir dizedir. Genel arama karakterlerine izin verilmez.

Konuyu SYSTEM.QPUBSUB.QUEUE.NAMELIST, bu görevin nasıl tamamlanacağına ilişkin yönergeler için [Akış ekleme](#) başlıklı konuya bakın.

Bu özellik belirtilmelidir ve isteğe bağlı olarak, gereken sayıda konu için yinelenebilir.

SubPoint (**MQPSC_SUBSCRIPTION_POINT**)

Yayının yayınlandığı abonelik noktası.

WebSphere Event Broker 6.0 içinde, <SubPoint> özelliğinin değeri, yayınlamayı işleyen Yayın düğümünün Abonelik Noktası özneliğinin değeridir.

IBM WebSphere MQ 7.0.1 içinde, <SubPoint> özelliğinin değeri, bir abonelik noktasının adıyla eşleşmelidir. Bkz. [Abonelik noktası ekleme](#).

PubOpt (**MQPSC_PUBLICATION_OPTION**)

Yayınlama seçenekleri özelliği aşağıdaki değerleri alabilir:

RetainPub

(**MQPSC_RETAIN_PUB**)

Kuyruk yöneticisi, yayının bir kopyasını alıkoymaktır. Bu seçenek belirlenmezse, kuyruk yöneticisi yayını yürürlükteki tüm abonelerine gönderir göndermez yayın silinir.

IsRetainedPub

(**MQPSC_IS_RETAINED_PUB**)

(Yalnızca bir kuyruk yöneticisi tarafından ayarlanabilir.) Bu yayın kuyruk yöneticisi tarafından korundu. Kuyruk yöneticisi, aboneliğin InformIfRemedi seçeneğine kaydedilmesi koşuluyla, bir aboneye bu yayının daha önce yayınlandığını ve alıkondüğünü bildirmek için bu seçeneği ayarlar. Yalnızca Aboneyi Kaydet ya da Güncellemeyi İste komut iletisine yanıt olarak ayarlanır. Doğrudan abonelere gönderilen alıkonan yayınlarda bu seçenek ayarlanmaz.

Yerel

(MQPSC_LOCAL)

Bu seçenek, kuyruk yöneticisine bu yayının diğer kuyruk yöneticilerine gönderilmemesi gerektiğini bildirir. Bu kuyruk yöneticisinde kayıtlı olan tüm aboneler, eşleşen abonelikleri varsa bu yayını alır.

OtherSubs

(MQPSC_OTHER_SUBS_ONLY)

Bu seçenek, bir yayıncının aynı konuya abone olduğu konferans tipi uygulamaların daha kolay işlenmesini sağlar. Kuyruk yöneticisine, eşleşen bir aboneliği olsa bile yayını yayıncının abone kuyruğuna göndermemesini söyler. Yayıncının abone kuyruğu, aşağıdaki listede açıklandığı gibi QMgrName, QNameve isteğe bağlı CorrelIdöğesinden oluşur.

CorrelAsTanıtıcısı

(MQPSC_CORREL_ID_AS_IDENTITY)

MQMD 'deki CorrelId (sıfır olmamalıdır), yayıncının abone olduğu uygulamalarda yayıncının abone kuyruğunun bir parçasıdır.

Yok

(MQPSC_NONE)

Tüm seçenekler varsayılan değerlerini alır. Bu, yayın seçenekleri özelliğinin atlanmasıyla aynı etkiye sahiptir. Aynı anda başka seçenekler belirtilirse, Yok yoksayılr.

Ek <PubOpt> öğeleri sunarak birden çok yayınlama seçeneğiniz olabilir.

Bu " zellik atlanırsa, varsayılan değer, yayın seçeneklerinin belirlenmesidir.

PubTime (MQPSC_PUBLISH_TIMESTAMP)

Değer, yayıncı tarafından ayarlanan isteğe bağlı bir yayın zaman damgasıdır. 16 karakter uzunluğunda ve biçimi:

```
YYYYMMDDHHMSSSTH
```

Evrensel saati kullanarak. Bu bilgiler, abonelere gönderilmeden önce kuyruk yöneticisi tarafından denetlenmez.

SeqNum (MQPSC_SEQUENCE_NUMBER)

Değer, yayıncı tarafından ayarlanan isteğe bağlı bir sıra numarasıdır.

Her yayınlama birlikte 1 artırılmalıdır. Ancak bu, yalnızca bu bilgileri abonelere iletebilen kuyruk yöneticisi tarafından denetlenmez.

Aynı konudaki yayınlar birbiriyle bağlantılı farklı kuyruk yöneticilerine yayınlandıysa, sıra numaralarının anlamlı olmasını sağlamak yayıncıların sorumluluğundadır.

QMgrName (MQPSC_Q_MGR_NAME)

Değer, yayıncının abone olduğu uygulamalarda, yayıncının abone kuyruğuna ilişkin kuyruk yöneticisinin adını içeren bir dizedir (bkz. OtherSubsOnly).

Bu özellik atlanırsa, varsayılan değer, ileti tanımlayıcısındaki (MQMD) ReplyToQMgr adıdır. Sonuçtaki ad boşsa, varsayılan olarak kuyruk yöneticisinin adını kullanır.

QName (MQPSC_Q_NAME)

Değer, yayıncının aynı zamanda abone olduğu uygulamalarda yayıncının abone kuyruğunun adını içeren bir dizedir (bkz. OtherSubsOnly).

Bu özellik atlanırsa, varsayılan değer, ileti tanımlayıcısındaki (MQMD) ReplyToQ adıdır; OtherSubsOnly ayarlandıysa bu ad boş olmamalıdır.

Örnek

Aşağıda, bir **Publish** komut iletisi için *NameValueVerileri* ile ilgili bazı örnekler verilmiştir.

İlk örnek, örnek uygulamadaki eşleşme benzetimcisi tarafından bir eşleşmenin başladığını belirtmek için gönderilen bir yayının örneğidir.

```
<psc>
  <Command>Publish</Command>
  <Topic>Sport/Soccer/Event/MatchStarted</Topic>
</psc>
```

İkinci örnek, korunan bir yayın içindir. Team1 ile Team2 arasındaki maçtaki en son puan yayınlanır.

```
<psc>
  <Command>Publish</Command>
  <PubOpt>RetainPub</PubOpt>
  <Topic>Sport/Soccer/State/LatestScore/Team1 Team2</Topic>
</psc>
```

Abone iletisini kaydet

Register Subscriber komut iletisi, bir abonelik noktasında bir ya da daha fazla konuya abone olmak istediğini belirtmek için bir abone ya da abone adına başka bir uygulama tarafından bir kuyruk yöneticisine gönderilir. Bir ileti içeriği süzgeci de belirtilebilir.

Yayınlama/abone olma süzgeç ifadelerinde, iç içe yerleştirme parantezleri performansın üstel olarak azalmasına neden olur. Parantezleri yaklaşık 6 'dan büyük bir derinliğe yerleştirmekten kaçının.

İleti SYSTEM.BROKER.CONTROL.QUEUE. Abonelikteki konu ya da konulara ilişkin erişim yetkisinin (kuyruk yöneticisinin sistem denetimcisi tarafından ayarlanan) yanı sıra, bu kuyruğa bir ileti koyma yetkisi de gereklidir.

Kullanıcının bazı konularda yetkisi varsa, ancak tümü değilse, yalnızca yetkili olanlar kaydedilir; bir uyarı yanıtı kayıtlı olmayanları gösterir.

Kuyruk yöneticisine bir komut iletisi gönderilirken gereken ileti tanımlayıcı (MQMD) değiştirgelerinin ayrıntıları için bkz. [“Kuyruk yöneticisine komut iletilerinde MQMD ayarları” sayfa 886](#) .

Kuyruğa yanıt geçici bir dinamik kuyruksa, kuyruk kapatıldığında kuyruk yöneticisi aboneliğin kaydını otomatik olarak geçersiz kılar.

Özellikler

Komut (**MQPSC_COMMAND**)

Değer: RegSub (**MQPSC_REGISTER_SUBSCRIBER**). Bu özellik belirtilmelidir.

Konu (**MQPSC_TOPIC**)

Abonenin yayınları almak istediği konu. Konunun bir parçası olarak genel arama karakterleri belirtilebilir.

Bu şekilde yaratılan aboneliği incelemek için **display sub** MQSC komutunu kullanırsanız, < Konu> etiketinin değeri, aboneliğin TOPICSTR özelliği olarak gösterilir.

Bu özellik gereklidir ve isteğe bağlı olarak gerektiği kadar konu için yinelenebilir.

SubPoint (**MQPSC_SUBSCRIPTION_POINT**)

Değer, aboneliğin eklendiği abonelik noktasıdır.

Bu özellik atlanırsa, varsayılan abonelik noktası kullanılır.

WebSphere Event Broker 6.0.içinde, <SubPoint> özelliğinin değeri, abone olunan Yayın düğümlerinin Abonelik Noktası özniteliğinin değeriyle eşleşmelidir.

IBM WebSphere MQ 7.0.1.içinde, <SubPoint> özelliğinin değeri, bir abonelik noktasının adıyla eşleşmelidir. Bkz. [Abonelik noktası ekleme](#).

Süzgeç (MQPSC_FILTER)

Değer, yayın iletilerinin içeriğine süzgeç olarak kullanılan bir SQL ifadesidir. Belirtilen konudaki bir yayın süzgeçle eşleşirse, aboneye gönderilir. Bu özellik, MQSUB ve MQOPEN çağrılarında kullanılan Seçim Dizgisine karşılık gelir. Daha fazla bilgi için [İletinin içeriğini seçme](#) başlıklı konuya bakın.

Bu özellik atlanırsa, içerik süzme işlemi gerçekleşmez.

RegOpt (MQPSC_REGISTRATION_OPTION)

Bu Kayıt Seçenekleri özelliği aşağıdaki değerleri alabilir:

AddName

(MQPSC_ADD_NAME)

Bu Aboneliği Kaydet komutunun geleneksel kimliğiyle eşleşen, ancak geçerli bir SubName değeri olmadan var olan bir abonelik için belirtildiğinde, bu komutta belirtilen SubName aboneliğe eklenir.

AddName belirtilirse SubName alanı zorunludur, tersi durumda MQRCCF_REG_OPTIONS_ERROR döndürülür.

CorrelAsTanıtıcısı

(MQPSC_CORREL_ID_AS_IDENTITY)

Eşleşen yayınları abone kuyruğuna gönderirken, ileti tanımlayıcıdaki (MQMD) CorrelId kullanılır. CorrelId sıfır olmamalıdır,

FullResp

(MQPSC_FULL_RESPONSE)

Belirtildiğinde, komut başarısız olmazsa yanıt iletilerinde aboneliğin tüm öznitelikleri döndürülür.

FullResp yalnızca komut iletileri tek bir aboneliğe başvurduğunda geçerlidir. Bu nedenle, komutta tek bir konuya izin verilir; tersi durumda, komut MQRCCF_REG_OPTIONS_ERROR dönüş koduyla başarısız olur.

InformIfRet

(MQPSC_INFORM_IF_ALIKONDU)

Kuyruk yöneticisi, bir yayın **Register Subscriber** ya da **Request Update** komut iletilerine yanıt olarak bir Yayınla iletileri gönderdiğinde alıkonursa aboneyi bilgilendirir. Kuyruk yöneticisi bunu, iletiye IsRetainedPub yayınlama seçeneğini ekleyerek yapar.

JoinExcl

(MQPSC_JOIN_EXCLUSIVE)

Bu seçenek, belirtilen SubIdentity ögesinin aboneliğe ilişkin kimlik kümesinin özel üyesi olarak eklenmesi gerektiğini ve kümeye başka kimliklerin eklenemeyeceğini belirtir.

Kimlik zaten 'paylaşılan' girişine katıldıysa ve kümedeki tek giriş ise, küme bu kimlik tarafından tutulan özel bir kilit olarak değiştirilir. Ters durumda, aboneliğin kimlik kümesinde (paylaşılan erişimle) başka kimlikleri varsa, komut MQRCCF_SUBSCRIPTION_IN_USE dönüş koduyla başarısız olur.

JoinShared

(MQPSC_JOIN_SHARED)

Bu seçenek, belirtilen SubIdentity ögesinin abonelik için kimlik kümesine eklenmesi gerektiğini belirtir.

Abonelik şu anda dışlayıcı olarak kilitliyse (JoinExcl seçeneği kullanılarak), aboneliği kilitli olan kimlik bu komut iletilerindekiyle aynı değilse, komut MQRCCF_SUBSCRIPTION_LOCKED dönüş koduyla başarısız olur. Bu durumda kilit otomatik olarak paylaşılan bir kilide değiştirilir.

Yerel

(MQPSC_LOCAL)

Abonelik yerel ve ağdaki diğer kuyruk yöneticilerine dağıtılmıyor. Diğer kuyruk yöneticilerinden yapılan yayınlar, ilgili genel aboneliği olmadığı sürece bu aboneye teslim edilmez.

NewPubs

(MQPSC_NEW_PUBS_ONLY)

Abonelik kaydedildiği sırada var olan alıkonan yayınlar aboneye gönderilmez; yalnızca yeni yayınlar gönderilir.

Bir abone bu seçeneği yeniden kaydeder ve artık ayarlanmayacak şekilde değiştirirse, önceden gönderilmiş olan bir yayın yeniden gönderilebilir.

NoAlter

(MQPSC_NO_ALTER)

Var olan bir eşleşen aboneliğin öznelikleri değiştirilmez.

Bir abonelik oluşturulurken bu seçenek yoksayılr. Belirtilen diğer tüm seçenekler yeni abonelik için geçerlidir.

SubIdentity birleştirme seçeneklerinden birine de sahipse (JoinExcl ya da JoinShared) Belirtilen kimlik, NoAlter belirtilip belirtilmediğine bakılmaksızın kimlik değiştirme kümesine eklenir.

Yok

(MQPSC_NONE)

Tüm kayıt seçenekleri varsayılan değerlerini alır.

Abone zaten kayıtlıysa, seçenekleri varsayılan değerlerine sıfırlanır (bunun kayıt seçenekleri özelliğini atlama ile aynı etkiye sahip olmadığını göz önünde bulundurun) ve abonelik süre bitimi **Register Subscriber** iletisinin MQMD ' den güncellenir.

Aynı anda başka kayıt seçenekleri belirtilirse, Yok yoksayılr.

NonPers

(MQPSC_NON_PERSISTENT)

Bu abonelikte eşleşen yayınlar, aboneye kalıcı olmayan iletiler olarak teslim edilir.

Müşteriler

(MQPSC_PERSISTENT)

Bu abonelikte eşleşen yayınlar, aboneye kalıcı ileti olarak teslim edilir.

PersAsPub

(MQPSC_PERSISTENT_AS_PUBLISH)

Bu abonelikte eşleşen yayınlar, yayıncı tarafından belirtilen kalıcılıkla aboneye teslim edilir. Bu varsayılan davranıştır.

PersAsKuyruk

(MQPSC_PERSISTENT_AS_Q)

Bu abonelikte eşleşen yayınlar, abone kuyruğunda belirtilen kalıcılıkla aboneye teslim edilir.

PubOnReqOnly

(YALNIZCA MQPSC_PUB_ON_REQUEST_ONLY)

Kuyruk yöneticisi, **Request Update** komut iletisinin yanıtı dışında, aboneye yayın göndermez.

VariableUserKimliği

(MQPSC_VARIABLE_USER_ID)

Belirtildiğinde, abonenin kimliği (kuyruk, kuyruk yöneticisi ve correlid) tek bir kullanıcı kimliğiyle sınırlı değildir. Bu, özgün kayıt iletisinin kullanıcı kimliğini abonenin kimliğiyle ilişkilendiren kuyruk yöneticisinin var olan davranışından farklıdır ve bundan sonra bu kimliği kullanan başka bir kullanıcıyı engeller. Yeni bir abone aynı kimliği kullanmaya çalışırsa MQRCCF_DUPLICATE_SUBSCRIPTION döndürülür.

Bu, kullanıcının uygun yetkisi varsa, herhangi bir kullanıcının aboneliği değiştirmesini ya da aboneliğin kaydını kaldırılmasını sağlar. Bu nedenle, kullanıcı kimliğinin özgün abonenin kimliğiyle eşleştiğini denetlemeniz gerekmez.

Bu seçeneği var olan bir aboneliğe eklemek için, komutun özgün abonelikte aynı kullanıcı kimliğinden gelmesi gerekir.

Request Update komutunun aboneliğinde `VariableUserkimliği` ayarlandıysa, bu, istek güncelleme sırasında hangi aboneliğe başvurulacağı belirtilecek şekilde ayarlanmalıdır. Ters durumda, aboneliği tanımlamak için **Request Update** komutunun kullanıcı kimliği kullanılır. Bir abonelik adı sağlanırsa, bu, diğer abone tanıtıcılarıyla birlikte geçersiz kılınır.

Bu seçenek kümesi olmayan bir **Register Subscriber** komut iletisi, bu seçeneğin ayarlandığı var olan bir aboneliğe başvurursa, bu seçenek bu abonelikten kaldırılır ve aboneliğin kullanıcı kimliği düzeltilmiştir. Aynı kimliğe (kuyruk, kuyruk yöneticisi ve ilinti tanıtıcısı) sahip, ancak ilişkilendirilmiş farklı bir kullanıcı kimliğine sahip bir abone varsa, `MQRCCF_DUPLICATE_IDENTITY` dönüş koduyla komut başarısız olur; abone kimliğiyle ilişkilendirilmiş tek bir kullanıcı kimliği olabilir.

Kayıt seçenekleri özelliği atlanırsa ve abone zaten kayıtlıysa, kayıt seçenekleri değiştirilmez ve abonelik süre bitimi **Register Subscriber** iletisinin `MQMD` 'sinden güncellenir.

Abone önceden kayıtlı değilse, tüm kayıt seçenekleri varsayılan değerlerini alarak yeni bir abonelik oluşturulur.

Varsayılan değerler `PersAsPub` 'dır ve başka seçenek ayarlanmaz.

QMgrName (MQPSC_Q_MGR_NAME)

Değer, kuyruk yöneticisi tarafından eşleşen yayınların gönderildiği abone kuyruğuna ilişkin kuyruk yöneticisinin adıdır.

Bu özellik atlanırsa, varsayılan değer, ileti tanımlayıcısındaki (MQMD) `ReplyToQMgr` adıdır. Sonuçtaki ad boşsa, varsayılan olarak kuyruk yöneticisinin `QMgrName` değerine ayarlanır.

QName (MQPSC_Q_NAME)

Değer, kuyruk yöneticisi tarafından eşleşen yayınların gönderildiği abone kuyruğunun adıdır.

Bu özellik atlanırsa, varsayılan değer, bu durumda boş olmaması gereken ileti tanımlayıcısındaki (MQMD) `ReplyToQ` adıdır.

Kuyruk geçici bir dinamik kuyruksa, yayınların kalıcı olmayan teslimi (`NonPers`) `<RegOpt>` özelliğinde belirtilmelidir.

Kuyruk geçici bir dinamik kuyruksa, kuyruk kapatıldığında kuyruk yöneticisi tarafından abonelik kaydı otomatik olarak kaldırılır.

SubName (MQPSC_SUBSCRIPTION_NAME)

Bu, belirli bir aboneliğe verilen bir addır. Bir aboneliğe gönderme yapmak için kuyruk yöneticisi, kuyruk ve isteğe bağlı `correlId` yerine bunu kullanabilirsiniz.

Bu **SubName** ile bir abonelik zaten varsa, aboneliğin diğer öznitelikleri (Konu, `QMgrName`, `QName`, `CorrelId`, `UserId`, `RegOpts`, `UserSubVerileri` ve Süre Bitimi) yeni `Aboneyi` `Kaydet` komut iletisinde geçirilen özniteliklerle geçersiz kılınır. Ancak, **SubName** `QName` alanı belirtilmeden kullanılırsa ve MQMD üstbilgisinde bir `ReplyToQ` belirtilirse, abone kuyruğu `ReplyToQ` olacak şekilde değiştirilir.

Bu komutun geleneksel kimliğiyle eşleşen bir abonelik zaten varsa, ancak **SubName** yoksa, **AddName** seçeneği belirtilmedikçe, Kayıt komutu `MQRCCF_DUPLICATE_SUBSCRIPTION` dönüş koduyla başarısız olur.

Aynı **SubName** 'nı belirten başka bir `Aboneyi` `Kaydet` komutunu ve yeni komuttaki `QMgrName`, `QName` ve `CorrelId` Konu değerlerini kullanarak var olan bir aboneliği değiştirmeye çalışırsanız, komut `MQRCCF_DUPLICATE_SUBSCRIPTION` dönüş koduyla başarısız olur. Bu, aynı aboneliğe başvuran iki abonelik adını önler.

SubIdentity (MQPSC_SUBSCRIPTION_IDENTITY)

Bu dizgi, abonelik ile ilgili bir uygulamayı göstermek için kullanılır. Uzunluk üst sınırı 64 karakter olan değişken uzunluklu bir karakter dizgisidir ve isteğe bağlıdır. Kuyruk yöneticisi, her abonelik için bir abone kimlikleri kümesi sağlar. Her abonelik, kimlik kümesinin yalnızca bir kimlik ya da sınırsız sayıda kimlik içermesine izin verebilir (bkz. **JoinShared** ve **JoinExcl** seçenekleri).

JoinShared ya da **JoinExcl** seçeneğini belirten bir abonelik komutu, **SubIdentity** ögesini aboneliğin kimlik kümesine ekler (önceden yoksa) ve var olan kimlik kümesi bu tür bir işleme izin veriyorsa; başka bir abone özel olarak katılmadıysa ya da kimlik kümesi boşsa.

SubIdentity 'nin belirtildiği bir Abone Kaydet komutunun sonucu olarak aboneliğin özniteliklerinde yapılan herhangi bir değişiklik, yalnızca bu abonelik için kimlik kümesinin tek üyesi olması durumunda başarılı olur. Ters durumda, komut **MQRCCF_SUBSCRIPTION_IN_USED** dönüş koduyla başarısız olur. Bu, ilgili diğer abonelerin haberi olmadan bir aboneliğin özniteliklerinin değişmesini önler.

64 karakterden uzun bir karakter dizgisi belirtirseniz, komut **MQRCCF_SUB_IDENTITY_ERROR** dönüş koduyla başarısız olur.

SubUserVerileri (MQPSC_SUBSCRIPTION_USER_DATA)

Bu, değişken uzunluklu bir metin dizgisidir. Değer, abonelik ile kuyruk yöneticisi tarafından saklanır, ancak aboneye yayın teslimi üzerinde herhangi bir etkisi yoktur. Değer, aynı aboneliğe yeni bir değerle yeniden kaydedilerek değiştirilebilir. Bu öznitelik, uygulamanın kullanımı için kullanılır.

SubUserVerileri, varsa bir aboneliğe ilişkin Meta Konu bilgilerinde (**MQCACF_REG_SUB_USER_DATA**) döndürülür.

NonPers, **PersAsPub**, **PersAsQueue**, and **PersKayıt** seçeneği değerlerinden birden fazlasını belirtirseniz, yalnızca sonuncusu kullanılır. Bu seçenekleri tek bir abonelikte birleştiremezsiniz.

Örnek

Burada bir **Register Subscriber** komut iletisi için **NameValueVerileri** örneği verilmiştir. Örnek uygulamada, sonuçlar hizmeti tüm eşleşmelerin en son puanlarını içeren konulara abonelik kaydı için 'Yayınlama olarak kalıcı' seçeneği ayarlanmış olarak bu iletiyi kullanır. **CorrelIdde** içinde olmak üzere abonenin kimliği, **MQMD** 'deki varsayılan değerlerden alınır.

```
<psc>
  <Command>RegSub</Command>
  <RegOpt>PersAsPub</RegOpt>
  <RegOpt>CorrelAsId</RegOpt>
  <Topic>Sport/Soccer/State/LatestScore/#</Topic>
</psc>
```

Güncelleme İste iletisi

Request Update komut iletisi, belirtilen konu ve belirtilen (isteğe bağlı) süzgeçle eşleşen abonelik noktası için yürürlükteki alıkonan yayınları istemek üzere bir aboneden kuyruk yöneticisine gönderilir.

Bu ileti **SYSTEM.BROKER.CONTROL.QUEUE**, kuyruk yöneticisinin denetim kuyruğu. İstek güncellemesindeki konuya ilişkin erişim yetkisinin yanı sıra, bu kuyruğa bir ileti koyma yetkisi de gereklidir; bu, kuyruk yöneticisinin sistem denetimcisi tarafından belirlenir.

Bu komut, abonenin kaydolduğunda **PubOnReqOnly** seçeneğini belirtmesi durumunda kullanılır. Kuyruk yöneticisinde eşleşen alıkonan yayınlar varsa, bunlar aboneye gönderilir. Kuyruk yöneticisinde eşleşen alıkonan yayın yoksa, istek **MQRCCF_NO_RETAINED_MSG** dönüş koduyla başarısız olur. İstekte bulunanın önceden aynı Konu, **SubPoint** ve Süzgeç değerlerine sahip bir aboneliği kaydetmiş olması gerekir.

Özellikler

Komut (MQPSC_COMMAND)

Değer: **ReqUpdate** (**MQPSC_REQUEST_UPDATE**). Bu özellik belirtilmelidir.

Konu (MQPSC_TOPIC)

Değer, abonenin istediği konudur; genel arama karakterlerine izin verilir.

Bu özellik belirtilmelidir, ancak bu iletide tek bir geçişe izin verilir.

SubPoint (MQPSC_SUBSCRIPTION_POINT)

Değer, aboneliğin eklendiği abonelik noktasıdır.

Bu özellik atlanırsa, varsayılan abonelik noktası kullanılır.

Süzgeç (MQPSC_FILTER)

Değer, yayın iletilerinin içeriğine süzgeç olarak kullanılan bir ESQL ifadesidir. Belirtilen konudaki bir yayın süzgeçle eşleşirse, aboneye gönderilir.

< Süzgeç > özelliği, şu anda güncelleme istediğiniz özgün abonelikte belirtilenle aynı değere sahip olmalıdır.

Bu özellik atlanırsa, içerik süzme işlemi gerçekleşmez.

RegOpt (MQPSC_REGISTRATION_OPTION)

Kayıt seçenekleri özelliği aşağıdaki değeri alabilir:

CorrelAsTanıtıcısı

(MQPSC_CORREL_ID_AS_IDENTITY)

Sıfır olmaması gereken ileti tanımlayıcıdaki (MQMD) CorrelId , eşleşen yayınları abone kuyruğuna gönderirken kullanılır.

Yok

(MQPSC_NONE)

Tüm seçenekler varsayılan değerlerini alır. Bu, <RegOpt> özelliğinin atlanmasıyla aynı etkiye sahiptir. Aynı anda başka seçenekler belirtilirse, Yok yoksayılır.

VariableUserKimliği

(MQPSC_VARIABLE_USER_ID)

Belirtildiğinde, abonenin kimliği (kuyruk, kuyruk yöneticisi ve correlid) tek bir kullanıcı kimliğiyle sınırlı değildir. Bu, özgün kayıt iletisinin kullanıcı kimliğini abonenin kimliğiyle ilişkilendiren kuyruk yöneticisinin var olan davranışından farklıdır ve bundan sonra bu kimliği kullanan başka bir kullanıcıyı engeller. Yeni bir abone aynı kimliği kullanmayı denerse, komut MQRCCF_DUPLICATE_SUBSCRIPTION dönüş koduyla başarısız olur.

Bu, herhangi bir kullanıcının uygun yetkiye sahip olduğunda aboneliği değiştirmesine ya da kayıttan çıkarmasına olanak sağlar. Bu nedenle, kullanıcı kimliğinin özgün aboneyle eşleştiğini denetlemeniz gerekmez.

Bu seçeneği var olan bir aboneliğe eklemek için, komutun özgün abonelikte aynı kullanıcı kimliğinden gelmesi gerekir.

Request Update komutunun aboneliğinde VariableUserkimliği ayarlandıysa, bu, istek güncelleme sırasında hangi aboneliğe başvurulacağı belirtilecek şekilde ayarlanmalıdır. Ters durumda, aboneliği tanımlamak için **Request Update** komutunun kullanıcı kimliği kullanılır. Bir abonelik adı sağlanırsa, bu, diğer abone tanıtıcılarıyla birlikte geçersiz kılınır.

Varsayılan değer, bu özellik atlanırsa, hiçbir kayıt seçeneği ayarlanmamaktır.

QMgrName (MQPSC_Q_MGR_NAME)

Değer, eşleşen alıkonan yayının kuyruk yöneticisi tarafından gönderildiği abone kuyruğuna ilişkin kuyruk yöneticisinin adıdır.

Bu özellik atlanırsa, varsayılan değer, ileti tanımlayıcısındaki (MQMD) ReplyToQMgr adıdır. Sonuçtaki ad boşsa, varsayılan olarak kuyruk yöneticisinin QMgrNamedeğerine ayarlanır.

QName (MQPSC_Q_NAME)

Değer, eşleşen alıkonan yayının kuyruk yöneticisi tarafından gönderildiği abone kuyruğunun adıdır.

Bu özellik atlanırsa, varsayılan değer, bu durumda boş olmaması gereken ileti tanımlayıcısındaki (MQMD) ReplyToQ adıdır.

SubName (MQPSC_SUBSCRIPTION_NAME)

Bu, belirli bir aboneliğe verilen bir addır. Bir **Request Update** komutunda belirtilirse, aboneliğin kendisinde VariableUserId ayarlanmadıkça, SubName değeri, kullanıcı kimliği dışındaki diğer tüm tanıtıcı alanlarından önceliklidir. VariableUserId ayarlanmazsa, *İstek Güncellemesi* komutu yalnızca komut iletilisinin kullanıcı kimliği aboneliğin kullanıcı kimliği ile eşleşiyorsa başarılı olur. Komut iletilisinin kullanıcı kimliği aboneliğin kullanıcı kimliğiyle eşleşmezse, komut MQRCCF_DUPLICATE_IDENTITY dönüş koduyla başarısız olur.

VariableUserId ayarlanırsa ve kullanıcı kimliği abonelikten farklıysa, yeni komut iletilisinin kullanıcı kimliğinin akış kuyruğuna göz atma ve aboneliğin abone kuyruğuna koyma yetkisi varsa komut başarılı olur. Ters durumda, komut MQRCCF_NOT_AUTHORIZED dönüş koduyla başarısız olur.

Bu komutun geleneksel kimliğiyle eşleşen bir abonelik varsa, ancak SubName yoksa, **Request Update** komutu MQRCCF_SUB_NAME_ERROR dönüş koduyla başarısız olur.

Geleneksel kimlikle eşleşen, ancak SubName belirtilmemiş bir komut iletilisi kullanılarak SubName olan bir abonelik için güncelleme isteme girişiminde bulunulursa, komut başarılı olur.

Örnek

Burada bir **Request Update** komut iletilisi için NameValueVerileri örneği verilmiştir. Örnek uygulamada, sonuçlar hizmeti tüm ekipler için en son puanları içeren yayınları istemek için bu iletiyi kullanır. CorrelIdde içinde olmak üzere abonenin kimliği, MQMD 'deki varsayılan değerlerden alınır.

```
<psc>
  <Command>ReqUpdate</Command>
  <RegOpt>CorrelAsId</RegOpt>
  <Topic>Sport/Soccer/State/LatestScore/#</Topic>
</psc>
```

Kuyruk Yöneticisi Yanıtı İletisi

Bir **Queue Manager Response** iletilisi, kuyruk yöneticisinden bir yayınlayıcının ya da abonenin ReplyToQ ögesine gönderilir; bu ileti, komut iletilisi tanımlayıcısı bir yanıt gerektiğini belirtmişse, kuyruk yöneticisi tarafından alınan bir komut iletilisinin başarılı ya da başarısız olduğunu gösterir.

Yanıt iletilisi, bir <psc> klasöründeki MQRFH2 üstbilgisinin NameValueVeri alanında bulunur.

Bir uyarı ya da hata durumunda, yanıt iletilisi komut iletilisindeki <psc> klasörünü ve <psc> klasörünü içerir. Varsa, ileti verileri kuyruk yöneticisi yanıt iletilisinde bulunmuyor. Bir hata durumunda, hataya neden olan iletilerin hiçbiri işlenmedi; bir uyarı durumunda, iletilinin bir kısmı başarıyla işlenmiş olabilir.

Yanıt gönderilirken bir hata oluşursa:

- Yayınlama iletileri için, kuyruk yöneticisi, MQPUT başarısız olursa yanıtı IBM MQ ileti kuyruğuna göndermeyi dener. Bu, yanıt yayınlayıcıya geri gönderilemese de yayının abonelere gönderilmesini sağlar.
- Diğer iletiler için ya da yayın yanıtı teslim edilmeyen iletiler kuyruğuna gönderilemezse, bir hata günlüğe kaydedilir ve komut iletilisi olağan durumda geriye işlenir. Bunun olup olmadığı, MQInput düğümünün nasıl yapılandırıldığına bağlıdır.

Özellikler

Tamamlama (MQPSCR_COMPLETION)

Üç değerden birini alabilen tamamlanma kodu:

tamam

Komut başarıyla tamamlandı

uyarı

Komut tamamlandı, ancak uyarıyla

hata

Komut başarısız oldu

Yanıt (MQPSCR_RESPONSE)

Komut uyarı ya da hatatamamlanma kodu üretmişse, komut iletisine verilen yanıt. Bir < Neden> özelliği içerir ve uyarı ya da hatanın nedenini belirten başka özellikler içerebilir.

Bir ya da daha çok hata olması durumunda, yalnızca ilk hatanın nedenini gösteren tek bir yanıt klasörü vardır. Bir ya da daha çok uyarı durumunda, her uyarı için bir yanıt klasörü vardır.

Neden (MQPSCR_REASON)

Tamamlanma kodu bir uyarı ya da hataise, tamamlanma kodunu niteleyen neden kodu. Aşağıdaki örnekte listelenen hata kodlarından birine ayarlanır. < Neden> özelliği bir < Yanıt> klasöründe bulunur. Neden kodunu, hatanın ya da uyarının nedenini gösteren, <psc> klasöründeki herhangi bir geçerli özellik (örneğin, bir konu adı) izleyebilir. Neden kodu alırsanız ???, Verilerin doğruluğunu denetleyin; örneğin, eşleşen açılı ayraçlar (< >).

Örnekler

Queue Manager Response iletisindeki NameValueData ile ilgili bazı örnekler aşağıda verilmiştir. Başarılı bir yanıt aşağıdaki gibi olabilir:

```
<pscr>
  <Completion>ok</Completion>
</pscr>
```

Burada bir hata yanıtı örneği verilmiştir; hata bir süzgeç hatasıdır. İlk NameValueVeri dizgisi yanıtı içerir; ikincisi özgün komutu içerir.

```
<pscr>
  <Completion>error</Completion>
  <Response>
    <Reason>3150</Reason>
  </Reponse>
</pscr>

<psc>
  ...
  command message (to which
  the queue manager is responding)
  ...
</psc>
```

Aşağıda bir uyarı yanıtı örneği verilmiştir (yetkisiz konular nedeniyle). İlk NameValueData dizgisi yanıtı içerir; ikinci NameValueData dizgisi özgün komutu içerir.

```
<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>
```

Yayınlama/abone olma neden kodları

Bu neden kodları, bir yayınlama/abone olma yanıtı <psc> klasörünün Neden alanında döndürülebilir. C ya da C++ programlama dillerinde bu kodları göstermek için kullanılacak sabitler de listelenir.

MQRC_değişmezleri, IBM MQ cmqc . h üstbilgi dosyasını gerektirir. MQRCCF_değişmezleri için IBM MQ cmqcfc . h üstbilgi dosyası gerekir (cmqpsc . h üstbilgi dosyasını gerektiren MQRCCF_FILTER_ERROR ve MQRCCF_YANLIŞ_USER dışında).

Neden kodu ve metin	Açıklama	Sertifikaı Veren
2336 MQRC_RFH_COMMAND_ERROR	Bir <psc> klasörünün < Command> alanı için geçerli değerler şunlardır: RegSub, DeregSub, publish, DeletePubve ReqUpdate. Diğer değerler, bu hata kodunun yayınlanmasıyla sonuçlanır.	Herhangi bir komut
2337 MQRC_RFH_PARM_ERROR	<psc> ve <mcd> klasörlerinin her ikisi de, içinde belirtilebilir geçerli bir parametre kümesine sahip. Bu klasörlerin tanımlarını denetleyin ve yanlış değıştirmeler belirtmediđinizi doğrulayın.	Herhangi bir komut
2338 MQRC_RFH_DUPLICATE_PARM	<psc> klasöründeki bazı parametreler (örneğin, Konu) yinelenebilir, ancak diğerleri (örneğin, Komut) yinelenemez. Yinelenemez bir parametreyi yinelemediđinizi denetleyin.	Herhangi bir komut
2339 MQRC_RFH_PARM_MISSING	<psc> ya da <mcd> klasörlerindeki bazı parametreler isteđe bađlıdır ve atlanabilir; bazıları zorunludur ve atlanmamalıdır. <psc> ve <mcd> klasörlerinize tüm zorunlu parametreleri dahil edip etmediđinizi denetleyin.	Herhangi bir komut
2551 MQRC_SELECTION_NOT_KULLANILAMIYOR	Belirtilen bir süzgece sahip hangi abonelerin yayını alması gerektiđini saptamak için kullanılacak genişletilmiş ileti seçimi sađlayıcısı yok.	Yayınlama, Abone Kaydetme ve Güncelleme İste
	Belirtilen abonenin süzgecini işlemek için kullanılacak genişletilmiş ileti seçimi sađlayıcısı yok.	Aboneyi Kaydet ve Güncelleme İste
2554 MQRC_CONTENT_HATA	Genişletilmiş bir ileti seçim sađlayıcısı, yürürlükteki ya da alıkonan yayında bir hata buldu.	Güncellemeyi Yayınla ve İste
3008 MQRCCF_COMMAND_FAILED	Komutun dođru yürütölmesini engelleyen bir iç hata oluştu. Komut yeniden yayınlandıđında hata oluşabilir. Kuyruk yöneticisine ilişkin sistem olay günlüđu, sorunu IBM' e bildirirken kullanılması gereken bilgileri içerir.	Herhangi bir komut

Neden kodu ve metin	Açıklama	Sertifikayı Veren
3072 MQRCCF_TOPIC_HATA	Konu parametresi için sağladığınız değerlerden biri ya da daha fazlası yanlış. Konu ile ilgili değerlerinizin belirtilen kısıtlamalara uygun olup olmadığını denetleyin.	Herhangi bir komut
3073 MQRCCF_NOT_REGISTERED	DeregSub ya da ReqUpdate komutunuzda belirttiğiniz SubPoint, Konu ve Süzgeç birleşimi, önceden kaydettirdiğiniz bir birleşim değildi ya da DeregAll seçeneği belirtiliyse, DeregSub komutu için SubPoint, Topic ya da Filter özelliklerinden biri herhangi bir aboneliğin kaydını silmek için kullanılmadı.	Abone Kaydını Kaldır ve Güncelleme İsteğini İste komutları
3074 MQRCCF_Q_MGR_AD_HATASI	Belirtilen kuyruk yöneticisi geçerli değil ya da kuyruk yöneticisi kullanılamıyor ya da yok.	Abone Kaydını Kaldır, Yayınla, Aboneyi Kaydet ve Güncellemeyi İste komutları
3076 MQRCCF_Q_AD_HATASI	Belirtilen kuyruk adı geçerli değil ya da kuyruk belirtilen kuyruk yöneticisinde yok.	Abone Kaydını Kaldır, Yayınla, Aboneyi Kaydet ve Güncellemeyi İste komutları
3077 MQRCCF_NO_RETAINED_MSG	Belirttiğiniz konu için alıkonan ileti yok. Bu, uygulama programınızın tasarımına bağlı olarak bir hata olabilir ya da olmayabilir.	Güncelleme İste komutu
3079 MQRCCF_HATA_Q	RegSub, DeregSubve ReqUpdate komutları her zaman SYSTEM.BROKER.CONTROL.QUEUE kuyruğu. Yayınlama ve Yayını Sil komutları, amaçlandıkları belirli yayınlama/abone olma ileti akışı için giriş kuyruğuna gönderilir; bu, ileti akışının tasarlandığı zaman belirlenir. Bir komut yanlış kuyruğa gönderilirse bu hata kodu döndürülür.	Herhangi bir komut
3080 MQRCCF_CORREL_ID_HATA	RegOpt değiştirgelerinizden biri olarak CorrelAsId belirttiniz. Ancak, MQMD 'nin CorrelId alanı geçerli bir ilinti tanıtıcısı içermiyor (MQCI_NONE olarak ayarlandı).	Abone Kaydını Kaldır ve Aboneyi Kaydet komutları
3081 MQRCCF_NOT_YETKILI	İstenen işlemi gerçekleştirme yetkiniz yok. Kuyruk yöneticisine ilişkin yetkilendirme ayarları, Topics Hierarchy düzenleyicisi kullanılarak sistem yöneticisi tarafından işlenir.	Abone komutlarını yayınlama ve kaydet

Neden kodu ve metin	Açıklama	Sertifikayı Veren
3083 MQRCCF_REG_OPTIONS_ERROR	<psc> klasöründe RegSub ya da DeregSub komutunuzu içeren tanınmayan bir RegOpt parametresi belirttiniz.	Abone Kaydını Kaldır ve Aboneyi Kaydet komutları
3084 MQRCCF_PUB_OPTIONS_HATA	Yayınlama komutunuzu içeren <psc> klasöründe tanınmayan bir PubOpt değişikliği belirttiniz.	Yayınlama komutu
3087 MQRCCF_DEL_OPTIONS_HATA	<psc> klasöründe DeletePub komutunuzu içeren tanınmayan bir DelOpt parametresi belirttiniz.	Yayını Sil komutu
3150 MQRCCF_FILTER_HATA	Süzgeç parametresi için belirtilen değer geçerli değil. Süzgeç ifadeleri için geçerli sözdizimini açıklayan bölümü denetleyin ve ifadenizin uyduğundan emin olun.	Abonenin Kaydını Kaldır, Aboneyi Kaydet ve Güncelleme Komutlarını İste
3151 MQRCCF_YANLIŞ KULLANICI	Belirtilen abonelik eşleşen bir abonelik zaten var; ancak, farklı bir kullanıcı tarafından kaydedildi. Bir abonelik yalnızca, başlangıçta kaydettirmiş olan kullanıcı tarafından değiştirilebilir ya da kayıttan kaldırılabilir.	Abonenin Kaydını Kaldır, Aboneyi Kaydet ve Güncelleme Komutlarını İste
3152 MQRCCF_DUPLICATE_SUBSCRIPTION	Farklı bir abonelik adıyla eşleşen bir abonelik zaten var.	
3153 MQRCCF_SUB_AD_HATASI	Abonelik adının biçimi geçerli değil ya da abonelik adı olmayan eşleşen bir abonelik zaten var.	
3154 MQRCCF_SUB_IDENTITY_ERROR	Abonelik kimliği parametresi hatalı. Sağlanan değer, izin verilen uzunluk üst sınırını aşıyor ya da abonelik kimliği şu anda aboneliğin kimlik kümesinin bir üyesi değil ve bir Birleştirme kaydı seçeneği belirtilmedi.	
3155 MQRCCF_SUBSCRIPTION_IN_USE	Bu kümenin tek üyesi olmadığında kimlik kümesinin bir üyesi tarafından bir aboneliği değiştirme ya da aboneliğin kaydını kaldırma girişiminde bulunuldu.	
3156 MQRCCF_SUBSCRIPTION_LOCKED	Abonelik şu anda başka bir kimlik tarafından özel olarak kilitleti.	
3157 MQRCCF_ALREADY_JOINED	Bir Birleştirme kaydı seçeneği belirtildi, ancak abone kimliği zaten aboneliğin kimlik kümesinin bir üyesiydi.	

Kuyruk yöneticisine komut iletilerinde MQMD ayarları

Kuyruk yöneticisine komut iletileri gönderen uygulamalar, ileti tanımlayıcısında (MQMD) aşağıdaki alan ayarlarını kullanır. Varsayılan değer olarak bırakılacak ya da olağan şekilde herhangi bir geçerli değere ayarlanabilecek alanlar burada listelenmez.

Rapor

Bkz. `MsgType` ve `CorrelId`.

MsgType

`MsgType` , `MQMT_REQUEST` ya da `MQMT_DATAGRAM` olarak ayarlanmalıdır. `MsgType` bu değerlerden birine ayarlanmazsa `MQRC_MSG_TYPE_ERROR` döndürülür.

`MsgType` , her zaman bir yanıt gerekiyorsa, bir komut ileti için `MQMT_REQUEST` olarak ayarlanmalıdır. Rapor alanındaki `MQRO_PAN` ve `MQRO_NAN` işaretleri bu durumda önemli değildir.

`MsgType` `MQMT_DATAGRAM` olarak ayarlanırsa, yanıtlar Rapor alanındaki `MQRO_PAN` ve `MQRO_NAN` işaretlerinin ayarına bağlıdır:

- `MQRO_PAN` tek başına, kuyruk yöneticisinin yalnızca komut başarılı olursa yanıt göndereceği anlamına gelir.
- `MQRO_NAN` tek başına, kuyruk yöneticisinin yalnızca komut başarısız olursa yanıt göndereceği anlamına gelir.
- Bir komut bir uyarıyla tamamlanırsa, `MQRO_PAN` ya da `MQRO_NAN` ayarlanırsa bir yanıt gönderilir.
- `MQRO_PAN` + `MQRO_NAN`, kuyruk yöneticisinin komut başarılı ya da başarısız olursa yanıt göndereceği anlamına gelir. Bu, kuyruk yöneticisinin perspektifinden `MsgType` ögesinin `MQMT_REQUEST` olarak ayarlanmasıyla aynı etkiye sahiptir.
- `MQRO_PAN` ya da `MQRO_NAN` ayarlanmazsa, hiçbir yanıt gönderilmez.

Biçim

`MQFMT_RF_HEADER_2` olarak ayarla

MsgId

Kuyruk yöneticisinin benzersiz bir değer oluşturması için bu alan olağan durumda `MQMI_NONE` olarak ayarlanır.

CorrelId

Bu alan herhangi bir değere ayarlanabilir. Gönderenin kimliği `CorrelId` içeriyorsa, kuyruk yöneticisi tarafından gönderene gönderilen tüm yanıt iletilerinde ayarlandığından emin olmak için Rapor alanında `MQRO_PASS_CORREL_ID` ile birlikte bu değeri belirtin.

ReplyToQ

Bu alan, varsa yanıtların gönderileceği kuyruğu tanımlar. Bu, gönderenin kuyruğu olabilir; bu, `QName` parametresinin iletiden çıkarılabilmesi avantajına sahiptir. Ancak, yanıtlar farklı bir kuyruğa gönderilirse, `QName` parametresi gerekir.

ReplyToQMgr

Bu alan, yanıtlar için kuyruk yöneticisini tanımlar. Bu alanı boş bırakırsanız (varsayılan değer), yerel kuyruk yöneticisi bu alana kendi adını koyar.

Kuyruk yöneticisi tarafından iletilen yayınlara ilişkin MQMD ayarları

Kuyruk yöneticisi, bir aboneye yayın gönderirken ileti tanımlayıcısındaki (MQMD) alanların bu ayarlarını kullanır. MQMD ' deki diğer tüm alanlar varsayılan değerlerine ayarlanır.

Rapor

Rapor `MQRO_NONE` olarak ayarlandı.

MsgType

`MsgType` , `MQMT_DATAGRAM` olarak ayarlandı.

Son kullanma tarihi

`Süre bitimi`, yayıncıdan alınan `Yayınla` iletilerindeki değere ayarlanır. Alınan bir ileti söz konusu olduğunda, kalan süre, iletinin kuyruk yöneticisinde bulunduğu yaklaşık süre kadar azalır.

Biçim

Biçim MQFMT_RF_HEADER_2 olarak ayarlandı

MsgId

MsgId benzersiz bir değere ayarlandı.

CorrelId

CorrelId abonenin kimliğinin bir parçasıysa, bu, kayıt sırasında abone tarafından belirtilen değerdir. Tersi durumda, kuyruk yöneticisi tarafından seçilen sıfır dışında bir değerdir.

Öncelik

Öncelik , yayınlayıcı tarafından ayarlanan değeri alır ya da yayınlayıcı MQPRI_PRIORITY_AS_Q_DEF belirtilirse çözülür.

Kalıcılık

Kalıcı Saklama , bu yayının gönderildiği aboneye ilişkin Aboneyi Kaydet iletisinde tersi belirtilmedikçe, yayınlayıcı MQPER_PERSISTENCE_AS_Q_DEF değerini belirtmişse, yayınlayıcı tarafından ayarlanan değeri alır ya da çözüldü olarak alır.

ReplyToQ

ReplyToQ , boşluk olarak ayarlanır.

ReplyToQMgr

ReplyToQMgr , kuyruk yöneticisinin adına ayarlanır.

UserIdentifier

UserIdentifier , abone kaydolduğunda ayarlandığı gibi, abonenin kullanıcı kimliğidir.

AccountingToken

AccountingToken , abonenin ilk kaydolduğunda ayarlandığı şekilde, abonenin hesap simgesidir.

AppIdentityVerileri

AppIdentityVerileri , abonenin ilk kaydolduğunda ayarlandığı gibi, abonenin uygulama kimliği verileridir.

PutAppTip

PutAppTip , MQAT_BROKER olarak ayarlandı.

PutAppAdı

PutAppAd , kuyruk yöneticisinin adının ilk 28 karakterine ayarlanır.

PutDate

PutDate , iletinin konma tarihidir.

PutTime

PutTime , iletinin konma zamanıdır.

AppOriginVerileri

AppOriginVerileri boş olarak ayarlanır.

Kuyruk yöneticisi yanıt iletilerindeki MQMD ayarları

Bir kuyruk yöneticisi, bir yayın iletisine yanıt gönderirken ileti tanımlayıcısındaki (MQMD) alanların bu ayarlarını kullanır. MQMD ' deki diğer tüm alanlar varsayılan değerlerine ayarlanır.

Rapor

Rapor , tüm sıfırlara ayarlanır.

MsgType

MsgType , MQMT_REPLY olarak ayarlandı.

Biçim

Biçim MQFMT_RF_HEADER_2 olarak ayarlandı

MsgId

MsgId ayarı, özgün komut iletisindeki Rapor seçeneklerine bağlıdır. Varsayılan olarak, kuyruk yöneticisinin benzersiz bir değer oluşturması için MQMI_NONE olarak ayarlanır.

CorrelId

CorrelId ayarı, özgün komut iletisindeki Rapor seçeneklerine bağlıdır. Varsayılan olarak bu, CorrelId değerinin, komut iletisinin MsgId ile aynı değere ayarlandığı anlamına gelir. Bu, komutları yanıtlarıyla ilintilendirmek için kullanılabilir.

Öncelik

Öncelik , özgün komut iletisiyle aynı değere ayarlanır.

Kalıcılık

Kalıcılık , özgün komut iletisinde ayarlanan değere ayarlanır.

Son kullanma tarihi

Süre bitimi , kuyruk yöneticisi tarafından alınan özgün komut iletisiyle aynı değere ayarlanır.

PutAppTip

PutAppTip , MQAT_BROKER olarak ayarlandı.

PutAppAdı

PutAppAd , kuyruk yöneticisinin adının ilk 28 karakterine ayarlanır.

Diğer bağlam alanları, MQPMO_PASS_IDENTITY_CONTEXT ile oluşturulduğu gibi ayarlanır.

Makine kodlamaları

Bu bölümde, ileti tanımlayıcısındaki *Encoding* alanının yapısı açıklanmaktadır.

Yapıdaki alanların bir özeti için bkz. [“MQMD-İleti tanımlayıcı” sayfa 420](#) .

Encoding alanı, dört ayrı alt alana bölünmüş 32 bitlik bir tamsayıdır; bu alt alanlar şunları tanımlar:

- İkili tamsayılar için kullanılan kodlama
- Paketlenmiş ondalık tamsayılar için kullanılan kodlama
- Kayan noktalı sayılar için kullanılan kodlama
- Ayrılmış bitler

Her alt alan, alt alana karşılık gelen konumlarda 1 bit ve başka bir yerde 0 bit bulunan bir bit maskesiyle tanımlanır. Bitler 0 biti en önemli biti, 31 biti en az anlamlı biti olacak şekilde numaralandırılır. Aşağıdaki maskeler tanımlandı:

MQENC_INTEGER_MASK

İkili tamsayı kodlaması için maske.

Bu alt alan, *Encoding* alanında 28-31 bit konumlarını kaplar.

MQENC_DECIMAL_MASK

Paketlenmiş ondalık tamsayı kodlaması için maske.

Bu alt alan, *Encoding* alanında 24-27 bit konumlarını kaplar.

MQENC_FLOAT_MASK

Kayan nokta kodlaması için maske.

Bu alt alan, *Encoding* alanında 20-23 bit konumlarını kaplar.

MQENC_RESERVED_MASK

Ayrılmış bitler için maske.

Bu alt alan, *Encoding* alanında 0 ile 19 arasındaki bit konumlarını kaplar.

İkili tamsayı kodlaması

Aşağıdaki değerler ikili tamsayı kodlaması için geçerlidir:

MQENC_INTEGER_UNDEFINED

İkili tamsayılar, tanımlanmamış bir kodlama kullanılarak gösterilir.

MQENC_INTEGER_NORMAL

İkili tamsayılar geleneksel şekilde gösterilir:

- Sayıdaki en az anlamlı bayt, sayıdaki herhangi bir baytın en yüksek adresine sahiptir; en önemli bayt en düşük adrese sahiptir
- Her bir bayttaki en az anlamlı bit, bir sonraki yüksek adresle bayta bitişiktir; her bayttaki en anlamlı bit, bir sonraki alt adresle birlikte bayta bitişiktir

MQENC_INTEGER_REVERSED

İkili tamsayılar MQENC_INTEGER_NORMAL ile aynı şekilde, ancak byte 'lar ters sırada düzenlenmiş olarak gösterilir. Her byte içindeki bitler, MQENC_INTEGER_NORMAL ile aynı şekilde düzenlenir.

Paketli-ondalık-tamsayı kodlaması

Paketlenmiş onlu tamsayı kodlaması için aşağıdaki değerler geçerlidir:

MQENC_DECIMAL_UNDEFINED

Paketlenmiş ondalık tamsayılar, tanımlanmamış bir kodlama kullanılarak gösterilir.

MQENC_DECIMAL_NORMAL

Paketlenmiş ondalık tamsayılar geleneksel olarak gösterilir:

- Sayının yazdırılabilir biçimindeki her ondalık basamak, X' 0 '-X' 9' aralığındaki tek bir onaltılı sayıyla paketlenmiş onlu olarak gösterilir. Her onaltılı sayı dört bit kaplar ve bu nedenle paketlenmiş onlu sayıdaki her bayt, sayının yazdırılabilir biçiminde iki ondalık basamağı temsil eder.
- Paketlenmiş onlu sayıdaki en az anlamlı bayt, en az anlamlı ondalık basamağı içeren bayttır. Bu bayt içinde, en önemli dört bit en az anlamlı ondalık basamağı içerir ve en az anlamlı dört bit işareti içerir. İşaret X'C '(pozitif), X 'D' (negatif) ya da X'F ' (işaretsiz).
- Sayıdaki en az anlamlı bayt, sayıdaki herhangi bir baytın en yüksek adresine sahiptir; en önemli bayt en düşük adrese sahiptir.
- Her bayttaki en az anlamlı bit, bir sonraki yüksek adresle bayta bitişiktir; her bayttaki en anlamlı bit, bir sonraki alt adresle birlikte bayta bitişiktir.

MQENC_DECIMAL_REVERSED

Paketlenmiş ondalık tamsayılar, MQENC_DECIMAL_NORMAL ile aynı şekilde, ancak byte 'lar ters sırada düzenlenmiş olarak gösterilir. Her byte içindeki bitler, MQENC_DECIMAL_NORMAL ile aynı şekilde düzenlenir.

Kayan nokta kodlaması

Kayan nokta kodlaması için aşağıdaki değerler geçerlidir:

MQENC_FLOAT_UNDEFINED

Kayan noktalı sayılar, tanımlanmamış bir kodlama kullanılarak gösterilir.

MQENC_FLOAT_IEEE_NORMAL

Kayan noktalı sayılar standart IEEE kullanılarak gösterilir⁴kayan nokta biçimi; baytlar aşağıdaki gibi düzenlenmiştir:

- Mantissa 'daki en az anlamlı bayt, sayıdaki herhangi bir baytın en yüksek adresine sahiptir; üssü içeren bayt en düşük adrese sahiptir
- Her bir bayttaki en az anlamlı bit, bir sonraki yüksek adresle bayta bitişiktir; her bayttaki en anlamlı bit, bir sonraki alt adresle birlikte bayta bitişiktir

IEEE float kodlamasının ayrıntıları IEEE Standard 754 'te bulunabilir.

MQENC_FLOAT_IEEE_REVERSED

Kayan noktalı sayılar, MQENC_FLOAT_IEEE_NORMAL ile aynı şekilde, ancak baytlar ters sırada düzenlenmiş olarak gösterilir. Her byte içindeki bitler, MQENC_FLOAT_IEEE_NORMAL ile aynı şekilde düzenlenir.

⁴ Elektrik ve Elektronik Mühendisleri Enstitüsü

MQENC_FLOAT_S390

Kayan noktalı sayılar standart System/390 kayan noktalı biçimi kullanılarak gösterilir; bu, System/370 tarafından da kullanılır.

Kodlamalar oluşturuluyor

MQMD ' de *Encoding* alanı için bir değer oluşturmak üzere, gerekli kodlamaları tanımlayan ilgili sabitler bir araya eklenebilir (aynı sabiti bir kereden fazla eklemeyin) ya da bitset OR işlemi kullanılarak birleştirilebilir (programlama dili bit işlemlerini destekliyorsa).

Hangi yöntem kullanılırsa kullanılsın, MQENC_INTEGER_ * kodlamalarında yalnızca birini MQENC_DECIMAL_ * kodlamalarında biriyle ve MQENC_FLOAT_ * kodlamalarında biriyle birleştirin.

Kodlamalar çözümleniyor

Encoding alanı alt alanlar içerir; bu nedenle, tamsayı, paketlenmiş ondalık ya da kayan noktalı sayı kodlamasını incelemesi gereken uygulamalar, açıklanan tekniklerden birini kullanmalıdır.

Bit işlemlerini kullanma

Programlama dili bit işlemlerini destekliyorsa, aşağıdaki adımları gerçekleştirin:

1. Gereken kodlama tipine göre aşağıdaki değerlerden birini seçin:

- İkili tamsayı kodlaması için MQENC_INTEGER_MASK
- Paketlenmiş ondalık tamsayı kodlaması için MQENC_DECIMAL_MASK
- Kayan nokta kodlaması için MQENC_FLOAT_MASK

Adeğerini çağırın.

2. Bitli AND işlemi kullanarak *Encoding* alanını A ile birleştirin; sonucu çağırın B.

3. B gerekli kodlamadır ve bu kodlama tipi için geçerli olan değerlerin her biriyle eşitlik için test edilebilir.

Aritmetik kullanılıyor

Programlama dili bit işlemlerini *desteklemiyorsa* , tamsayı aritmetiğini kullanarak aşağıdaki adımları gerçekleştirin:

1. Gereken kodlama tipine göre aşağıdaki değerlerden birini seçin:

- İkili tamsayı kodlaması için 1
- Paketlenmiş ondalık tamsayı kodlaması için 16
- Kayan nokta kodlaması için 256

Adeğerini çağırın.

2. *Encoding* alanının değerini A ile bölün; Bsonucunu çağırın.

3. B ögesini 16 'ya bölün; sonucu Colarak çağırın.

4. C 'i 16 ile çarparak B ' dan çıkarın; Dsonucunu çağırın.

5. D ' i A ile çarpma; Esonucunu çağırın.

6. E gerekli kodlamadır ve bu kodlama tipi için geçerli olan değerlerin her biriyle eşitlik için test edilebilir.

Makine mimarisi kodlamasının özeti

Makine mimarilerine ilişkin kodlamalar [Çizelge 631 sayfa 891](#) içinde gösterilmektedir.

Çizelge 631. Makine mimarilerine ilişkin kodlamalar özeti

Makine mimarisi	İkili tamsayı kodlaması	Paketlenmiş ondalık tamsayı kodlaması	Kayan nokta kodlaması
IBM i	normal	normal	IEEE normal
Intel x86	Tersine Çevrilmiş	Tersine Çevrilmiş	IEEE tersine çevrildi
PowerPC	normal	normal	IEEE normal
System/390	normal	normal	System/390

Rapor seçenekleri ve ileti işaretleri

Bu kısım, MQGET, MQPUT ve MQPUT1 çağrılarında belirtilen ileti tanımlayıcısı MQMD ' nin bir parçası olan *Report* ve *MsgFlags* alanlarını açıklar.

Bu bölümdeki konular aşağıdakileri açıklar:

- Rapor alanının yapısı ve kuyruk yöneticisinin bunu nasıl işlediği
- Bir uygulama rapor alanını nasıl analiz eder?
- İleti işaretleri alanının yapısı

MQMD ileti tanımlayıcısı hakkında daha fazla bilgi için bkz. "[MQMD-İleti tanımlayıcı](#)" sayfa 420.

Rapor alanının yapısı

Bu bilgiler, rapor alanının yapısını açıklar.

Report alanı, üç ayrı alt alana bölünmüş 32 bitlik bir tamsayıdır. Bu alt alanlar aşağıdakileri tanımlar:

- Yerel kuyruk yöneticisi bunları tanımazsa reddedilen rapor seçenekleri
- Yerel kuyruk yöneticisi bunları tanımasa da her zaman kabul edilen rapor seçenekleri
- Yalnızca diğer bazı koşullar karşılandığında kabul edilen rapor seçenekleri

Her alt alan, alt alana karşılık gelen konumlarda 1 bit ve başka bir yerde 0 bit bulunan bir bit maskesiyle tanımlanır. Bir alt alandaki bitlerin bitişik olması gerekmez. Bitler 0 biti en önemli biti, 31 biti en az anlamlı biti olacak şekilde numaralandırılır. Alt alanları tanımlamak için aşağıdaki maskeler tanımlanır:

MQRO_REJECT_UNSUP_MASK

Bu maske, yerel kuyruk yöneticisi tarafından desteklenmeyen rapor seçeneklerinin MQPUT ya da MQPUT1 çağrısına MQCC_FAILED ve neden kodu MQRC_REPORT_OPTIONS_ERROR ile başarısız olmasına neden olduğu *Report* alanındaki bit konumlarını tanımlar.

Bu alt alan 3 ve 11 ile 13 arasındaki bit konumlarını kaplar.

MQRO_ACCEPT_UNSUP_MASK

Bu maske, *Report* alanında, yerel kuyruk yöneticisi tarafından desteklenmeyen rapor seçeneklerinin yine de MQPUT ya da MQPUT1 çağrılarında kabul edildiği bit konumlarını tanımlar. Bu durumda, MQRC_UNKNOWN_REPORT_OPTION neden koduyla MQCC_WARNING tamamlanma kodu döndürülür.

Bu alt alan 0-2, 4-10 ve 24-31 arasındaki bit konumlarını kaplar.

Bu alt alana aşağıdaki rapor seçenekleri eklenmiştir:

- 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_SÜRE SONU
- 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

Bu maske, *Report* alanındaki bit konumlarını tanımlar; yerel kuyruk yöneticisi tarafından desteklenmeyen rapor seçenekleri yine de MQPUT ya da MQPUT1 çağrılarında *kabul edilir* ve her iki koşul da karşılanır:

- İleti uzak bir kuyruk yöneticisine gönderiliyor.
- Uygulama iletiyi doğrudan yerel bir iletim kuyruğuna koymuyor (yani, MQOPEN ya da MQPUT1 çağrısında belirtilen nesne tanımlayıcısındaki *ObjectQMGrName* ve *ObjectName* alanları tarafından tanımlanan kuyruk yerel bir iletim kuyruğu değil).

Bu koşullar karşılandıysa, MQRC_UNKNOWN_REPORT_OPTION neden koduyla MQCC_WARNING (tamamlanma kodu) döndürülür; yoksa, MQRC_REPORT_OPTIONS_ERROR neden koduyla MQCC_FAILED döndürülür.

Bu alt alan, 14 ile 23 arasındaki bit konumlarını kaplar.

Bu alt alana aşağıdaki rapor seçenekleri eklenmiştir:

- MQRO_COA
- MQRO_COA_WITH_DATA
- MQRO_COA_WITH_FULL_DATA
- MQRO_COD
- MQRO_COD_WITH_DATA
- MQRO_COD_WITH_FULL_DATA

Kuyruk yöneticisinin tanımadığı *Report* alanında herhangi bir seçenek belirtilirse, kuyruk yöneticisi, *Report* alanını o alt alana ilişkin maskeyle birleştirmek için bit temelinde AND (VE) işlemini kullanarak her alt alanı denetler. Bu işlemin sonucu sıfır değilse, daha önce açıklanan tamamlanma kodu ve neden kodları döndürülür.

MQCC_WARNING döndürülürse, başka uyarı koşulları varsa hangi neden kodunun döndürüleceği tanımlanmaz.

Yerel kuyruk yöneticisi tarafından tanınmayan rapor seçeneklerini belirtme ve kabul etme yeteneği, *uzak* bir kuyruk yöneticisi tarafından tanınan ve işlenen bir rapor seçeneğiyle bir ileti gönderilirken yararlıdır.

Rapor alanı çözümleniyor

Report alanı alt alanlar içerir; bu nedenle, iletiyi gönderen kişinin belirli bir raporu isteyip istemediğini denetlemesi gereken uygulamalar, açıklanan tekniklerden birini kullanmalıdır.

Bit işlemlerini kullanma

Programlama dili bit işlemlerini destekliyorsa, aşağıdaki adımları gerçekleştirin:

1. Denetlenecek rapor tipine göre aşağıdaki değerlerden birini seçin:

- COA raporu için MQRO_COA_WITH_FULL_DATA
- COD raporu için MQRO_COD_WITH_FULL_DATA
- Kural dışı durum raporu için MQRO_EXCEPTION_WITH_FULL_DATA
- Süre sonu raporu için MQRO_EXPIRATION_WITH_FULL_DATA

Adeğerini çağırın.

z/OSüzerinde MQRO_*_WITH_FULL_DATA değerleri yerine MQRO_*_WITH_DATA değerlerini kullanın.

2. Bitli AND işlemini kullanarak *Report* alanını A ile birleştirin; sonucu çağırın B.
3. Bu rapor tipi için mümkün olan her değerle eşitlik için B ' i test edin.

Örneğin, A MQRO_EXCEPTION_WITH_FULL_DATA ise, iletiyi gönderen tarafından nelerin belirtildiğini belirlemek için aşağıdakilerin her biriyle eşitlik için B sınavasını yapın:

- MQRO_NONE
- MQRO_EXCEPTION
- MQRO_EXCEPTION_WITH_DATA
- MQRO_EXCEPTION_WITH_FULL_DATA

Testler, uygulama mantığı için en uygun sırayla gerçekleştirilebilir.

MQRO_PASS_MSG_ID ya da MQRO_PASS_CORREL_ID seçeneklerini sınamak için benzer bir yöntem kullanın; bu iki değişmezden hangisi uygunsa A değerini seçin ve daha önce açıklandığı gibi devam edin.

Aritmetik kullanılıyor

Programlama dili bit işlemlerini *desteklemiyorsa* , tamsayı aritmetiğini kullanarak aşağıdaki adımları gerçekleştirin:

1. Denetlenecek rapor tipine göre aşağıdaki değerlerden birini seçin:

- COA raporu için MQRO_COA
- COD raporu için MQRO_COD
- Kural dışı durum raporu için MQRO_EXCEPTION
- Süre bitimi raporu için MQRO_SÜRE bitimi

Adeğerini çağırın.

2. *Report* alanını A ile bölün; Bsonucunu çağırın.
3. B ögesini 8 ' e bölün; Csonucunu çağırın.
4. C ögesini 8 ile çarparak B içinden çıkarın; Dsonucunu çağırın.
5. D ' i A ile çarpma; Esonucunu çağırın.
6. Bu rapor tipi için mümkün olan her değerle eşitlik için E ' i test edin.

Örneğin, A MQRO_EXCEPTION ise, iletiyi gönderen tarafından ne belirtildiğini belirlemek için aşağıdakilerden her biriyle eşitlik için E sınavasını yapın:

- MQRO_NONE
- MQRO_EXCEPTION
- MQRO_EXCEPTION_WITH_DATA
- MQRO_EXCEPTION_WITH_FULL_DATA

Testler, uygulama mantığı için en uygun sırayla gerçekleştirilebilir.

Aşağıdaki sözde kod, kural dışı durum raporu iletileri için bu tekniği gösterir:

A = MQRO_EXCEPTION
B = Report/A

C = B/8
D = B - C*8
E = D*A

MQRO_PASS_MSG_ID ya da MQRO_PASS_CORREL_ID seçeneklerini sınamak için benzer bir yöntem kullanın; A Bu iki değişmezden hangisi uygunsa, değeri seçin ve daha önce açıklandığı gibi devam edin, ancak önceki adımlarda 8 değerini 2değeriyle değiştirin.

İletinin yapısı-işaretler alanı

Bu bilgiler, ileti işaretleri alanının yapısını açıklar.

MsgFlags alanı, üç ayrı alt alana bölünmüş 32 bitlik bir tamsayıdır. Bu alt alanlar aşağıdakileri tanımlar:

- Yerel kuyruk yöneticisi bunları tanımazsa reddedilen ileti işaretleri
- Yerel kuyruk yöneticisi bunları tanımasa da, her zaman kabul edilen ileti işaretleri
- Yalnızca diğer bazı koşullar yerine getirildiğinde kabul edilen ileti işaretleri

Not: *MsgFlags* içindeki tüm alt alanlar, kuyruk yöneticisi tarafından kullanılmak üzere ayrılmıştır.

Her alt alan, alt alana karşılık gelen konumlarda 1 bit ve başka bir yerde 0 bit bulunan bir bit maskesiyle tanımlanır. Bitler 0 biti en önemli biti, 31 biti en az anlamlı biti olacak şekilde numaralandırılır. Alt alanları tanımlamak için aşağıdaki maskeler tanımlanır:

MQMF_REJECT_UNSUP_MASK

Bu maske, yerel kuyruk yöneticisi tarafından desteklenmeyen ileti işaretlerinin MQPUT ya da MQPUT1 çağrısına MQCC_FAILED tamamlama kodu ve MQRC_MSG_FLAGS_ERROR neden kodu ile başarısız olmasına neden olduğu *MsgFlags* alanındaki bit konumlarını tanımlar.

Bu alt alan 20-31 bit konumlarını kaplar.

Bu alt alana aşağıdaki ileti işaretleri eklenmiştir:

- MQMF_LAST_MSG_IN_GROUP
- MQMF_LAST_SEGMENT
- MQMF_MSG_IN_GROUP
- MQMF_SEGMENT
- MQMF_SEGMENTATION_ALLOWED
- MQMF_SEGMENTATION_INHIBE

MQMF_ACCEPT_UNSUP_MASK

Bu maske, *MsgFlags* alanında, yerel kuyruk yöneticisi tarafından desteklenmeyen ileti işaretlerinin yine de MQPUT ya da MQPUT1 çağrılarında kabul edildiği bit konumlarını tanımlar. Tamamlama kodu: MQCC_OK.

Bu alt alan 0-11 arasındaki bit konumlarını kaplar.

MQMF_ACCEPT_UNSUP_IF_XMIT_MASK

Bu maske, *MsgFlags* alanında yerel kuyruk yöneticisi tarafından desteklenmeyen ileti işaretlerinin yine de MQPUT ya da MQPUT1 çağrılarında *kabul edildiği* ve her iki koşulun da karşılandığı bit konumlarını tanımlar:

- İleti uzak bir kuyruk yöneticisine gönderiliyor.
- Uygulama iletiyi doğrudan yerel bir iletim kuyruğuna koymuyor (yani, MQOPEN ya da MQPUT1 çağrısında belirtilen nesne tanımlayıcısındaki *ObjectQMgrName* ve *ObjectName* alanları tarafından tanımlanan kuyruk yerel bir iletim kuyruğu değil).

Bu koşullar karşılandıysa, tamamlama kodu MQCC_OK döndürülür ve MQRC_MSG_FLAGS_ERROR neden koduna sahip MQCC_FAILED döndürülür.

Bu alt alan 12 ile 19 arasındaki bit konumlarını kaplar.

MsgFlags alanında kuyruk yöneticisinin tanımadığı işaretler belirtilirse, kuyruk yöneticisi, *MsgFlags* alanını o alt alana ilişkin maskeyle birleştirmek için bit olarak AND işlemini kullanarak her alt alanı

denetler. Bu işlemin sonucu sıfır değilse, daha önce açıklanan tamamlanma kodu ve neden kodları döndürülür.

Veri dönüştürme çıkışı

Bu konu derlemi, veri dönüştürme çıkışına ilişkin arabirimi ve veri dönüştürme gerektiğinde kuyruk yöneticisi tarafından gerçekleştirilen işlemleri açıklar.

Veri dönüştürme hakkında daha fazla bilgi için <https://www.ibm.com/support/pages/node/317869> adresindeki *IBM MQ altında Veri Dönüştürme* adresine bakın.

Veri dönüştürme çıkışı, uygulama iletisi verilerini alan uygulamanın gerektirdiği gösteriye dönüştürmek için MQGET çağrısının işlenmesinin bir parçası olarak çağrılır. Uygulama iletisi verilerinin dönüştürülmesi isteğe bağlıdır; MQGET çağrısında MQGMO_CONVERT seçeneğinin belirtilmesini gerektirir.

Aşağıdaki konular açıklanmıştır:

- MQGMO_CONVERT seçeneğine yanıt olarak kuyruk yöneticisi tarafından gerçekleştirilen işleme; bkz. [“Dönüştürme işlemi” sayfa 895](#).
- Yerleşik bir biçim işlenirken kuyruk yöneticisi tarafından kullanılan işleme kuralları; bu kurallar, kullanıcı tarafından yazılan çıkışlar için de önerilir. Bkz. [“İşleme kuralları” sayfa 896](#).
- Rapor iletilerini dönüştürmeye ilişkin dikkat edilmesi gereken özel noktalar; bkz. [“Rapor iletilerinin dönüştürülmesi” sayfa 900](#).
- Veri dönüştürme çıkışına geçirilen parametreler; bkz. [“MQ_DATA_CONV_EXIT-Veri dönüştürme çıkışı” sayfa 913](#).
- Karakter verilerini farklı gösterimler arasında dönüştürmek için çıkıştan kullanılacak bir çağrı; bkz. [“MQXCNVK-Karakterlerin dönüştürülme” sayfa 907](#).
- Çıkışa özgü veri yapısı parametresi; bkz. [“MQDXP-Veri dönüştürme çıkış parametresi” sayfa 901](#).

Dönüştürme işlemi

Bu bilgiler, kuyruk yöneticisi tarafından MQGMO_CONVERT seçeneğine yanıt olarak gerçekleştirilen işlemleri açıklar.

MQGET çağrısında MQGMO_CONVERT seçeneği belirtilirse ve uygulamaya döndürülecek bir ileti varsa, kuyruk yöneticisi aşağıdaki işlemleri gerçekleştirir:

1. Aşağıdakilerden biri ya da birkaçı doğruysa, dönüştürme gerekmez:

- İleti verileri, MQGET çağrısı yayınlayan uygulamanın gerektirdiği karakter kümesinde ve kodlamada zaten var. Uygulama, çağrısı yayınlamadan önce, MQGET çağrısının **MsgDesc** değiştirgesindeki *CodedCharSetId* ve *Encoding* alanlarını gerekli değerlere ayarlamalıdır.
- İleti verilerinin uzunluğu sıfır.
- MQGET çağrısının **Buffer** değiştirgesinin uzunluğu sıfır.

Bu durumlarda ileti, MQGET çağrısının verildiği uygulamaya dönüştürülmeden döndürülür; **MsgDesc** değiştirgesindeki *CodedCharSetId* ve *Encoding* değerleri, iletideki denetim bilgilerindeki değerlere ayarlanır ve çağrı, aşağıdaki tamamlanma kodu ve neden kodu birleşimlerinden biriyle tamamlanır:

Çizelge 632. Tamamlanma kodu ve neden kodu birleşimleri

Tamamlanma kodu	Neden Kodu
MQCC_OK	MQRC_NONE
MQCC_UYARISI	MQRC_TRUNCATED_MSG_ACCEPTED
MQCC_UYARISI	MQRC_TRUNCATED_MSG_FAILED

Aşağıdaki adımlar, ileti verilerinin karakter kümesi ya da kodlaması **MsgDesc** parametresindeki karşılık gelen değerden farklıysa ve dönüştürülecek veriler varsa gerçekleştirilir:

2. İletideki denetim bilgisindeki *Format* alanında MQFMT_NONE değeri varsa, ileti MQCC_WARNING ve neden kodu MQRC_FORMAT_ERROR ile dönüştürülmez.
Diğer tüm durumlarda dönüştürme işlemi devam eder.
3. İleti kuyruktan kaldırılır ve **Buffer** parametresiyle aynı büyüklükte geçici bir arabelleğe yerleştirilir. Göz atma işlemleri için, ileti kuyruktan kaldırılmak yerine geçici arabelleğe kopyalanır.
4. İletinin arabelleğe sığması için kesilmesi gerekirse, aşağıdaki işlem yapılır:
 - MQGMO_ACCEPT_TRUNCATED_MSG seçeneği belirtilmediyse, ileti MQCC_WARNING ve neden kodu MQRC_TRUNCATED_MSG_FAILED ile dönüştürülmeden döndürülür.
 - MQGMO_ACCEPT_TRUNCATED_MSG seçeneği *belirtildiyse* , tamamlanma kodu MQCC_WARNING olarak ayarlandıysa, neden kodu MQRC_TRUNCATED_MSG_ACCEPTED olarak ayarlanır ve dönüştürme işlemi devam eder.
5. İleti kesilmeden arabelleğe yerleştirilebiliyorsa ya da MQGMO_ACCEPT_TRUNCATED_MSG seçeneği belirtildiyse, aşağıdaki işlem yapılır:
 - Biçim yerleşik bir biçimse, arabellek kuyruk yöneticisinin veri dönüştürme hizmetine geçirilir.
 - Biçim yerleşik bir biçim değilse, arabellek, biçimle aynı adı taşıyan kullanıcı tarafından yazılmış bir çıkışa geçirilir. Çıkış bulunamazsa, ileti dönüştürülmeden döndürülür; tamamlanma kodu MQCC_WARNING ve neden kodu MQRC_FORMAT_ERROR.

Hata ortaya çıkmazsa, veri dönüştürme hizmetinden ya da kullanıcı tarafından yazılan çıkışın çıkışı dönüştürülen iletidir ve MQGET çağrısını yayınlayan uygulamaya döndürülecek tamamlanma kodu ve neden kodu döndürülür.
6. Dönüştürme başarılı olursa, kuyruk yöneticisi dönüştürülen iletiyi uygulamaya döndürür. Bu durumda, MQGET çağrısının döndürdüğü tamamlanma kodu ve neden kodu aşağıdaki birleşimlerden biridir:

Çizelge 633. Tamamlanma kodu ve neden kodu birleşimleri

Tamamlanma kodu	Neden Kodu
MQCC_OK	MQRC_NONE
MQCC_UYARISI	MQRC_TRUNCATED_MSG_ACCEPTED

Ancak, dönüştürme kullanıcı tarafından yazılan bir çıkış tarafından gerçekleştirilirse, dönüştürme başarılı olsa da, diğer neden kodları döndürülebilir.

Dönüştürme başarısız olursa, kuyruk yöneticisi, **MsgDesc** değiştirgesindeki *CodedCharSetId* ve *Encoding* alanları iletideki denetim bilgilerindeki değerlere ve MQCC_WARNING tamamlanma koduyla uygulamaya dönüştürülmemiş iletiyi döndürür.

İşleme kuralları

Yerleşik bir biçimi dönüştürürken, kuyruk yöneticisi açıklanan işleme kurallarını izler.

Kuyruk yöneticisi tarafından zorlanmamasına rağmen, kullanıcı tarafından yazılan çıkışlar da bu kurallara uymalıdır. Kuyruk yöneticisi tarafından dönüştürülen yerleşik biçimler şunlardır:

- MQFMT_ADMIN
- MQFMT_CICS (yalnızca z/OS)
- MQFMT_COMMAND_1
- MQFMT_COMMAND_2
- MQFMT_DEAD_LETTER_HEADER
- MQFMT_DIST_HEADER
- MQFMT_EVENT sürüm 1
- MQFMT_EVENT sürüm 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 (yalnızca z/OS)
- MQFMT_XMIT_Q_HEADER

1. İleti dönüştürme sırasında genişler ve **Buffer** değıştirgesinin büyüklüğünü aşarsa, aşağıdaki işlem gerçekleştirilir:
 - MQGMO_ACCEPT_TRUNCATED_MSG seçeneđi belirtilmediyse, ileti MQCC_WARNING ve neden kodu MQRC_CONVERTED_MSG_TOO_BIG ile dönüştürülmeden döndürülür.
 - MQGMO_ACCEPT_TRUNCATED_MSG seçeneđi *was* belirtilirse, ileti kesilir, tamamlanma kodu MQCC_WARNING olarak ayarlanırsa, neden kodu MQRC_TRUNCATED_MSG_ACCEPTED olarak ayarlanır ve dönüştürme işlemi devam eder.
2. Kesme oluşursa (dönüştürme öncesinde ya da sırasında), **Buffer** değıştirgesinde döndürülen geçerli byte sayısı arabelleđin uzunluđundan az olabilir.

Örneđin, 4 baytlık bir tamsayı ya da bir DBCS karakteri arabelleđin sonunu takarsa bu durum oluşabilir. Eksik bilgi öđesi dönüştürülmedi ve döndürülen iletideki byte 'lar geçerli bilgi içermiyor. Dönüştürme sırasında dönüştürülmeden önce kesilen bir ileti küçültülürse de bu durum oluşabilir.

Döndürülen geçerli bayt sayısı arabelleđin uzunluđundan azsa, arabelleđin sonundaki kullanılmayan bayt sayısı boş değere ayarlanır.
3. Bir dizi ya da dizgi arabelleđin sonunu tamamlarsa, verilerin çođu dönüştürülür; yalnızca, eksik olan belirli dizi öđesi ya da DBCS karakteri dönüştürülmez; önceki dizi öđeleri ya da karakterler dönüştürülür.
4. Kesme oluşursa (dönüştürme öncesinde ya da sırasında), **DataLength** parametresi için döndürülen uzunluk, kesilmeden önceki dönüştürülmemiş iletinin uzunluđudur.
5. Dizgiler tek baytlık karakter takımları (SBCS), çift baytlık karakter takımları (DBCS) ya da çok baytlık karakter takımları (MBCS) arasında dönüştürüldüğünde, dizgiler genişletilebilir ya da daralabilir.
 - PCF biçimlerinde MQFMT_ADMIN, MQFMT_EVENT ve MQFMT_PCF, MQCFST ve MQCFSL yapılarındaki dizgiler, dönüştürmeden sonra dizgiyi sığdırmak için gereken şekilde genişletilir ya da daralır.

Dizgi listesi yapısı MQCFSL için, listedeki dizgiler farklı miktarlarda genişletilebilir ya da daraltabilir. Bu durumda, kuyruk yöneticisi kısa dizgileri, dönüşümden sonraki en uzun dizgiyle aynı uzunlukta olacak şekilde boşluklarla doldurur.

 - MQFMT_REF_MSG_HEADER biçiminde, SrcEnvOffset, SrcNameOffset, DestEnvOffset ve DestNameOffset alanları tarafından adreslenen dizgiler, dönüştürmeden sonra dizgileri sığdırmak için gereken şekilde genişler ya da daraltır.
 - MQFMT_RF_HEADER biçiminde, NameValueString alanı, dönüşümden sonra ad-deđer çiftlerini barındırmak için gereken şekilde genişletir ya da daraltır.
 - Sabit alan büyüklüklü yapılarla, kuyruk yöneticisi, önemli bir bilgi kaybolmaması koşuluyla, dizgilerin sabit alanları içinde genişletilmesine ya da daralmasına izin verir. Bu bağlamda, alandaki ilk boş karakteri izleyen sondaki boşluklar ve karakterler önemsiz olarak değeriendirilir.
 - Dizgi genişler, ancak alandaki dönüştürülen dizgiyi barındırmak için yalnızca önemsiz karakterlerin atılması gerekirse, dönüştürme başarılı olur ve çağrı MQCC_OK ve neden kodu MQRC_NONE ile tamamlanır (başka hata olmadığı varsayılarak).

- Dizgi genişler, ancak dönüştürülen dizginin alana sığması için önemli karakterlerin atılmasını gerektiriyorsa, ileti dönüştürülmeden döndürülür ve çağrı MQCC_WARNING ve MQRC_CONVERTED_STRING_TOO_BIG neden koduyla tamamlanır.

Not: Neden kodu MQRC_CONVERTED_STRING_TOO_BIG, MQGMO_ACCEPT_TRUNCATED_MSG seçeneğinin belirtilip belirtilmediğine bakılmaksızın bu durumda sonuçlanır.

- Dizilim kasılırsa, kuyruk yöneticisi dizgiyi alan uzunluğuna kadar boşluklarla doldurur.

6. Bir ya da daha çok MQ üstbilgi yapısından ve ardından kullanıcı verilerinden oluşan iletiler için, üstbilgi yapılarından biri ya da daha fazlası dönüştürülebilir, ancak iletinin geri kalanı dönüştürülemez. Ancak (iki istisna dışında) her üstbilgi yapısındaki *CodedCharSetId* ve *Encoding* alanları, üstbilgi yapısını izleyen verilerin karakter kümesini ve kodlamasını her zaman doğru olarak gösterir.

Bu iki kural dışı durum, bu yapılardaki *CodedCharSetId* ve *Encoding* alanlarındaki değerlerin önemli olmadığı MQCIH ve MQIIH yapılarıdır. Bu yapılar için, yapıyı izleyen veriler MQCIH ya da MQIIH yapısıyla aynı karakter kümesi ve kodlamadır.

7. Alınmakta olan iletinin denetim bilgilerindeki ya da **MsgDesc** değiştirgesindeki *CodedCharSetId* ya da *Encoding* alanları tanımsız ya da desteklenmeyen değerleri belirtiyorsa, kuyruk yöneticisi, iletiyi dönüştürürken tanımlanmamış ya da desteklenmeyen değerlerin kullanılması gerekmiyorsa hatayı yoksayabilir.

Örneğin, iletideki *Encoding* alanı desteklenmeyen bir kayar noktalı sayı kodlaması belirtiyorsa, ancak ileti yalnızca tamsayı verileri içeriyorsa ya da dönüştürme gerektirmeyen kayan nokta verileri içeriyorsa (kaynak ve hedef kayan nokta kodlamaları aynı olduğu için), hata tanılanmayabilir.

Hata tanılanırsa, ileti MQCC_WARNING ve MQRC_SOURCE_*_ERROR ya da MQRC_TARGET_*_ERROR neden kodlarından biriyle (uygun şekilde) dönüştürülmez; **MsgDesc** değiştirgesindeki *CodedCharSetId* ve *Encoding* alanları, iletideki denetim bilgilerindeki değerlere ayarlanır.

Hata tanı konmazsa ve dönüştürme başarıyla tamamlanırsa, **MsgDesc** değiştirgesindeki *CodedCharSetId* ve *Encoding* alanlarında döndürülen değerler, MQGET çağrısına ilişkin uygulamanın belirttiği değerlerdir.

8. Her durumda, ileti uygulamaya dönüştürülmeden döndürülürse, tamamlanma kodu MQCC_WARNING olarak ayarlanır ve **MsgDesc** parametresindeki *CodedCharSetId* ve *Encoding* alanları, dönüştürülmemiş verilere uygun değerlere ayarlanır. Bu, MQFMT_NONE için de yapılır.

Reason parametresi, iletinin de kesilmesi gerekmedikçe, dönüştürmenin neden gerçekleştirilemediğini gösteren bir koda ayarlanır; kesmeyle ilgili neden kodları, dönüştürmeyle ilgili neden kodlarından önceliklidir. (Kesilmiş bir iletinin dönüştürülüp dönüştürülmediğini saptamak için, **MsgDesc** değiştirgesindeki *CodedCharSetId* ve *Encoding* alanlarında döndürülen değerleri denetleyin.)

Bir hata tanısı konusunda, belirli bir neden kodu döndürülür ya da genel neden kodu MQRC_NOT_CONVERTED döndürülür. Döndürülen neden kodu, temel veri dönüştürme hizmetinin tanılama yeteneklerine bağlıdır.

9. Tamamlama kodu MQCC_WARNING döndürülürse ve birden çok neden kodu ilgiliyse, öncelik sırası aşağıdaki gibidir:

- a. Aşağıdaki nedenler diğerlerinden önceliklidir; bu gruptaki nedenlerden yalnızca biri ortaya çıkabilir:

- MQRC_SIGNAL_REQUEST_ACCEPTED
- MQRC_TRUNCATED_MSG_ACCEPTED

- b. Geri kalan neden kodları içindeki öncelik sırası tanımlı değil.

10. MQGET çağrısı tamamlandığında:

- Aşağıdaki neden kodu, iletinin başarıyla dönüştürüldüğünü gösterir:

- MQRC_NONE

- Aşağıdaki neden kodları *Olabilir* iletisinin başarıyla dönüştürüldüğünü gösterir (öğrenmek için **MsgDesc** parametresindeki *CodedCharSetId* ve *Encoding* alanlarını denetleyin):

- MQRC_MSG_MARKED_BROWSE_CO_OP
- MQRC_TRUNCATED_MSG_ACCEPTED
- Diğer tüm neden kodları iletinin dönüştürülmediğini gösterir.

Aşağıdaki işleme yerleşik biçimlere özgüdür; kullanıcı tanımlı biçimler için geçerli değildir:

11. Aşağıdaki biçimler dışında:

- MQFMT_ADMIN
- MQFMT_COMMAND_1
- MQFMT_COMMAND_2
- MQFMT_EVENT
- MQFMT_IMS_VAR_STRING
- MQFMT_PCF
- MQFMT_STRING

Yerleşik biçimlerin hiçbiri, kuyruk adlarında geçerli olan karakterler için SBCS karakterleri olmayan karakter kümelerinden ya da karakter takımlarına dönüştürülemez. Böyle bir dönüştürme gerçekleştirme girişiminde bulunulursa, ileti MQCC_WARNING tamamlanma kodu ve MQRC_SOURCE_CCSID_ERROR ya da MQRC_TARGET_CCSID_ERROR neden kodu ile dönüştürülmez.

Unicode karakter kümesi UTF-16 , kuyruk adlarında geçerli olan karakterler için SBCS karakterleri olmayan bir karakter kümesi örneğidir.

12. Yerleşik bir biçime ilişkin ileti verileri kesilirse, ileti içindeki dizgi uzunluğu ya da öge ya da yapı sayısı içeren alanlar, uygulamaya gerçekte döndürülen verilerin uzunluğunu yansıtacak şekilde ayarlanmaz; ileti verileri içinde bu tür alanlar için döndürülen değerler, *kesilmeden önce*ileti için geçerli olan değerlerdir.

Kesilmiş MQFMT_ADMIN iletisi gibi iletileri işlerken, uygulamanın döndürülen verilerin sonundan sonraki verilere erişmeye çalışmadığını doğrulayın.

13. Biçim adı MQFMT_DEAD_LETTER_HEADER ise, ileti verileri bir MQDLH yapısıyla başlar ve bunu büyük olasılıkla sıfır ya da daha fazla byte uygulama iletisi verisi izler. Uygulama iletisi verilerinin biçimi, karakter kümesi ve kodlaması, iletinin başında MQDLH yapısındaki Format, CodedCharSetId ve Encoding alanları tarafından tanımlanır. MQDLH yapısı ve uygulama iletisi verileri farklı karakter kümeleri ve kodlamaları içerebileceğinden, MQDLH yapısının ve uygulama iletisi verilerinin biri, diğeri ya da her ikisi de dönüştürme gerektirebilir.

Kuyruk yöneticisi önce MQDLH yapısını gerektiği şekilde dönüştürür. Dönüştürme başarılı olursa ya da MQDLH yapısı dönüştürme gerektirmeyse, kuyruk yöneticisi, uygulama iletisi verilerinin dönüştürülmesinin gerekli olup olmadığını görmek için MQDLH yapısındaki CodedCharSetId ve Encoding alanlarını denetler. Dönüştürme gerekiyorsa, kuyruk yöneticisi, MQDLH yapısında Format alanı tarafından belirtilen adla kullanıcı tarafından yazılan çıkışı çağırır ya da dönüştürmenin kendisini gerçekleştirir (Format yerleşik bir biçimin adıdır).

MQGET çağrısı bir MQCC_WARNING tamamlanma kodu döndürürse ve neden kodu dönüştürmenin başarısız olduğunu gösteren kodlardan biriyse, aşağıdakilerden biri geçerlidir:

- MQDLH yapısı dönüştürülemedi. Bu durumda, uygulama ileti verileri de dönüştürülmez.
- MQDLH yapısı dönüştürüldü, ancak uygulama iletisi verileri dönüştürülmedi.

Uygulama, önceden geçerli olan değerleri saptamak için, **MsgDesc** değiştirgesindeki CodedCharSetId ve Encoding alanlarında ve MQDLH yapısında döndürülen değerleri inceleyebilir.

14. Biçim adı MQFMT_XMIT_Q_HEADER ise, ileti verileri bir MQXQH yapısıyla başlar ve bunu büyük olasılıkla sıfır ya da daha fazla byte ek veri izler. Bu ek veriler genellikle uygulama iletisi verileridir (sıfır uzunluklu olabilir), ancak ek verilerin başında bir ya da daha fazla MQ üstbilgi yapısı da bulunabilir.

MQXQH yapısı, kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır. MQXQH yapısını izleyen verilerin biçimi, karakter kümesi ve kodlaması, MQXQH içinde bulunan MQMD yapısındaki

Format, CodedCharSetIdve Encoding alanları tarafından verilir. Sonraki her MQ üstbilgi yapısı için, yapıdaki Format, CodedCharSetIdve Encoding alanları o yapıyı izleyen verileri açıklar; bu veriler başka bir MQ üstbilgi yapısı ya da uygulama iletisi verileridir.

Bir MQFMT_XMIT_Q_HEADER iletisi için MQGMO_CONVERT seçeneği belirtilirse, uygulama iletisi verileri ve MQ üstbilgi yapılarının bazıları dönüştürülür, *ancak MQXQH yapısındaki verilerdeğildir*. MQGET çağrısından döndüğünüzde:

- **MsgDesc** değiştirgesindeki Format, CodedCharSetIdve Encoding alanlarının değerleri, uygulama iletisi verilerini değil, MQXQH yapısındaki verileri açıklar; bu nedenle değerler, MQGET çağrısının yayınlandığı uygulamanın belirttiği değerlerle aynı değildir.

Bunun sonucu, MQGMO_CONVERT seçeneği belirtilen bir iletim kuyruğundan sürekli olarak ileti alan bir uygulamanın, **MsgDesc** değiştirgesindeki CodedCharSetId ve Encoding alanlarını, her MQGET çağrısından önce uygulama iletisi verileri için gereken değerlere geri çevirmesi gerektirir.

- Son MQ üstbilgi yapısındaki Format, CodedCharSetIdve Encoding alanlarının değerleri, uygulama iletisi verilerini açıklar. Başka bir MQ üstbilgi yapısı yoksa, uygulama iletisi verileri MQXQH yapısı içindeki MQMD yapısında bu alanlar tarafından tanımlanır. Dönüştürme başarılı olursa, değerler, MQGET çağrısı yayınlayan uygulama tarafından **MsgDesc** değiştirgesinde belirtilenlerle aynı olur.

İleti bir dağıtım listesi iletirse, MQXQH yapısını bir MQDH yapısı izler (bunun yanı sıra MQOR ve MQPMR kayıtları dizileri); bu yapıyı sıfır ya da daha fazla MQ üstbilgi yapısı ve sıfır ya da daha fazla uygulama iletisi verisi izleyebilir. MQXQH yapısı gibi, MQDH yapısı da kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır ve MQGMO_CONVERT seçeneği belirtilse bile MQGET çağrısında dönüştürülmez.

Daha önce açıklanan MQXQH ve MQDH yapılarının işlenmesi, öncelikle ileti kuyruklarından ileti aldıklarında ileti kanalı aracılığı tarafından kullanılmak üzere tasarlanmıştır.

Rapor iletilerinin dönüştürülmesi

Genel olarak bir rapor iletisi, özgün iletiyi gönderen tarafından belirtilen rapor seçeneklerine göre çeşitli miktarlarda uygulama iletisi verisi içerebilir. Ancak, bir etkinlik raporu, değişmezde * _WITH_DATA ' dan söz eden rapor seçeneği olmadan veri içerebilir.

Özellikle, bir rapor iletisi aşağıdakilerden birini içerebilir:

1. Uygulama iletisi verisi yok
2. Özgün iletideki bazı uygulama iletisi verileri

Özgün iletiyi gönderen kişi MQRO_ * _WITH_DATA belirtiyorsa ve ileti 100 byte 'tan uzunsa bu durum oluşur.

3. Özgün iletideki tüm uygulama iletisi verileri

Bu durum, özgün iletiyi gönderen kişi MQRO_ * _WITH_FULL_DATA değerini belirttiğinde ya da MQRO_ * _WITH_DATA değerini belirttiğinde ve ileti 100 byte ya da daha kısa olduğunda oluşur.

Kuyruk yöneticisi ya da ileti kanalı aracısı bir rapor iletisi oluşturduğunda, biçim adını özgün iletiden rapor iletisindeki denetim bilgisindeki *Format* alanına kopyalar. Bu nedenle, rapor iletisindeki biçim adı, rapor iletisinde gerçekte var olan uzunluktan farklı bir veri uzunluğu anlamına gelebilir (daha önce 1 ve 2. durumlar).

Rapor iletisi alındığında MQGMO_CONVERT seçeneği belirtilirse:

- Daha önce 1. vaka için, veri dönüştürme çıkışı çağrılmaz (çünkü rapor iletisinde veri yoktur).
- Daha önce 3. durumda, biçim adı ileti verilerinin uzunluğunu doğru olarak belirtir.
- Ancak daha önce 2. durumda, biçim adının belirttiği uzunluktan *daha kısa* bir iletiyi dönüştürmek için veri dönüştürme çıkışı çağrılır.

Ayrıca, çıkışa geçirilen neden kodu genellikle MQRC_NONE olur (yani, neden kodu iletinin kesildiğini göstermez). Bunun nedeni, ileti verilerinin MQGET çağrısına yanıt olarak alıcının kuyruk yöneticisi tarafından değil, rapor iletisinin *gönderen* tarafından kesilmesiydi.

Bu olasılıklar nedeniyle, veri dönüştürme çıkışı, iletilen verilerin uzunluğunu saptamak için biçim adını kullanmamalıdır; bunun yerine, çıkışın sağlanan verilerin uzunluğunu denetmesi ve biçim adının belirttiği uzunluktan daha az veri dönüştürmeye hazır olması gerekir. Veriler başarıyla dönüştürülebiliyorsa, çıkış tarafından tamamlama kodu MQCC_OK ve neden kodu MQRC_NONE döndürülmelidir. Dönüştürülecek ileti verilerinin uzunluğu, çıkışa **InBufferLength** değiştirgesi olarak geçirilir.

Ürüne duyarlı programlama arabirimi

MQDXP-Veri dönüştürme çıkış parametresi

MQDXP yapısı, MQGET çağrısının işlenmesinin bir parçası olarak ileti verilerini dönüştürmek için çıkış çağrıldığında kuyruk yöneticisinin veri dönüştürme çıkışına aktardığı bir parametredir. Veri dönüştürme çıkışının ayrıntıları için MQ_DATA_CONV_EXIT çağrısının tanımına bakın.

MQDXP 'deki karakter verileri yerel kuyruk yöneticisinin karakter kümesinde; bu, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından verilir. MQDXP 'deki sayısal veriler yerli makine kodlamasında; bu, MQENC_NATIVE tarafından verilir.

Çıkışla yalnızca MQDXP 'deki *DataLength*, *CompCode*, *Reason* ve *ExitResponse* alanları değiştirilebilir; diğer alanlarda yapılan değişiklikler yoksayıdır. Ancak, dönüştürülmekte olan ileti mantıksal iletinin yalnızca bir kısmını içeren bir bölümse, *DataLength* alanı değiştirilemez.

Denetim, çıkıştan kuyruk yöneticisine döndüğünde, kuyruk yöneticisi MQDXP 'de döndürülen değerleri denetler. Döndürülen değerler geçerli değilse, kuyruk yöneticisi *ExitResponse* ; içinde MQXDR_CONVERSION_FAILED değerini döndürmüş gibi işlemeye devam eder. Ancak kuyruk yöneticisi, bu durumda çıkışın döndürdüğü *CompCode* ve *Reason* alanlarının değerlerini yoksayar ve bunun yerine, bu alanların çıkışta *giriş* olarak sahip olduğu değerleri kullanır. MQDXP 'de aşağıdaki değerler bu işlemin gerçekleşmesine neden olur:

- *ExitResponse* alanı MQXDR_OK değil, MQXDR_CONVERSION_FAILED değil
- *CompCode* alanı MQCC_OK değil ve MQCC_WARNING değil
- *DataLength* alan sıfırdan küçük ya da *DataLength* alanı, dönüştürülmekte olan ileti mantıksal iletinin yalnızca bir kısmını içeren bir bölüm olduğunda değişti.

Aşağıdaki tablo, yapıdaki alanları özetler.

Çizelge 634. MQDXP 'deki Alanlar		
Alan	Açıklama	Konu
<i>StrucId</i>	Yapı tanıttıcısı	<u>StrucId</u>
<i>Version</i>	Yapı sürümü numarası	<u>Sürüm</u>
<i>AppOptions</i>	Uygulama seçenekleri	<u>AppOptions</u>
<i>Encoding</i>	Uygulamanın gerektirdiği sayısal kodlama	<u>Kodlama</u>
<i>CodedCharSetId</i>	Uygulamanın gerektirdiği karakter kümesi	<u>CodedCharSetId</u>
<i>DataLength</i>	İleti verilerinin bayt cinsinden uzunluğu	<u>DataLength</u>
<i>CompCode</i>	Tamamlanma kodu	<u>CompCode</u>
<i>Reason</i>	Neden kodu niteleyici <i>CompCode</i>	<u>Neden</u>
<i>ExitResponse</i>	Çıkıştan yanıt	<u>ExitResponse</u>
<i>Hconn</i>	Bağlantı tanıttıcı	<u>Hconn</u>

Çizelge 634. MQDXP 'deki Alanlar (devamı var)

Alan	Açıklama	Konu
<i>pEntryPoints</i>	MQIEP yapısının adresi	pEntryNoktalar

Alanlar

MQDXP yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır.

AppOptions

Tip: MQLONG

Bu, MQGET çağrısı yayınlayan uygulamanın belirttiği MQGMO yapısının *Options* alanının bir kopyasıdır. Çıkışın, MQGMO_ACCEPT_TRUNCATED_MSG seçeneğinin belirtilip belirtilmediğini saptamak için bunları incelemesi gerekebilir.

Bu, çıkışa bir giriş alanıdır.

CodedCharSetId

Tip: MQLONG

Bu, MQGET çağrısını yayınlayan uygulamanın gerektirdiği karakter kümesinin kodlanmış karakter kümesi tanıtıcısıdır; daha fazla ayrıntı için MQMD yapısındaki *CodedCharSetId* alanına bakın. Uygulama MQGET çağrısında MQCCSI_Q_MGR özel değerini belirtiyorsa, kuyruk yöneticisi çıkışı çağırılmadan önce kuyruk yöneticisi tarafından kullanılan karakter kümesinin gerçek karakter kümesi tanıtıcısı olarak değiştirir.

Dönüştürme başarılı olursa, çıkışın bunu ileti tanımlayıcısındaki *CodedCharSetId* alanına kopyalaması gerekir.

Bu, çıkışa bir giriş alanıdır.

CompCode

Tip: MQLONG

Çıkış çağrıldığında, çıkış bir şey yapmazsa, MQGET çağrılmasını yayınlayan uygulamaya döndürülen tamamlanma kodunu içerir. Her zaman MQCC_WARNING olur; ileti kesildi ya da ileti dönüştürme gerektiriyor ve bu işlem henüz yapılmadı.

Çıkışta, bu alan MQGET çağrısının **CompCode** değiştirilmesinde uygulamaya döndürülecek tamamlanma kodunu içerir; yalnızca MQCC_OK ve MQCC_WARNING geçerlidir. Çıkışın çıkışta bu alanı nasıl ayarlayabileceğine ilişkin öneriler için *Reason* alanının açıklamasına bakın.

Bu, çıkışa bir giriş/çıkış alanıdır.

DataLength

Tip: MQLONG

Çıkış çağrıldığında, bu alan uygulama ileti verilerinin özgün uzunluğunu içerir. İleti uygulama tarafından sağlanan arabelleğe sığacak şekilde kesildiyse, çıkışa sağlanan iletinin boyutu *DataLength* değerinden *daha küçük* olur. Çıkışa sağlanan iletinin boyutu, oluşan herhangi bir kesilmeden bağımsız olarak, her zaman çıkışın **InBufferLength** parametresi tarafından verilir.

Kesme, çıkışa girişte MQRC_TRUNCATED_MSG_ACCEPTED değerini içeren *Reason* alanıyla gösterilir.

Çoğu dönüştürmenin bu uzunluğu değiştirmesi gerekmez, ancak gerekirse bir çıkış bunu yapabilir; çıkış tarafından ayarlanan değer, MQGET çağrısının **DataLength** değiştirilmesinde uygulamaya döndürülür. Ancak, dönüştürülmekte olan ileti mantıksal iletinin yalnızca bir kısmını içeren bir bölümse, bu uzunluk değiştirilemez. Bunun nedeni, uzunluğun değiştirilmesinin mantıksal iletideki sonraki kesimlerin görece konumlarının yanlış olmasına neden olmasıdır.

Çıkış verilerin uzunluğunu değiştirmek isterse, kuyruk yöneticisinin ileti verilerinin uygulamanın arabelleğine *dönüştürülmemiş* verilerin uzunluğuna dayalı olarak uyup uymadığına önceden karar verdiğini unutmayın. Bu karar, iletinin kuyruktan kaldırılıp kaldırılmayacağını (ya da göz atma imlecinin bir göz atma isteği için taşınacağını) ve dönüştürmenin neden olduğu veri uzunluğundaki herhangi bir

değişiklikten etkilenmeyeceğini belirler. Bu nedenle, dönüştürme çıktıların uygulama ileti verilerinin uzunluğunun değişmesine neden olmaması önerilir.

Karakter dönüştürme bir uzunluk değişikliğini gösteriyorsa, bir dizilim bayt cinsinden aynı uzunlukta başka bir dizgiye dönüştürülebilir, sondaki boşluklar kesilebilir ya da gerektiğinde boşluklarla doldurulur.

İleti uygulama ileti verisi içermezse çıkış çağrılmaz; bu nedenle *DataLength* her zaman sıfırdan büyüktür.

Bu, çıkışa bir giriş/çıkış alanıdır.

Encoding

Tip: MQLONG

Uygulamanın gerektirdiği sayısal kodlama.

Bu, MQGET çağrısının yayınlanması için uygulamanın gerektirdiği sayısal kodlamadır; daha fazla ayrıntı için MQMD yapısındaki *Encoding* alanına bakın.

Dönüştürme başarılı olursa, çıkış bunu ileti tanımlayıcısındaki *Encoding* alanına kopyalar.

Bu, çıkışa bir giriş alanıdır.

ExitOptions

Tip: MQLONG

Bu ayrılmış bir alandır; değeri 0.

ExitResponse

Tip: MQLONG

Çıkıştan cevap geldi. Bu, çıkış tarafından, dönüştürmenin başarılı olduğunu ya da başka bir şekilde gerçekleştiğini belirtmek için ayarlanır. Aşağıdakilerden biri olmalıdır:

MQXDR_OK

Dönüştürme başarılı oldu.

Çıkış bu değeri belirtiyorsa, kuyruk yöneticisi MQGET çağrısı yayınlayan uygulamaya aşağıdakileri döndürür:

- Çıkıştaki *CompCode* alanının değeri
- Çıkıştaki *Reason* alanının değeri
- Çıkıştaki *DataLength* alanının değeri
- Çıkışın çıkış arabelleğinin içeriği *OutBuffer*. Döndürülen bayt sayısı, çıkışın **OutBufferLength** parametresinden daha küçük ve çıkıştaki *DataLength* alanının değeridir.

Çıkışın ileti tanımlayıcı parametresindeki *Encoding* ve *CodedCharSetId* alanları *her ikisi* değiştirilmediyse, kuyruk yöneticisi şunları döndürür:

- Çıkışa ilişkin giriş MQDXP yapısındaki *Encoding* ve *CodedCharSetId* alanlarının değeri.

Çıkışın ileti tanımlayıcı parametresindeki *Encoding* ve *CodedCharSetId* alanlarından biri ya da her ikisi değiştirildiyse, kuyruk yöneticisi şunları döndürür:

- Çıkışta çıkışın ileti tanımlayıcı parametresindeki *Encoding* ve *CodedCharSetId* alanlarının değeri

MQXDR_CONVERSION_FAILED

Dönüştürme başarısız oldu.

Çıkış bu değeri belirtiyorsa, kuyruk yöneticisi MQGET çağrısı yayınlayan uygulamaya aşağıdakileri döndürür:

- Çıkıştaki *CompCode* alanının değeri
- Çıkıştaki *Reason* alanının değeri
- Çıkışa giriş üzerindeki *DataLength* alanının değeri

- Çıkışın giriş arabelleğinin içeriği *InBuffer*. **InBufferLength** parametresi tarafından döndürülen bayt sayısı

Çıkış *InBuffer*değiştirdiyse, sonuçlar tanımsız olur.

ExitResponse , çıkıştan bir çıkış alanıdır.

Hconn

Tip: MQHCONN

Bu, MQXCNCV çağrısında kullanılacak bir bağlantı tanıtıcısıdır. Bu tanıtıcı, MQGET çağrısını yayınlayan uygulama tarafından belirtilen tanıtıcı ile aynı olmayabilir.

pEntryPoints

Tip: PMQIEP

MQI ve DCI çağrılarının yapılabileceği bir MQIEP yapısının adresi.

Reason

Tip: MQLONG

Neden kodu niteleyici *CompCode*.

Çıkış çağrıldığında, çıkış hiçbir şey yapmayı seçmezse, MQGET çağrısı yayınlayan uygulamaya döndürülen neden kodunu içerir. Olası değerler arasında, iletinin uygulama tarafından sağlanan arabelleğe sığması için kesildiğini belirten MQRC_TRUNCATED_MSG_ACCEPTED ve MQRC_NOT_CONVERTED yer alır; bu, iletinin dönüştürme gerektirdiğini, ancak bunun henüz yapılmadığını gösterir.

Çıkışın çıkışında, bu alan MQGET çağrısının **Reason** değıştirgesinde uygulamaya döndürüleceği nedeni içerir; aşağıdakiler önerilir:

- *Reason* , çıkışa girişte MQRC_TRUNCATED_MSG_ACCEPTED değeri kabul edildiyse, dönüştürme başarılı ya da başarısız olursa olsun, *Reason* ve *CompCode* alanları değıştirilmemelidir.

(*CompCode* alanı MQCC_OK değilse, iletiyi alan uygulama, ileti tanımlayıcıdaki *Encoding* ve *CodedCharSetId* değeri istenen değeri karşılaştırarak bir dönüştürme hatası saptayabilir; buna karşılık, uygulama kesilmiş bir iletiyi arabelleğe uygun bir iletiden ayıramaz. Bu nedenle, dönüştürme başarısızlığını gösteren nedenlerden herhangi biri yerine MQRC_TRUNCATED_MSG_ACCEPTED (MQRC_TRUNCATED_MSG_ACCEPTED) döndürülmelidir.)

- *Reason* ' in çıkışa girişte başka bir değeri varsa:
 - Dönüştürme başarılı olursa, *CompCode* MQCC_OK olarak ayarlanmalı ve *Reason* MQRC_NONE olarak ayarlanmalıdır.
 - Dönüştürme başarısız olursa ya da ileti genişler ve arabelleğe sığacak şekilde kesilmesi gerekirse, *CompCode* MQCC_WARNING (ya da değıştirilmeden bırakılmalıdır) olarak ayarlanmalı ve *Reason* , hatanın türünü belirtmek için listelenen değeri birine ayarlanmalıdır.

Dönüştürme sonrası ileti arabellek için çok büyükse, yalnızca MQGET çağrısı yayınlayan uygulama MQGMO_ACCEPT_TRUNCATED_MSG seçeneğini belirtiyorsa bu iletinin kesilmesi gerektiğini unutmayın:

- Bu seçenek belirtildiyse, MQRC_TRUNCATED_MSG_ACCEPTED (mqrc_truncated_msg_accepted) nedeni döndürülür.
- Bu seçenek belirlenmediyse, ileti dönüştürülmeden döndürülür; neden kodu MQRC_CONVERTED_MSG_TOO_BIG.

Listelenen neden kodlarının, dönüştürmenin başarısız olma nedenini belirtmek için çıkış tarafından kullanılması önerilir; ancak çıkış, uygun görülürse, MQRC_* kodları kümesinden başka değeri döndürebilir. Ayrıca, MQRC_APPL_FIRST-MQRC_APPL_LAST arasındaki değeri aralığı, çıkışın MQGET çağrısı yayınlayan uygulamayla iletişim kurmak istediği koşulları belirtmek için çıkış tarafından kullanılmak üzere ayrılır.

Not: İleti başarıyla dönüştürülemezse, kuyruk yöneticisinin dönüştürülmemiş iletiyi döndürmesi için çıkışın *ExitResponse* alanında MQXDR_CONVERSION_FAILED döndürmesi gerekir. Bu, *Reason* alanında döndürülen neden kodundan bağımsız olarak geçerlidir.

MQRC_APPL_FIRST

(900, X'384 ') Uygulama tanımlı neden kodu için daha düşük değer.

MQRC_APPL_LAST

(999, X'3E7') Uygulama tanımlı neden kodu için en yüksek değer.

MQRC_CONVERTED_MSG_TOO_BIG

(2120, X'848 ') Dönüştürülen veriler arabellek için çok büyük.

MQRC_NOT_CONVERTED

(2119, X'847 ') İleti verileri dönüştürülmedi.

MQRC_SOURCE_CCSID_HATA

(2111, X'83F') Kaynak kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_SOURCE_DECIMAL_ENC_HATA

(2113, X'841 ') İletideki paketli ondalık kodlama tanınmıyor.

MQRC_SOURCE_FLOAT_ENC_HATA

(2114, X'842 ') İletide kayan nokta kodlaması tanınmıyor.

MQRC_SOURCE_INTEGER_ENC_HATA

(2112, X'840 ') Kaynak tamsayı kodlaması tanınmıyor.

MQRC_TARGET_CCSID_HATA

(2115, X'843 ') Hedef kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_TARGET_DECIMAL_ENC_HATA

(2117, X'845 ') Alıcı tarafından belirtilen Paketlenmiş ondalık kodlama tanınmıyor.

MQRC_TARGET_FLOAT_ENC_HATA

(2118, X'846 ') Alıcı tarafından belirtilen kayan nokta kodlaması tanınmıyor.

MQRC_TARGET_INTEGER_ENC_HATA

(2116, X'844 ') Hedef tamsayı kodlaması tanınmıyor.

MQRC_TRUNCATED_MSG_ACCEPTED

(2079, X'81F') İzlenen ileti döndürüldü (işleme tamamlandı).

Bu, çıkışa bir giriş/çıkış alanıdır.

StrucId

Tip: MQCHAR4

Yapı tanıtıcısı. Değer şöyle olmalıdır:

MQDXP_STRUC_ID

Veri dönüştürme çıkış parametresi yapısına ilişkin tanıtıcı.

C programlama dili için, MQDXP_STRUC_ID_ARRAY değişmezi de tanımlanır; bu değer MQDXP_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu, çıkışa bir giriş alanıdır.

Version

Tip: MQLONG

Yapı sürümü numarası. Değer şöyle olmalıdır:

MQDXP_VERSION_1

Veri dönüştürme çıkış parametresi yapısına ilişkin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQDXP_CURRENT_VERSION

Veri dönüştürme çıkış parametresi yapısının yürürlükteki sürümü.

Not: Bu yapının yeni bir sürümü tanıtıldığında, var olan parçanın düzeni değişmez. Bu nedenle çıkış, *Version* alanının, çıkışın kullanması gereken alanları içeren en düşük sürüme eşit ya da bu sürümden büyük olup olmadığını kontrol etmelidir.

Bu, çıkışa bir giriş alanıdır.

C bildirimini

```
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 */
};
```

COBOL bildirimini (yalnızca 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.
```

System/390 çevirici bildirimini

```
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
MQDXP_AREA     DS CL(MQDXP_LENGTH)
```

MQXCNCV-Karakterlerin dönüştürülme

MQXCNCV çağrısı, C programlama dilini kullanarak karakterleri bir karakter kümesinden diğerine dönüştürür.

Bu çağrı, IBM MQ çerçeve arabirimlerinden biri olan IBM MQ Veri Dönüştürme Arabirimi 'nin (DCI) bir parçasıdır.

Not: Bu çağrı hem uygulama hem de veri dönüştürme çıkış ortamlarından kullanılabilir.

Sözdizimi

MQXCNCV (*Hconn, Options, SourceCCSID, SourceLength, SourceBuffer, TargetCCSID, TargetLength, TargetBuffer, DataLength, CompCode, Reason*)


Parametreler

Hconn.

Tip: MQHCONN-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir.

Bir veri dönüştürme çıkışında, Hconn genellikle MQDXP yapısının Hconn alanındaki veri dönüştürme çıkışına geçirilen tanıtıcı olur; bu tanıtıcı, MQGET çağrısını yayınlayan uygulama tarafından belirtilen tanıtıcı ile aynı olmayabilir.

 IBM üzerinde, Hconn için aşağıdaki özel değer belirtilebilir:

MQHC_DEF_HCONN

Varsayılan bağlantı tanıtıcısı.

CICS TS 3.2 ya da üstü bir uygulama çalıştırıyorsanız, MQXCNCV çağrısını çağıran karakter dönüştürme çıkış programının OPENAPI olarak tanımlandığından emin olun. Bu tanımlama, yanlış bir bağlantıdan kaynaklanan 2018 MQRC_HCONN_ERROR hatasını önler ve MQGET ' in tamamlanmasına izin verir.

Seçenekler

Tip: MQLONG-input

MQXCNCV işlemini denetleyen seçenekler.

Tanımlanan seçeneklerden sıfır ya da daha fazlası belirtilebilir. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliorsa).

Varsayılan dönüştürme seçeneği: Aşağıdaki seçenek, varsayılan karakter dönüşümünün kullanımını denetler:

MQDCC_DEFAULT_CONVERSION

Varsayılan dönüştürme.

Bu seçenek, çağrıda belirtilen karakter kümelerinden biri ya da her ikisi desteklenmiyorsa, varsayılan karakter dönüşümünün kullanılabilmesini belirtir. Bu, kuyruk yöneticisinin dizgiyi dönüştürürken, belirtilen karakter kümesine yaklaşan, kuruluş tarafından belirtilen bir varsayılan karakter kümesini kullanmasını sağlar.

Not: Dizgiyi dönüştürmek için yaklaşık bir karakter kümesi kullanılmasının sonucu, bazı karakterlerin yanlış dönüştürülebilmesinden kaynaklanır. Bu, yalnızca belirtilen karakter kümesi ve varsayılan karakter kümesi için ortak olan karakterler kullanılarak önlenir.

Varsayılan karakter kümeleri, kuyruk yöneticisi kurulduğunda ya da yeniden başlatıldığında bir yapılandırma seçeneği tarafından tanımlanır.

MQDCC_DEFAULT_CONVERSION belirtilmezse, kuyruk yöneticisi dizgiyi dönüştürmek için yalnızca belirtilen karakter kümelerini kullanır ve karakter kümelerinden biri ya da her ikisi desteklenmiyorsa çağrı başarısız olur.

Bu seçenek aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

Doldurma seçeneği: Aşağıdaki seçenek, dönüştürülen dizgiyi hedef arabelleğe sığdırmak için, kuyruk yöneticisinin dönüştürülen dizgiyi boşluklarla doldurmasını ya da anlamlı olmayan karakterleri atmasını sağlar:

MQDCC_FILL_TARGET_BUFFER

Hedef arabelleği doldur.

Bu seçenek, dönüştürmenin hedef arabelleğinin tam olarak doldurulabileceği şekilde gerçekleştirilmesini ister:

- Dizgi dönüştürüldüğünde kasılırsa, hedef arabelleği doldurmak için sondaki boşluklar eklenir.
- Dizgi dönüştürüldüğünde genişletilirse, dönüştürülen dizginin hedef arabelleğe sığması için sondaki anlamlı olmayan karakterler atılır. Bu işlem başarılı olursa, çağrı MQCC_OK ve neden kodu MQRC_NONE ile tamamlanır.

Çok az sayıda sonda karakter varsa, hedef arabelleğe sığabilecek kadar çok dizgi yerleştirilir ve çağrı MQCC_WARNING ve neden kodu MQRC_CONVERTED_MSG_TOO_BIG ile tamamlanır.

Önemsiz karakterler şunlardır:

- Sondaki boşluklar
- Dizgide ilk boş karakterin ardından gelen karakterler (ancak ilk boş karakterin kendisi hariç)
- TargetCCSID ve TargetLength dizgisi, hedef arabellek geçerli karakterlerle tam olarak ayarlanamazsa, çağrı MQCC_FAILED ve neden kodu MQRC_TARGET_LENGTH_ERROR ile başarısız olur. Bu durum, TargetCCSID saf bir DBCS karakter takımı (UTF-16 gibi) olduğunda ortaya çıkabilir, ancak TargetLength tek sayıda bayt olan bir uzunluğu belirtir.
- TargetLength , SourceLength değerinden küçük ya da daha büyük olabilir. MQXCNCV ' den dönüştürme DataLength , TargetLength ile aynı değere sahiptir.

Bu seçenek belirtilmezse:

- Dizginin, hedef arabellek içinde gerektiği gibi daralmasına ya da genişlemesine izin verilir. Önemsiz sondaki karakterler eklenmez ya da atılmaz.

Dönüştürülen dizgi hedef arabelleğe sığarsa, çağrı MQCC_OK ve neden kodu MQRC_NONE ile tamamlanır.

Dönüştürülen dizgi hedef arabellek için çok büyükse, hedef arabelleğe sığan dizginin çoğu hedef arabelleğe yerleştirilir ve çağrı MQCC_WARNING ve neden kodu MQRC_CONVERTED_MSG_TOO_BIG ile tamamlanır. Bu durumda TargetLength bayttan az döndürülebilir.

- TargetLength , SourceLength değerinden küçük ya da daha büyük olabilir. MQXCNCV ' den dönüştürme, DataLength TargetLength değerinden küçük ya da bu değere eşit.

Bu seçenek aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

Kodlama seçenekleri: Tanımlanan seçenekler, kaynak ve hedef dizgilerin tamsayı kodlamasını belirtmek için kullanılabilir. İlgili kodlama, yalnızca karşılık gelen karakter kümesi tanıtıcısı, ana bellekteki karakter kümesinin gösteriminin ikili tamsayılar için kullanılan kodlamaya bağlı olduğunu gösterdiğinde kullanılır. Bu yalnızca belirli çok baytlık karakter kümelerini etkiler (örneğin, UTF-16 karakter kümeleri).

Karakter takımı tek baytlık karakter takımı (SBCS) ya da tamsayı kodlamasına bağımlı olmayan ana saklama alanında gösterimi olan çok baytlık bir karakter takımıysa, kodlama dikkate alınmaz.

MQDCC_TARGET_ * değerlerinden biriyle birlikte MQDCC_SOURCE_ * değerlerinden yalnızca biri belirtilmelidir:

MQDCC_SOURCE_ENC_NATIVE

Kaynak kodlama, ortam ve programlama dili için varsayılan kodlamadır.

MQDCC_SOURCE_ENC_NORMAL

Kaynak kodlaması normal.

MQDCC_SOURCE_ENC_REVERSED

Kaynak kodlaması tersine çevrildi.

MQDCC_SOURCE_ENC_UNDEFINED

Kaynak kodlaması tanımlı değil.

MQDCC_TARGETC_NATIVE

Hedef kodlama, ortam ve programlama dili için varsayılan değerdir.

MQDCC_TARGET_ENC_NORMAL

Hedef kodlaması normal.

MQDCC_TARGET_ENC_REVERSED

Hedef kodlaması tersine çevrildi.

MQDCC_TARGET_ENC_UNDEFINED

Hedef kodlama tanımlı değil.

Önceden tanımlanan kodlama değerleri doğrudan Options alanına eklenebilir. Ancak, kaynak ya da hedef kodlama MQMD 'deki Encoding alanından ya da başka bir yapıdan elde edilirse, aşağıdaki işlemler yapılmalıdır:

1. Tamsayı kodlaması, kayar ve paketlenmiş ondalık kodlamaları ortadan kaldırılarak Encoding alanından çıkarılmalıdır; bunun nasıl yapılacağına ilişkin ayrıntılar için bkz. [“Kodlamalar çözümleniyor” sayfa 890](#) .
2. Adım 1 'den kaynaklanan tamsayı kodlaması, Options alanına eklenmeden önce uygun katsayıyla çarpılmalıdır. Bu etkenler şunlardır:
 - Kaynak kodlaması için MQDCC_SOURCE_ENC_FACTOR
 - Hedef kodlama için MQDCC_TARGET_ENC_FACTOR





Aşağıdaki örnek kod, bunun C programlama dilinde nasıl kodlanabileceğini gösterir:

```
Options = (MsgDesc.Encoding & MQENC_INTEGER_MASK)
          * MQDCC_SOURCE_ENC_FACTOR
          + (DataConvExitParms.Encoding & MQENC_INTEGER_MASK)
          * MQDCC_TARGET_ENC_FACTOR;
```

Belirtilmezse, kodlama seçenekleri varsayılan olarak tanımsız (MQDCC_ * _ENC_UNDEFINED) olarak ayarlanır. Çoğu durumda bu, MQXCNCV çağrısının başarıyla tamamlanmasını etkilemez. Ancak, karşılık gelen karakter kümesi, gösterime bağımlı çok byte 'lık bir karakter kümeiye (örneğin, bir UTF-16 karakter kümesi), çağrı MQRC_SOURCE_INTEGER_ENC_ERROR ya da MQRC_TARGET_INTEGER_ENC_ERROR neden koduyla başarısız olur.

Kodlama seçenekleri aşağıdaki ortamlarda desteklenir:

-  AIX

-  IBM i
-  Linux
-  Windows
-  z/OS

Varsayılan seçenek: Önceden açıklanan seçeneklerden hiçbiri belirtilmezse, aşağıdaki seçenek kullanılabilir:

MQDCC_NONE

Seçenek belirtilmedi.

MQDCC_NONE, program belgelerine yardımcı olmak için tanımlandı. Bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

SourceCCSID

Tip: MQLONG-input

Bu, `SourceBuffer` içindeki giriş dizgisinin kodlanmış karakter takımı tanıtıcısıdır.

SourceLength

Tip: MQLONG-input

Bu, `SourceBuffer` ; içindeki giriş dizgisinin bayt cinsinden uzunluğudur. Sıfır ya da daha büyük olmalıdır.

SourceBuffer

Tip: MQCHAR x `SourceLength` -giriş

Bu, bir karakter kümesinden diğerine dönüştürülecek dizgiyi içeren arabellektir.

TargetCCSID

Tip: MQLONG-input

Bu, `SourceBuffer` ' in dönüştürüleceği karakter takımının kodlanmış karakter takımı tanıtıcısıdır.

TargetLength

Tip: MQLONG-input

Bu, çıkış arabelleğinin bayt cinsinden uzunluğudur `TargetBuffer` ; Sıfır ya da daha büyük olmalıdır. `SourceLength` değerinden küçük ya da büyük olabilir.

TargetBuffer

Tip: MQCHAR x `TargetLength` -çıkış

Bu, `TargetCCSID` tarafından tanımlanan karakter kümesine dönüştürüldükten sonraki dizedir. Dönüştürülen dizgi, dönüştürülmemiş dizgiden daha kısa ya da daha uzun olabilir. **DataLength** parametresi, döndürülen geçerli bayt sayısını gösterir.

DataLength

Tip: MQLONG-output

Bu, `TargetBuffer` çıkış arabelleğinde döndürülen dizginin uzunluğudur. Dönüştürülen dizgi, dönüştürülmemiş dizgiden daha kısa ya da daha uzun olabilir.

CompCode

Tip: MQLONG-output

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici CompCode.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_CONVERTED_MSG_TOO_BIG

(2120, X'848 ') Dönüştürülen veriler arabellek için çok büyük.

CompCode MQCC_FAILED ise:

MQRC_DATA_LENGTH_HATA

(2010, X'7DA') Veri uzunluğu parametresi geçersiz.

MQRC_DBCS_HATA

(2150, X'866 ') DBCS dizgisi geçersiz.

MQRC_HCONN_HATA

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

MQRC_OPTIONS_HATA

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_RESOURCE_PROBLEM

(2102, X'836 ') Sistem kaynakları yetersiz.

MQRC_SOURCE_BUFFER_ERROR (QRC_SOURCE_BUFFER_ERROR)

(2145, X'861 ') Kaynak arabellek değiştirgesi geçersiz.

MQRC_SOURCE_CCSID_HATA

(2111, X'83F') Kaynak kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_SOURCE_INTEGER_ENC_HATA

(2112, X'840 ') Kaynak tamsayı kodlaması tanınmıyor.

MQRC_SOURCE_LENGTH_ERROR

(2143, X'85F') Kaynak uzunluğu parametresi geçersiz.

MQRC_STORAGE_KULLANILAMIYOR

(2071, X'817 ') Kullanılabilir bellek yetersiz.

MQRC_TARGET_BUFFER_HATA

(2146, X'862 ') Hedef arabellek değiştirgesi geçersiz.

MQRC_TARGET_CCSID_HATA

(2115, X'843 ') Hedef kodlanmış karakter takımı tanıtıcısı geçersiz.

MQRC_TARGET_INTEGER_ENC_HATA

(2116, X'844 ') Hedef tamsayı kodlaması tanınmıyor.

MQRC_TARGET_LENGTH_HATA

(2144, X'860 ') Hedef uzunluk parametresi geçersiz.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili ayrıntılı bilgi için bkz. [İletiler ve neden kodları](#).

C çağırısı

MQXCNCV (Hconn, Options, SourceCCSID, SourceLength, SourceBuffer,

```
TargetCCSID, TargetLength, TargetBuffer, &DataLength,  
&CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
```

COBOL bildiri (yalnızca IBM i)

IBM i

```
CALL 'MQXCNCV' USING HCONN, OPTIONS, SOURCECCSID, SOURCELENGTH,  
                    SOURCEBUFFER, TARGETCCSID, TARGETLENGTH,  
                    TARGETBUFFER, DATALENGTH, COMPCODE, REASON.
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
** 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.
```

S/390 çevirici bildiri

```
CALL MQXCNCV, (HCONN, OPTIONS, SOURCECCSID, SOURCELENGTH, X  
              SOURCEBUFFER, TARGETCCSID, TARGETLENGTH, TARGETBUFFER, X  
              DATALENGTH, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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  
* conversion  
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
```


*			conversion
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-Veri dönüştürme çıkışı

MQ_DATA_CONV_EXIT çağrısı, veri dönüştürme çıkışına geçirilen parametreleri açıklar.

Kuyruk yöneticisi tarafından MQ_DATA_CONV_EXIT adlı bir giriş noktası sağlanmadı (kullanım notu [11](#) ' e bakın).

Bu tanım, IBM MQ çerçeve arabirimlerinden biri olan IBM MQ Veri Dönüştürme Arabirimi 'nin (DCI) bir parçasıdır.

Sözdizimi

MQ_DATA_CONV_EXIT (*DataConvExitParms*, *MsgDesc*, *InBufferLength*, *InBuffer*, *OutBufferLength*, *OutBuffer*)

Parametreler

DataConvExitParms

Tip: MQDXP-giriş/çıkış

Bu yapı, çıkışın çağrılmasıyla ilgili bilgileri içerir. Çıkış, dönüştürmenin sonucunu belirtmek için bu yapıdaki bilgileri ayarlar. Bu yapıdaki alanların ayrıntıları için bkz. "[MQDXP-Veri dönüştürme çıkış parametresi](#)" sayfa 901 .


MsgDesc

Tip: MQMD-giriş/çıkış

Çıkışa girişte, bu, **InBuffer** değişirgesindeki çıkışa geçirilen ileti verileriyle ilişkilendirilmiş ileti tanımlayıcısıdır.

Not: Çıkışa geçirilen **MsgDesc** değişirgesi her zaman, çıkışı çağırın kuyruk yöneticisi tarafından desteklenen en son MQMD sürümüdür. Çıkışın farklı ortamlar arasında taşınabilir olması amaçlanıyorsa, çıkış, çıkışın erişmesi gereken alanların yapıda var olduğunu doğrulamak için MsgDesc içindeki *Version* alanını denetler.

Aşağıdaki ortamlarda, çıkışa bir version-2 MQMD geçirilir:

-  AIX
-  IBM i
-  Linux
-  Windows

Veri dönüştürme çıkışını destekleyen diğer tüm ortamlarda, çıkışa version-1 MQMD geçirilir.

Çıkışta, dönüştürme başarılı olursa, çıkış *Encoding* ve *CodedCharSetId* alanlarını uygulama tarafından istenen değerlerle değiştirir; bu değişiklikler uygulamaya geri yansıtılır. Çıkışın yapıda yaptığı diğer değişiklikler yoksayılar; bunlar uygulamaya geri yansıtılmaz.

Çıkış, MQDXP yapısının *ExitResponse* alanında MQXDR_OK değerini döndürür, ancak ileti tanımlayıcısındaki *Encoding* ya da *CodedCharSetId* alanlarını değiştirmezse, kuyruk yöneticisi bu alanlar için, MQDXP yapısındaki ilgili alanların çıkışa girişte sahip olduğu değerleri döndürür.

InBufferUzunluğu

Tip: MQLONG-input

InBufferbayt cinsinden uzunluk.

Bu, InBuffer giriş arabelleğinin uzunluğudur ve çıkış tarafından işlenecek bayt sayısını belirtir. InBufferLength , dönüşümden önceki ileti verilerinin uzunluğundan ve MQGET çağrısında uygulama tarafından sağlanan arabelleğin uzunluğundan daha azdır.

Değer her zaman sıfırdan büyüktür.

InBuffer

Tip: MQBYTEInBufferLength -giriş

Dönüştürülemeyen iletiyi içeren arabellek.

Bu, dönüştürmeden önce ileti verilerini içerir. Çıkış verileri dönüştüremezse, kuyruk yöneticisi çıkış tamamlandıktan sonra bu arabelleğin içeriğini uygulamaya döndürür.

Not: Çıkış InBuffer ; ögesini değiştirmemelidir. Bu parametre değiştirilirse, sonuçlar tanımsız olur.

C programlama dilinde, bu parametre bir işaretçi-to-void olarak tanımlanır.

OutBufferUzunluğu

Tip: MQLONG-input

OutBufferbayt cinsinden uzunluk.

Bu, OutBufferçıkış arabelleğinin uzunluğudur ve MQGET çağrısında uygulama tarafından sağlanan arabelleğin uzunluğuyla aynıdır.

Değer her zaman sıfırdan büyüktür.

OutBuffer

Tip: MQBYTEOutBufferLength -çıkış

Dönüştürülen iletiyi içeren arabellek.

Çıkışta, dönüştürme başarılı olduysa (**DataConvExitParms** değiştirgesinin ExitResponse alanındaki MQXDR_OK değeriyle belirtildiği gibi), OutBuffer istenen gösterimde uygulamaya teslim edilecek ileti verilerini içerir. Dönüştürme başarısız olduysa, çıkışın bu arabellekte yaptığı değişiklikler yoksayılır.

C programlama dilinde, bu parametre bir işaretçi-to-void olarak tanımlanır.

Kullanım notları

1. Veri dönüştürme çıkışı, bir MQGET çağrısının işlenmesi sırasında denetimi alan, kullanıcı tarafından yazılan bir çıkıştır. Veri dönüştürme çıkışı tarafından gerçekleştirilen işlev, çıkışın sağlayıcısı tarafından tanımlanır; ancak, çıkış burada açıklanan kurallara ve ilişkili değiştirge yapısına uygun olmalıdır MQDXP.

Veri dönüştürme çıkışı için kullanılacak programlama dilleri, ortam tarafından belirlenir.

2. Çıkış yalnızca aşağıdaki deyimlerin tümü doğruysa çağrılır:

- MQGET çağrısında MQGMO_CONVERT seçeneği belirtildi
- İleti tanımlayıcıdaki Format alanı MQFMT_NONE değil
- İleti gerekli gösterimde değil; yani, iletinin CodedCharSetId ve Encoding iletilerinden biri ya da her ikisi, MQGET çağrısında sağlanan ileti tanımlayıcısında uygulama tarafından belirtilen değerden farklı
- Kuyruk yöneticisi dönüştürme işlemini başarıyla gerçekleştirmedi
- Uygulamanın arabelleğinin uzunluğu sıfırdan büyük
- İleti verilerinin uzunluğu sıfırdan büyük
- MQGET işlemi sırasında şu ana kadar gelen neden kodu: MQRC_NONE ya da MQRC_TRUNCATED_MSG_ACCEPTED

3. Bir çıkış yazılırken, çıkışı, kesilen iletileri dönüştürmesini sağlayacak şekilde kodlamayı düşünün. Kesilmiş iletiler aşağıdaki şekillerde ortaya çıkabilir:

- Alan uygulama iletiden küçük bir arabellek sağlıyor, ancak MQGET çağrısında MQGMO_ACCEPT_TRUNCATED_MSG seçeneğini belirtiyor.

Bu durumda, çıkışa giriş için **DataConvExitParms** değiştirgesindeki Reason alanı MQRC_TRUNCATED_MSG_ACCEPTED değerini içeriyor.

- İletiyi gönderen kişi, iletiyi göndermeden önce kesmiş. Bu, örneğin rapor iletilerinde gerçekleşebilir (daha fazla ayrıntı için bkz. "Rapor iletilerinin dönüştürülmesi" sayfa 900).

Bu durumda, çıkışa girişteki **DataConvExitParms** değiştirgesindeki Reason alanı MQRC_NONE değerini içerir (alan uygulama ileti için yeterli büyüklükte bir arabellek sağladyasa).

Bu nedenle, çıkışa girişteki Reason alanının değeri, iletinin kesilip kesilmediğine karar vermek için her zaman kullanılamaz.

Kesilmiş bir iletinin ayırt edici özelliği, **InBufferLength** değiştirgesindeki çıkışa sağlanan uzunluğun, ileti tanımlayıcısındaki **Format** alanında bulunan biçim adının belirttiği uzunluktan az olması. Bu nedenle çıkış, herhangi bir veriyi dönüştürmeye çalışmadan önce **InBufferLength** değerini kontrol etmelidir; çıkış, biçim adı tarafından belirtilen tam veri miktarının sağlandığını varsaymamalıdır.

Çıkış, kesilen iletileri dönüştürmek için yazılmamışsa ve **InBufferLength** beklenen değerden küçükse, **DataConvExitParms** değiştirgesinin **ExitResponse** alanında **CompCode** ve **Reason** alanları MQCC_WARNING ve MQRC_FORMAT_ERROR olarak ayarlanmış olarak MQXDR_CONVERSION_FAILED değerini döndürür.

Çıkış, kesilmiş iletileri dönüştürmek için yazıldıysa, **InBuffer**' in sonundan sonraki verileri incelemeye ya da dönüştürmeye çalışmamaya dikkat ederek, verilerin büyük bir bölümü dönüştürülür (sonraki kullanım notuna bakın). Dönüştürme başarıyla tamamlanırsa, çıkış **DataConvExitParms** değiştirgesindeki Reason alanını değiştirmeden bırakır. İleti alıcının kuyruk yöneticisi tarafından kesildiyse MQRC_TRUNCATED_MSG_ACCEPTED, ileti gönderen tarafından kesildiyse MQRC_NONE döndürülür.

Ayrıca, bir iletinin dönüştürme sırasında **OutBuffer** değerinden büyük olduğu noktaya genişletilmesi de mümkündür. Bu durumda, çıkışın iletinin kesilip kesilmeyeceğine karar vermesi gerekir; **DataConvExitParms** parametresindeki **AppOptions** alanı, alan uygulamanın MQGMO_ACCEPT_TRUNCATED_MSG seçeneğini belirtip belirtmediğini belirtir.

4. Genellikle, **InBuffer** içinde çıkışa sağlanan iletideki tüm veriler dönüştürülür ya da hiçbiri dönüştürülmez. Ancak, ileti dönüştürmeden önce ya da dönüştürme sırasında kesilirse, bunun için bir kural dışı durum oluşur; bu durumda arabelleğin sonunda eksik bir öge olabilir (örneğin: çift baytlık bir karakterin 1 baytı ya da 4 baytlık bir tamsayının 3 baytı). Bu durumda, eksik ögeyi atlayın ve **OutBuffer** içindeki kullanılmayan byte 'ları boş değere ayarlayın. Ancak, bir dizi ya da dizgi içindeki tüm öğelerin ya da karakterlerin dönüştürülmesi gerekir.
5. İlk kez bir çıkış gerektiğinde, kuyruk yöneticisi, biçimle (uzantılar dışında) aynı ada sahip bir nesneyi yüklemeye çalışır. Yüklenen nesne, bu biçim adına sahip iletileri işleyen çıkışı içermelidir. Tüm ortamlar bunu gerektirmese de, çıkış adını ve çıkışı içeren nesnenin adını aynı yapmayı göz önünde bulundurun.
6. Uygulama, kuyruk yöneticisine bağlandığından bu yana **Format** ' u kullanan ilk iletiyi almaya çalıştığında çıkışın yeni bir kopyası yüklenir. CICS ya da IMS uygulamaları için bu, CICS ya da IMS altsisteminin kuyruk yöneticisine bağlanması anlamına gelir. Kuyruk yöneticisi önceden yüklenmiş bir kopyayı attıysa, başka zamanlarda da yeni bir kopya yüklenebilir. Bu nedenle, bir çıkış, çıkışın bir çağrısından diğerine bilgi iletmek için statik saklama alanını kullanmayı denememelidir; çıkış, iki çağırma arasında boşaltılabilir.
7. Kuyruk yöneticisi tarafından desteklenen yerleşik biçimlerden biriyle aynı ada sahip kullanıcı tarafından sağlanan bir çıkış varsa, kullanıcı tarafından sağlanan çıkış yerleşik dönüştürme yordamının yerini almaz. Böyle bir çıkışın çağrıldığı durumlar şunlardır:
 - Yerleşik dönüştürme yordamı, ilgili **CodedCharSetId** ya da **Encoding** ürününe ya da bu üründen gelen dönüştürmeleri işleyemezse, ya da
 - Yerleşik dönüştürme yordamı verileri dönüştüremediyse (örneğin, dönüştürülemeyen bir alan ya da karakter olduğu için).


8. Çıkışın kapsamı ortama bağlıdır. Diğer biçimlerle çakışma riskini en aza indirmek için Format adları seçilmelidir. Biçim adını tanımlayan uygulamayı tanımlayan karakterlerle başlamayı düşünün.
9. Veri dönüştürme çıkışı, MQGET çağrısını yayınlayan programın ortamına benzer bir ortamda çalışır; ortam adres alanını ve kullanıcı tanımını içerir (varsa). Program, ileti dönüştürmeyi desteklemeyen bir hedef kuyruk yöneticisine ileti gönderen bir ileti kanalı aracısı olabilir. Çıkış, kuyruk yöneticisinin ortamında çalışmadığı için kuyruk yöneticisinin bütünlüğünü bozamaz.
10. Çıkış tarafından kullanılacak tek MQI çağrısı MQXCNCV; diğer MQI çağrılarını kullanma girişimi başarısız oldu; neden kodu MQRC_CALL_IN_PROGRESS ya da diğer öngörülemez hatalar.
11. Kuyruk yöneticisi tarafından MQ_DATA_CONV_EXIT adlı bir giriş noktası sağlanmadı. Ancak, C programlama dilinde MQ_DATA_CONV_EXIT adı için bir typedef sağlanır ve bu, parametrelerin doğru olduğundan emin olmak için kullanıcı tarafından yazılan çıkışı bildirmek için kullanılabilir. Çıkışın adı, tüm ortamlarda gerekli olmasa da, biçim adıyla (MQMD 'deki Format alanında bulunan ad) aynı olmalıdır.

Aşağıdaki örnekte, MYFORMAT biçimini işleyen çıkışın C programlama dilinde nasıl bildirildiği gösterilmektedir:

```
#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.  z/OS üzerinde, bir API geçiş çıkışı da yürürlükte ise, veri dönüştürme çıkışından sonra çağrılır.

C çağrısı

```
exitname (&DataConvExitParms, &MsgDesc, InBufferLength,
          InBuffer, OutBufferLength, OutBuffer);
```

Çıkışa geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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 */
```

COBOL bildirimini (yalnızca IBM i)



```
CALL 'exitname' USING DATACONVEXITPARMS, MSGDESC, INBUFFERLENGTH,
                      INBUFFER, OUTBUFFERLENGTH, OUTBUFFER.
```

Çıkışa geçirilen değıştirmeler ařađıdaki gibi bildirilir:

```
** 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).
```

System/390 çevirici bildirimi

```
CALL EXITNAME, (DATACONVEXITPARMS,MSGDESC,INBUFFERLENGTH,      X
                INBUFFER,OUTBUFFERLENGTH,OUTBUFFER)
```

Çıkışa geçirilen değıştirmeler ařađıdaki gibi bildirilir:

```
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
```

MQRFH2 öğeleri olarak belirtilen özellikler

İleti dışı tanımlayıcı özellikleri, MQRFH2 üstbilgi klasörlerinde öğe olarak belirtilebilir. Özellikler olarak belirtilmekte olan MQRFH2 öğelerine genel bakış.

Bu, IBM MQ JMS ve XMS istemcilerinin önceki sürümleriyle uyumluluđu sürdürür. Bu bölümde, MQRFH2 üstbilgilerinde özelliklerin nasıl belirtileceđini açıklanmaktadır.

MQRFH2 öğelerini özellik olarak kullanmak için, öğeleri *IBM MQ classes for Java Kullanılması* konusunda açıklandığı gibi belirtin. Bu bilgiler, [“MQRFH2 -Kurallar ve biçimlendirme üstbilgisi 2” sayfa 523](#) içinde açıklanan bilgileri tamamlar.

Özellik veri tiplerini MQRFH2 veri tipleriyle eşleme

Bu konu, ilgili MQRFH2 veri tipleriyle eşlenen ileti özelliđi tipleriyle ilgili bilgi sağlar.

İleti özelliđi tipi	MQRFH2 veri tipi
MQBYTE []	bin.hex
MQBOOL	boole
MQINT8	i1
MQINT16	i2
MQINT32	i4
MQINT64	i8
MQFLOAT32	r4

Çizelge 635. Desteklenen MQRFH2 veri tipleri (devamı var)	
İleti özelliği tipi	MQRFH2 veri tipi
MQFLOAT64	r8
MQCHAR []	dizgi

Veri tipi olmayan herhangi bir ögenin "string" tipinde olduğu varsayılır.

Belirtilmemiş bir tamsayı olan MQRFH2 veri tipi `int`, `i8` olarak kabul edilir.

`xsi:nil='true'` öge özniteliğiyle boş değer gösterilir. Boş olmayan değerler için `xsi:nil='false'` özniteliğini kullanmayın.

Örneğin, aşağıdaki özelliğin boş değeri vardır:

```
<NullProperty xsi:nil='true'></NullProperty>
```

Bir byte ya da karakter dizgisi özelliğinin değeri boş olabilir. Bu, sıfır uzunluklu öge değerine sahip bir MQRFH2 ögesiyle gösterilir.

Örneğin, aşağıdaki özelliğin boş bir değeri vardır:

```
<EmptyProperty></EmptyProperty>
```

Desteklenen MQRFH2 klasörleri

İleti tanımlayıcı alanlarının özellik olarak kullanılmasına genel bakış.

`<jms>`, `<mcd>`, `<mqext>` ve `<usr>` klasörleri MQRFH2 üstbilgisi ve JMS içinde açıklanmaktadır. `<usr>` klasörü, bir iletiyle ilişkilendirilmiş JMS uygulaması tanımlı özellikleri iletmek için kullanılır. `<usr>` klasöründe gruplara izin verilmez.

MQRFH2 üstbilgisi ve JMS aşağıdaki ek klasörleri destekler:

- `<mq>`

Bu klasör, IBM MQ tarafından kullanılan MQ tanımlı özellikler için kullanılır ve ayrılır.

- `<mq_usr>`

Özellikler bir JMS özelliğinin gereksinimlerini karşılamayabileceğinden, JMS kullanıcı tanımlı özellikleri olarak gösterilmeyen uygulama tanımlı özellikleri iletmek için bu klasör kullanılabilir. Bu klasör, `<usr>` klasörünün içermediği grupları içerebilir.

- `content='properties'` özniteliğiyle işaretlenmiş herhangi bir klasör.

Böyle bir klasör, içerikteki `<mq_usr>` klasörünün eşdeğeridir.

- `<mqps>`

Bu klasör, IBM MQ yayınlama/abone olma özellikleri için kullanılır.

IBM MQ, WAS/SIB tarafından kullanılmakta olan şu klasörleri de destekler:

- `<sib>`

Bu klasör, JMS özellikleri olarak gösterilmeyen ya da `JMS_IBM_*` özellikleriyle eşlenen, ancak WAS/SIB uygulamalarına gösterilen WAS/SIB sistem iletişi özellikleri için kullanılır ve ayrılır; bunlar ileriye dönük ve ters yöneltme yolu özelliklerini içerir.

Bayt dizileri oldukları için en azından bazıları JMS özellikleri olarak gösterilemez. Uygulamanız bu klasöre özellikler eklerse, değer yoksayılır ya da kaldırılır.

- `<sib_usr>`

Bu klasör, desteklenen tipte olmadıkları için JMS kullanıcı özellikleri olarak gösterilemeyen WAS/SIB kullanıcı iletileri özellikleri için kullanılır ve ayrılır; bunlar WAS/SIB uygulamalarına gösterilir.

Bunlar, SIMessage arabiriminden alabileceğiniz ya da ayarlayabileceğiniz kullanıcı özellikleridir, ancak bayt dizisinin içeriği gerekli özellik değeriyle eşlenir.

IBM MQ uygulamanız klasöre isteğe bağlı bir bin.hex ögesi yazarsa, geri yükleme beklendiği biçimde olmadığı için uygulama büyük olasılıkla bir IOExceptionalır. bin.hex ögesinden başka bir öge eklerseniz, bir ClassCastExceptionalrsınız.

Bu klasörü kullanarak özellikleri WAS/SIB için kullanılabilir kılmayı denemeyin; bunun yerine <usr> klasörünü bu amaçla kullanın.

- <sib_context>

Bu klasör, WAS/SIB kullanıcı uygulamalarına ya da JMS özellikleri olarak gösterilmeyen WAS/SIB sistem iletileri özellikleri için kullanılır. Bunlar, web hizmetleri ve benzeri için kullanılan güvenlik ve işlemsel özellikleri içerir.

Uygulamanız bu klasöre özellik eklememelidir.

- <mqema>

Bu klasör <mqext> klasörü yerine WAS/SIB tarafından kullanıldı.

MQRFH2 klasör adları büyük ve küçük harfe duyarlıdır.

Aşağıdaki klasörler, küçük harfli ya da büyük harfli karakterlerin herhangi bir karışımında ayrılır:

- Öneki mq ya da wmq olan herhangi bir klasör; IBM MQ tarafından kullanılmak üzere ayrılmıştır.
- Öneki sib olan herhangi bir klasör; WAS/SIB tarafından kullanılmak üzere ayrılmıştır.
- <Root> ve <Body> klasörleri; ayrılmış, ancak kullanılmamış.

Şu klasörler, ileti özelliklerini içeren olarak tanınmıyor:

- <psc>

IBM Integration Bus tarafından aracıya yayınlama/abone olma komut iletilerini iletmek için kullanılır.

- <pscr>

IBM Integration Bus tarafından, yayınlama/abone olma komut iletilerine yanıt olarak aracıdaki bilgileri içermek için kullanılır.

- IBM tarafından tanımlanmamış, content='properties' özniteliğiyle işaretlenmemiş herhangi bir klasör.

<psc> ya da <pscr> klasörlerinde content='properties' belirtmeyin. Bunu yaparsanız, bu klasörler özellik olarak kabul edilir ve IBM Integration Bus beklendiği gibi çalışmasını durdurabilir.

Uygulamanız özellikler içeren iletiler oluşturuyorsa, MQRFH2 üstbilgilerinde özellikleri içeren bir MQRFH2 üstbilgisi olarak tanınması için, üstbilgi iletinin başında zincirlenebilecek üstbilgiler listesinde olmalıdır.

MQRFH2 ' den önce herhangi bir sayıda MQH standart üstbilgisi ya da bir MQCIH, bir MQDLH, bir MQIIH, bir MQTM, bir MQTMC2ya da bir MQXQH olabilir. Bir dizgi ya da MQCFH, zincirlenemediği için ayrıştırması sonlandırır.

Bir iletinin birden çok MQRFH2 üstbilgisi içermesi mümkündür. Aynı ada sahip klasörler, örneğin WAS/SIB tarafından kısıtlanmadıkça farklı üstbilgilerde birlikte var olabilir. Klasörlerin tümü önemli üstbilgilerdeyse, klasörler tek bir mantıksal klasör olarak işlenir.

Önemli üstbilgilerdeki klasörler önemli olmayan üstbilgilerde bu klasörlerle birleştirilemezken, anlamlı üstbilgilerde aynı adı taşıyan klasörler birleştirilebilir ve çakışan özellikler kaldırılabilir. Uygulamalarınız, iletilerindeki özelliklerin yerleşim düzenine bağlı olmamalıdır.

MQRFH2 grupları, <wmq>, <jms>, <mcd>, <usr>, <mqext>, <sib>, <sib_usr>, <sib_context>ve <mqema> klasörlerindeki özellikler için ayrıştırılır.

IBMtanımlı özellikler klasörlerindeki gruplar, <wmq> ve <mq> klasörleri dışında, özellikler için ayrıştırılır.

MQRFH2 klasörü karma içerik içeremez; bir klasör ya da grup gruplar ya da özellikler ya da bir değer içerebilir, ancak her ikisini birden içeremez.

İletinin ilk ya da sonraki bir bölümü, ileti tanımlayıcısında bu özelliklerden başka IBM MQ tanımlı özellikleri içeremez. Bu nedenle, bu tür özellikleri içeren bir iletinin MQMF_SEGMENT ya da MQMF_SEGMENTATION_ALLOWED kümesi ile yerleştirilmesi, koyma işleminin MQRC_SEGMENTATION_NOT_ALLOWED ile başarısız olmasına neden olur.


Ancak, ileti grupları IBM MQ tanımlı özellikleri içerebilir.

MQRFH2 üstbilgilerinin oluşturulması

IBM MQ , ileti özelliklerini MQRFH2 gösterimine çevirirse, iletiye MQRFH2 eklemelidir. MQRFH2 ögesini ayrı bir üstbilgi olarak ekler ya da var olan bir üstbilgiyle birleştirir.

IBM MQ tarafından yeni MQRFH2 üstbilgilerinin oluşturulması, bir iletide var olan üstbilgileri kesintiye uğratabilir. Üstbilgiler için ileti arabelleğini ayrıştıran uygulamalar, arabellekteki üstbilgilerin sayısının ve konumunun bazı durumlarda değişebileceğini bilmelidir. IBM MQ , ileti özelliklerini var olan bir MQRFH2 üstbilgisiyle birleştirerek bir iletiye özellik ekleme etkisini en aza indirmeye çalışır. Ayrıca, oluşturulan bir MQRFH2 ögesini ileti arabelleğindeki diğer üstbilgilerle görel olarak sabit bir konuma ekleyerek etkiyi en aza indirmeyi dener.

MQMDve istediğiniz sayıda MQXQH, MQRFHve MQDLH üstbilgilerinin ardından, hangi sırada olursa olsun, oluşturulan bir MQRFH2 üstbilgisi yerleştirilir. Oluşturulan MQRFH2 üstbilgisi, MQMD, MQXQH, MQDLHya da MQRFH üstbilgisi olmayan ilk üstbilginin hemen önüne yerleştirilir.

 z/OS sistemlerinde, oluşturulan MQRFH2 üstbilgisi uygulamanın CCSIDiçinde oluşturulur. Bu, aşağıdaki gibi tanımlanır:

- DLL arabirimini kullanan toplu LE uygulamaları için CCSID , **MQCONN** komutu verildiğinde (varsayılan değer 1047 'dir) yürürlükteki ülke değeriyle ilişkilendirilmiş CODESET değeridir.
- Toplu MQ sınırlı kod öbeklerinden biriyle bağ tanımlanan toplu LE uygulamaları için CCSID , **MQCONN** ' den sonra yayınlanan ilk MQI çağrısı sırasında (varsayılan değer 1047 'dir) yürürlükteki ülke değeriyle ilişkilendirilmiş CODESET değeridir.
- z/OS UNIX System Services (z/OS UNIX) iş parçacığında çalışan LE dışı toplu uygulamalar için CCSID , **MQCONN** sonrasında yayınlanan ilk MQI çağrısı sırasında THLICCSID değeridir (varsayılan değer 1047 'dir).
- Diğer toplu iş uygulamaları için CCSID , kuyruk yöneticisinin CCSID ' udur.

LE uygulamaları için, ülke değeri setlocale() / CEESETL LE çağrılabilir hizmeti kullanılarak değiştirilebilir. z/OS UNIX iş parçacıklarında çalışan LE dışı uygulamalar için, THLICCSID değeri z/OS UNIX eşleme makrosu **BPXYTHLI**kullanılarak değiştirilebilir.

Oluşturulan MQRFH2 ' yi birleştirme kuralları

Aşağıdaki kurallar, oluşturulan bir MQRFH2 ögesinin var olan bir MQRFH2ile birleştirilmesi için geçerlidir. Oluşturulan MQRFH2 üstbilgisi, varsa, var olan bir MQRFH2 üstbilgisiyle birleştirilir:

1. Var olan MQRFH2 aynı konumda IBM MQ , oluşturulan bir MQRFH2ya da daha önceki bir konumu üstbilgi zincirine yerleştirir.
2. Oluşturulan özelliklerin CCSID 'si, var olan MQRFH2içindeki NameValueCCSID ile aynıdır.

Tersi durumda, oluşturulan üstbilgi, daha önce açıklanan konumda arabelleğe ayrı olarak yerleştirilir.

Var olan bir MQRFH2 içinde klasörleri birleştirme kuralları

İleti özellikleri var olan bir MQRFH2ile birleştirilirse, ileti özellikleriyle eşleşen klasörler için var olan MQRFH2 taranır ve bunları birleştirir. Eşleşen bir klasör yoksa, var olan klasörlerin sonuna yeni bir klasör eklenir. Eşleşen bir klasör varsa, klasör aranır. Eşleşen özelliklerin üzerine yazılır. Yenileri klasörün sonuna eklenir.

MQRFH2 klasör kısıtlamaları

MQRFH2 üstbilgilerinde klasör kısıtlamalarına genel bakış

MQRFH2 kısıtlamaları aşağıdaki klasörler için geçerlidir:

- <us1> klasöründeki öge adları JMS önekiyle başlamamalıdır; Bu özellik adları JMS tarafından kullanılmak üzere ayrılmıştır ve kullanıcı tanımlı özellikler için geçerli değildir.

Böyle bir öge adı, MQRFH2 'nin ayrıştırılmasının başarısız olmasına neden olmaz, ancak IBM MQ ileti özelliği API' leri tarafından erişilemez.

- <us1> klasöründeki öge adları, küçük ya da büyük harf, NULL, TRUE, FALSE, NOT, AND, OR, BETWEEN, LIKE, IN, IS ve ESCAPE karışık olmamalıdır. <us1> , seçicide belirli bir özellik için klasör belirtilmediğinde kullanılan varsayılan klasör olduğundan, bu adlar SQL anahtar sözcükleriyle eşleşir ve ayrıştırma seçimlerini zorlaştırır.

Böyle bir öge adı, MQRFH2 'nin ayrıştırılmasının başarısız olmasına neden olmaz, ancak IBM MQ ileti özelliği API' leri tarafından erişilemez.

- <us1> klasörünün içerik modeli aşağıdaki gibidir:

- Geçerli bir XML adı, iki nokta üst üste işareti içermemesi koşuluyla, öge adı olarak kullanılabilir.
- Yalnızca, içiçe yerleşimli klasörler olmayan yalın öğelere izin verilir.
- Bir dt="xxx" özniteliği tarafından değiştirilmedikçe, tüm öğeler varsayılan dizgi tipini alır.
- Tüm öğeler isteğe bağlıdır, ancak bir klasörde bir kereden fazla geçmemelidir.

- İleti özelliklerini içeren herhangi bir klasördeki öge adları nokta (.) içermemelidir (Unicode karakteri U+002E); bunun nedeni, sıradüzenini belirtmek için özellik adlarında kullanılmasıdır.

Böyle bir öge adı, MQRFH2 'nin ayrıştırılmasının başarısız olmasına neden olmaz, ancak IBM MQ ileti özelliği API' leri tarafından erişilemez.

Genel olarak, MQRFH2 'nin bazı öğelerine IBM MQ ileti özelliği API' leri aracılığıyla erişilememesine rağmen, geçerli XML stili veriler içeren MQRFH2 üstbilgileri IBM MQ tarafından hatasız olarak ayrıştırılabilir.

MQRFH2 öge adı çakışmaları

MQRFH2 öge adları içindeki çakışmalara genel bakış.

Bir ileti özelliğine yalnızca bir değer eklenebilir. Bir özelliğe erişme girişimi bir değer çakışmasına yol açarsa, bir özellik diğerinden daha çok seçilir.

Bir klasör aynı adı taşıyan bir öge içermezse, MQRFH2 öğelerine erişmek için IBM MQ sözdizimi bir ögenin benzersiz olarak tanımlanmasına izin verir. Bir klasör aynı ada sahip birden fazla öge içeriyorsa, kullanılan özelliğin değeri, iletinin başına en yakın olandır.

Aynı ileti içinde aynı adı taşıyan iki ya da daha fazla klasör farklı anlamlı MQRFH2 üstbilgisinde bulunuyorsa bu geçerlidir.

İleti dışı bir tanımlayıcı özelliği iki kez ayarlandıktan sonra MQGET çağrısı işlendiğinde bir çakışma oluşabilir: hem bir MQSETMP çağrısı aracılığıyla hem de doğrudan işlenmemiş MQRFH2 üstbilgisinde.

Bu durumda, bir API çağrısı tarafından iletiyle ilişkilendirilen özellik, ileti verilerindeki (yani, işlenmemiş MQRFH2 üstbilgisindeki) bir özelliğe göre tercih edilir. Bir çakışma oluşursa, ileti verilerinden mantıksal olarak önce geldiği varsayılır.

Özellik adlarından MQRFH2 klasör ve öge adlarına eşleme

MQRFH2 üstbilgisinde özellik adları ve öge adları arasındaki farklılıklara genel bakış.

MQRFH2 üstbilgilerini oluşturan tanımlı API' lerden herhangi birini kullanırken, ileti özelliklerini belirtmek için (örneğin, MQ JMS), özellik adının MQRFH2 klasöründeki öge adı olması gerekmez.

Bu nedenle, hem ögeyi içeren klasör adı, hem de öge adı dikkate alınarak, özellik adından MQRFH2 ögesine bir eşleme gerçekleştirilir. IBM MQ classes for JMS ' in bazı örnekleri [IBM MQ classes for Java ' yi kullanma](#) içinde belgelenmiştir.

Çizelge 636. MQRFH2 klasör ve öge adlarıyla eşlenen özellik adları		
Özellik adı	MQRFH2 klasör adı	MQRFH2 öge adı
JMSDestination	jms	Dst
JMSType	mcd	Type, Set, Fmt
xxx (kullanıcı tanımlı, burada xxx JMSile başlar)	usr	xxx

Bu nedenle, bir JMS uygulaması JMSDestination özelliğine eriştiğinde bu, <jms> klasöründeki Dst ögesiyle eşlenir.

Özellikleri MQRFH2 öğeleri olarak belirtirken IBM MQ , öğelerini aşağıdaki gibi tanımlar:

Çizelge 637. MQRFH2 klasör, grup ve öge adlarıyla eşlenen özellik adları			
Özellik adı	MQRFH2 klasör adı	MQRFH2 grup adı	MQRFH2 öge adı
<Property>	<usr>	geçerli değil	<Property>
<folder>. <Property>	<folder>	geçerli değil	<Property>
<folder>. <group>. <Property>	<folder>	<group>	<Property>

Örneğin, bir IBM MQ uygulaması Property1 özelliğine erişmeye çalıştığında bu, <usr> klasöründeki Property1 ögesiyle eşlenir. wmq . Property2 özelliği, <wmq> klasöründeki Property2 özelliğiyle eşlenir.

Özellik adı birden çok içeriyorsa. karakter, kullanılan MQRFH2 öge adı, sondan sonra gelen addır. karakter ve MQRFH2 grupları bir sıradüzen oluşturmak için kullanılır; iç içe geçmiş MQRFH2 gruplarına izin verilir.

<mcd>, <jms>ve <mqext> klasörlerindeki MQRFH2 içinde bulunan JMS üstbilgisine ve sağlayıcıya özgü özelliklere, bir IBM MQ uygulaması tarafından [Kullanım IBM MQ classes for Java](#) içinde tanımlanan kısa adlar kullanılarak erişilir.

JMS kullanıcı tanımlı özelliklerine <usr> klasöründen erişilir. Bir IBM MQ uygulaması, özelliğin JMS uygulamaları tarafından kullanıcı tanımlı özelliklerinden biri olarak görüntülenmesi kabul ediliyorsa, uygulama özellikleri için <usr> klasörünü kullanabilir.

Kabul edilebilir değilse, başka bir klasör seçin; <wmq_usr> klasörü, bu türJMS dışı özellikler için standart bir konum olarak sağlanır.

Uygulamalarınız, aşağıdakileri göz önünde bulundurursanız, “MQRFH2 öğeleri olarak belirtilen özellikler” sayfa 917 içinde belgelenmemiş, iyi tanımlanmış bir kullanımla herhangi bir MQRFH2 klasörünü belirtebilir ve kullanabilir:

1. Klasör zaten kullanılıyor olabilir ya da ileride, içinde bulunan özelliklere tanımsız erişim sağlayan başka bir uygulama tarafından kullanılıyor olabilir; özellik adlarına ilişkin önerilen adlandırma kuralı için [Özellik adları](#) konusuna bakın.
2. Yalnızca kullanıcı tanımlı özellikler için <usr> klasörüne erişebilen IBM MQ classes for JMS ya da XMS istemcisinin önceki sürümleri bu özelliklere erişemez.
3. Klasör, değeri propertiesolarak ayarlanmış content özniteliğiyle işaretlenmelidir; örneğin, content=' properties '.

“MQSETMP-İleti özelliğini ayarla” sayfa 771 , bu özniteliği gerektiği şekilde otomatik olarak ekler. Bu öznitelik, IBMtanımlı klasörlerin hiçbirine eklenmemelidir; örneğin, <jms> ve <usr>. Bu, iletinin IBM WebSphere MQ 7.0öncesinde IBM MQ classes for JMS istemcisi tarafından reddedilmesine neden olur. bir MessageFormatExceptionile.

<usr> klasörü, <Property> sözdiziminin özellikleri için varsayılan konum olduğundan, aynı adı kullanarak aynı kullanıcı tanımlı özellik değerine erişmek için bir IBM MQ uygulaması ve bir JMS uygulaması.

Ayrılmış klasör adları

Birden çok ayrılmış klasör adı vardır. Klasör örnekleriniz gibi adlar kullanamazsınız; örneğin, Root ayrılmış olduğundan Root.Property1 geçerli bir özelliğe erişmez. Aşağıdaki listede ayrılmış klasör adları bulunmaktadır:

- Kök
- Gövde
- Özellikler
- Çevre
- LocalEnvironment
- DestinationList
- ExceptionList
- InputBody
- InputRoot
- InputProperties
- InputLocalOrtamı
- InputDestination(Giriş Hedefi) Listesi
- InputExceptionListesi
- OutputRoot
- OutputLocalOrtamı
- OutputDestination(Çıkış Hedefi) Listesi
- OutputExceptionListesi

Özellik tanımlayıcı alanlarını MQRFH2 üstbilgilerine eşleme

Bir özellik MQRFH2 ögesine çevrildiğinde, özellik tanımlayıcısının önemli alanlarını belirtmek için aşağıdaki öge öznitelikleri kullanılır: MQPD alanlarının MQRFH2 öge özniteliklerine nasıl çevrildiğini açıklar.

Destek

Destek özelliği tanımlayıcı alanı üç öge özniteliğe bölünür

- **sr** öge özniteliği, MQPD_REJECT_UNSUP_MASK bit maskesindeki değerleri belirtir.
- **sa** öge özniteliği, MQPD_ACCEPT_UNSUP_MASK bit maskesindeki değerleri belirtir.
- **sx** öge özniteliği, MQPD_ACCEPT_UNSUP_IF_XMIT_MASK bit maskesindeki değerleri belirtir.

Bu öge öznitelikleri yalnızca < mq > klasöründe geçerlidir ve özellikleri içeren diğer klasörlerdeki öğelerde ayarlandıysa yoksayılır.

Çizelge 638. MQRFH2 öge öznitelikleriyle eşlenen MQPD alanları		
Destek değeri	MQRFH2 öge özniteliği	MQRFH2 öznitelik değeri
MQPD_SUPPORT_OPTIONAL	Ct	isteğe bağlı Bu varsayılan değerdir.
MQPD_SUPPORT_REQUIRED	sr	gerekli

Çizelge 638. MQRFH2 öge öznitelikleriyle eşlenen MQPD alanları (devamı var)		
Destek değeri	MQRFH2 öge özniteliği	MQRFH2 öznitelik değeri
MQPD_SUPPORT_REQUIRED_IF_LOCAL	sx	yerel

Bağlam

Bir özelliğin ait olduğu ileti bağlamını belirtmek için **context** öge özniteliğini kullanın. Yalnızca bir değer kullanın. Bu öge özniteliği, özellikleri içeren herhangi bir klasördeki bir özellik için geçerlidir.

Çizelge 639. MQRFH2 öznitelik değerleriyle eşlenen bağlam değerleri	
Bağlam değeri	MQRFH2 öznitelik değeri
MQPD_NO_CONTEXT	yok Bu varsayılan değerdir.
MQPD_KULLANICI_BAĞLAMı	kullanıcı

CopyOptions

Bir özelliğin kopyalanması gereken iletileri belirtmek için **copy** öge özniteliğini kullanın. Birden çok değer kabul edilebilir; birden çok değeri virgülle ayırın. Örneğin, **copy='reply'** ve **copy='publish,report'** geçerlidir. Bu öge özniteliği, özellikleri içeren herhangi bir klasördeki bir özellik için geçerlidir.

Not: Öznitelik tanımlamasında, tek tırnak işaretleri ya da çift tırnak işaretleri geçerli kullanım biçimidir; örneğin, **copy='reply'** ya da **copy="report"**

Çizelge 640. CopyOption değerleri MQRFH2 öznitelik değerleriyle eşlendi	
CopyOption değeri	MQRFH2 öznitelik değeri
MQPD_COPY_FORWARD	ileri
MQPD_COPY_REPLY	Yanıt
MQPD_COPY_REPORT	rapor
MQPD_COPY_PUBLISH	yayınlamak
MQPD_KOPYASI_Tümü	tümü Bunu başka bir değerle belirtmeyin. Başka bir değerle kullanıldığında bu, none dışında herhangi bir değerden önceliklidir.
MQPD_COPY_DEFAULT	varsayılan Bu varsayılan değerdir. MQCOPY_FORWARD, MQCOPY_REPORT ve MQCOPY_PUBLISH olmak üzere üç değer belirtmesine eşdeğerdir. Bunu başka bir değerle belirtmeyin.
MQPD_KOPYASI_YOK	yok Bunu başka bir değerle belirtmeyin. Başka bir değerle kullanıldığında bu öncelik kazanır.

< mq> MQRFH2 klasörüne ilişkin kısıtlamalar

Bir ileti kuyruğa konduğunda, iletinin MQtanımlı özelliklerine göre işlenebilmesi için bir < mq> klasörü aranır. MQtanımlı özelliklerin verimli bir şekilde ayrıştırılmasına izin vermek için klasör için aşağıdaki kısıtlamalar geçerlidir:

- Yalnızca iletideki ilk önemli < mq> klasöründeki özellikler MQtarafından işlendi; iletideki diğer herhangi bir < mq> klasöründeki özellikler yoksayıldı.
- Klasör UTF-8içindeyse, klasörde yalnızca tek baytlık UTF-8 karakterlerine izin verilir. Klasördeki çok baytlı bir karakter, ayrıştırmanın başarısız olmasına ve iletinin reddedilmesine neden olabilir.
- < mq> klasörüne MQRFH2 gruplarını eklemeyin. Bir özellik değerinde U+003C Unicode karakterinin bulunması iletinin reddedilmesine neden olur.
- Klasörde çıkış dizgilerini kullanmayın. Bir çıkış dizgisi, ögenin gerçek değeri olarak kabul edilir.
- Klasör içinde yalnızca Unicode karakteri U+0020 beyaz alan olarak kabul edilir. Diğer tüm karakterler önemli olarak değerlendirilir ve klasör ayrıştırmasının başarısız olmasına ve iletinin reddedilmesine neden olabilir.

< mq> klasörünün ayrıştırılması başarısız olursa ya da klasör bu kısıtlamalara uymazsa, ileti CompCode **MQCC_FAILED** ve Reason **MQRC_RFH_RESTRICTED_FORMAT_ERR** ile reddedilir.

Geçerli olmayan MQRFH2 üstbilgileri

MQPUT, MQPUT1ya da MQGET çağrı işlemleri sırasında, hangi klasörlerin içerildiğini denetlemek ve klasörlerin özellikler içerip içermediğini saptamak için iletideki MQRFH2 üstbilgilerinin kısmi bir ayrıştırması oluşabilir. Geçerli olmayan MQRFH2 üstbilgilerine genel bakış.

Yapı geçerli olmadığı için iletinin kısmi ayrıştırması başarıyla tamamlanamıyorsa, örneğin, StrucLength alanı çok küçük:

- MQPUT ya da MQPUT1 çağrısı MQRC_RFH_ERROR neden koduyla başarısız olur; uygulamanın bazı IBM WebSphere MQ 7 seçenekleri içerdiği saptanabilir; böylece, var olan uygulamalar başarısız olmaz.
- MQGET çağrısı başarıyla döndü ve hatayı içeren MQRFH2 , sağladığınız arabellekte döndürüldü.

Belirli bir klasörün özellikler içerip içermediği saptanamadığı için kısmi ayrıştırma başarısız olursa, örneğin, klasör <<jmsolarak başlar; bu nedenle, klasör adı belirlenmeden ayrıştırma başarısız olur:

- MQPUT ya da MQPUT1 çağrısı MQRC_RFH_FORMAT_ERROR neden koduyla başarısız olur; uygulamanın bazı IBM WebSphere MQ 7 seçenekleri içerdiği saptanabilir; böylece, var olan uygulamalar başarısız olmaz.
- MQGET çağrısı başarıyla döndü ve hatayı içeren MQRFH2 , sağladığınız arabellekte döndürüldü.
- Kuyruk yöneticisi içinde dahili olarak bulunurken, hatalı biçimlendirilmiş klasör nedeniyle ileti reddedilmez, ancak klasör her zaman içinde hiçbir özellik yokmuş gibi işlenir.

Bir ileti, böyle bir sözdizimi hatası içeren bir klasörle kuyruk yöneticisi ağında akabilir; ancak, iletideki bir ya da daha çok klasör aşağıdaki gibidir:

- Geçerli
- Başarıyla ayrıştırıldı
- İletinin işlenmesinde kullanılır

Bu nedenle, algılama garanti edilmez.

Uygulamalarınızdan biri bir özelliğe erişmek için “MQSETMP-İleti özelliğini ayarla” sayfa 771ya da MQINQMP kullanıyorsa ve bu, bir MQRFH2 klasörünün tam olarak ayrıştırılmasına neden olursa, ayrıştırmanın tamamlanamayacağı bir hata saptanırsa, API çağrısına uygun bir dönüş koduyla gösterilir. Klasördeki hiçbir özellik uygulama tarafından kullanılamaz.

Bir MQRFH2 klasörünü tam olarak ayrıştırma girişiminde bulunulursa ve ayrıştırıcı tanınmayan öge öznitelikleri ya da tanınmayan bir veri tipi bulursa, ayrıştırma devam eder ve uyarı verilmeden başarıyla tamamlanır; bu bir ayrıştırma hatası oluşturmaz.

Kod sayfası dönüştürme

Bu bölümde kod kümesi adları ve CCSID ' ler, ulusal dil, z/OS dönüştürme, IBM i dönüştürme ve Unicode dönüştürme desteği açıklanmaktadır.

Her ulusal dil bölümünde aşağıdaki bilgiler listelenir:

- Desteklenen yerel CCSID ' ler
- Desteklenmeyen kod sayfası dönüştürmeleri

Bilgilerde aşağıdaki terimler kullanılır:

AIX **AIX**

IBM MQ for AIX ögesini gösterir.

Linux **Linux**

IBM MQ for Linux for Intel ve IBM MQ for Linux for zSeries' yi gösterir.

IBM i **OS/400**

IBM MQ for IBM i ögesini gösterir.

Windows **Windows**

IBM MQ for Windows ögesini gösterir.

z/OS **z/OS**

IBM MQ for z/OS ögesini gösterir.

Veri dönüştürmenin varsayılan değeri, hedef (alan) sistemde gerçekleştirilecek dönüştürmenin gerçekleştirilmesidir.

Kaynak ürün dönüştürmeyi destekliyorsa, kaynakta CONVERT kanal özneliği YES değerine ayarlanarak bir kanal ayarlanabilir ve veri alışverişi gerçekleştirilebilir.

Not:

1. IBM MQ MQI client bilgilerine ilişkin dönüştürme sunucuda gerçekleşir; bu nedenle, sunucunun istemci CCSID 'sinden sunucu CCSID 'sine dönüştürmeyi desteklemesi gerekir.
2. Dönüştürme, CSD/PTF tarafından en son IBM MQ sürümüne eklenen desteği içerebilir. Bu dönüştürmeyi etkinleştirmek üzere bir CSD/PTF kurmanız gerekip gerekmediğini görmek için en son hizmet düzeyinin içeriğini denetleyin.
3. IBM MQ kuyruk yöneticisi CCSID 'si Karışık ya da SBCS olmalıdır.
4. İşletim sistemi tarafından desteklenmeyen bazı CCSID ' ler (örneğin, AIX üzerinde 850) uygulama tarafından kullanılmaya devam edebilir ve IBM MQ kuyruk yöneticisi CCSID 'si olarak da ayarlanabilir. Bu yalnızca geriye doğru uyumluluk amacıyla kullanılabilir ve ilgili dönüştürme çizelgeleri kurulu değilse dönüştürme başarısız olur.

Bazı CCSID numaralarıyla bazı sektör kod kümesi adları arasında çapraz başvuru için bkz. [Çizelge 641 sayfa 927](#) .

İlgili başvurular

“Ulusal diller” sayfa 927

Bu bilgiler, IBM MQ tarafından desteklenen dilleri içerir.

Kod kümesi adları ve CCSID ' ler

Kod kümesi adları ve her kod kümesi adı için karşılık gelen CCSID ' ler.

z/OS

IBM MQ for z/OS , dile özgü çizelgelerde listelenenden daha fazla dönüştürme sağlar. Dönüştürmenin tam listesi için bkz. [Çizelge 674 sayfa 952](#).

<i>Çizelge 641. Kod kümesi adları ve CCSID ' ler</i>	
Kod kümesi adları	CCSID ' ler
ISO 8859-1	819
ISO 8859-2	912
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 (avro)	923
big5	950
eucJP	954 5050 33722
eucKR	970
eucTW	964
eucCN	1383
PCK	943
GBK	1386
koi8-r	878

Ulusal diller

Bu bilgiler, IBM MQ tarafından desteklenen dilleri içerir.






IBM MQ tarafından desteklenen diller şunlardır:

- ABD İngilizcesi-konuya bakın [“ABD İngilizcesi” sayfa 928](#)
- Almanca-konuya bakın [“Almanca” sayfa 929](#)
- Danca ve Norveççe-konuya bakın [“Danca ve Norveççe” sayfa 929](#)
- Fince ve İsveççe-konuya bakın [“Fince ve İsveççe” sayfa 930](#)
- İtalyanca-konuya bakın [“İtalyanca” sayfa 931](#)
- İspanyolca-konuya bakın [“İspanyolca” sayfa 932](#)
- İngiltere İngilizcesi/Galce-konuya bakın [“İngiltere İngilizcesi /Galce” sayfa 932](#)
- Fransızca-konuya bakın [“Fransızca” sayfa 933](#)
- Çok dilli-konuya bakın [“Çok dilli” sayfa 933](#)
- Portekizce-konuya bakın [“Portekizce” sayfa 934](#)
- İzlandaca-konuya bakın [“İzlandaca” sayfa 935](#)
- Doğu Avrupa dilleri-konuya bakın [“Doğu Avrupa dilleri” sayfa 936](#)
- Kiril-konuya bakın [“Kiril” sayfa 937](#)
- Estonya-konuya bakın [“Estonyaca” sayfa 938](#)
- Letonca ve Litvanca-konuya bakın [“Letonca ve Litvanca” sayfa 939](#)

- Ukrayanca-konuya bakın [“Ukraynaca” sayfa 940](#)
- Yunanca-konuya bakın [“Yunanca” sayfa 940](#)
- Türkçe-konuya bakın [“Türkçe” sayfa 941](#)
- İbranice-konuya bakın [“İbranice” sayfa 942](#)
- Farsça-konuya bakın [“Farsça” sayfa 944](#)
- Urduca- [“Urduca” sayfa 944](#) konusuna bakın
- Tay dili-konuya bakın [“Tay dili” sayfa 944](#)
- Lao-konuya bakın [“Lao” sayfa 945](#)
- Vietnamca-konuya bakın [“Vietnam dili” sayfa 945](#)
- Japonca Latin SBCS-konuya bakın [“Japonca-Latin SBCS” sayfa 946](#)
- Japonca Katakana SBCS-konuya bakın [“Japonca Katakana SBCS” sayfa 947](#)
- Japonca Kanji/Latin Karışık-konuya bakın [“Japonca Kanji/Latince Karışık” sayfa 948](#)
- Japonca Kanji/Katakana Karışık-konuya bakın [“Japonca Kanji/Katakana Karışık” sayfa 949](#)
- Korece-konuya bakın [“Korece” sayfa 950](#)
- Basitleştirilmiş Çince-konuya bakın [“Yalınlaştırılmış Çince” sayfa 951](#)
- Geleneksel Çince-konuya bakın [“Geleneksel Çince” sayfa 951](#)

ABD İngilizcesi

ABD İngilizcesi için CCSID ve CCSID dönüşümüne ilişkin ayrıntılar.

Çizelge 642. Desteklenen platformlarda ABD İngilizcesi için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i  z/OS	37, 924, 1140
 AIX	819, 923, 5348
 Windows	437, 850, 1252, 5348, 858
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayfası:

37

923, 858 kod sayfalarına dönüştürülmez

924



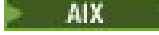


437, 858, 1051, 1140, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1140

924, 1051, 1275 kod sayfalarına dönüştürülmez

Almanca

Almanca için CCSID ve CCSID dönüştürmesinin ayrıntıları.

<i>Çizelge 643. Desteklenen altyapılarda Almanca için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	273, 924, 1141
 z/OS	
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayfası:

273

858, 923, 924, 1275 kod sayfalarına dönüştürülmez

924



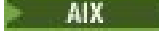


273, 437, 858, 1051, 1141, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1141

924, 1051, 1275 kod sayfalarına dönüştürülmez

Danca ve Norveççe

Danca ve Norveççe için CCSID ve CCSID dönüşümünün ayrıntıları.

<i>Çizelge 644. Desteklenen altyapılarda Danca ve Norveççe için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	277, 924, 1142.
 z/OS	
 AIX	819, 923, 5348
 Windows	850, 858, 865, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i

IBM i

Kod sayfası:

277

858, 923, 924, 1275 kod sayfalarına dönüştürülmez

924

277, 858, 865, 1051, 1142, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1142

924, 865, 1051, 1275 kod sayfalarına dönüştürülmez

AIX

AIX

Kod sayfası:

819

Kod sayfası 865 'e dönüştürülmez

Windows

Windows






Kod sayfası:

865

1051, 1275 kod sayfalarına dönüştürülmez

Fince ve İsveççe

Fince ve İsveççe için CCSID ve CCSID dönüşümünün ayrıntıları.

<i>Çizelge 645. Desteklenen platformlarda Fince ve İsveççe için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	278, 924, 1143
 z/OS	
 AIX	819, 923, 5348
 Windows	437, 850, 858, 865, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i

IBM i

Kod sayfası:

278

858, 923, 924, 1275 kod sayfalarına dönüştürülmez

924

278, 437, 858, 865, 1051, 1143, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1143

Kod sayfalarına dönüştürülmez: 865, 924, 1051, 1275

AIX



Kod sayısı:

819

Kod sayısı 865 'e dönüştürülmez

850

Kod sayısı 865 'e dönüştürülmez

Windows



Kod sayısı:

865

1051, 1275 kod sayfalarına dönüştürülmez

İtalyanca

İtalyanca için CCSID ve CCSID dönüştürmesinin ayrıntıları.

<i>Çizelge 646. Desteklenen altyapılarda İtalyanca için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
IBM i	280, 924, 1144
z/OS	
AIX	819, 923, 5348
Windows	437, 850, 858, 1252, 5348
Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayısı:

280

858, 923, 924, 1275 kod sayfalarına dönüştürülmez

924






280, 437, 858, 1051, 1144, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1144

924, 1051, 1275 kod sayfalarına dönüştürülmez

İspanyolca

İspanyolca için CCSID ve CCSID dönüştürmesinin ayrıntıları.

<i>Çizelge 647. Desteklenen altyapılarda İspanyolca için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	284, 924, 1145
 z/OS	
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayfası:

284

858, 923, 924, 1275 kod sayfalarına dönüştürülmez

924






284, 437, 858, 1051, 1145, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1145

924, 1051, 1275 kod sayfalarına dönüştürülmez

İngiltere İngilizcesi /Galce

İngiltere İngilizcesi/Galce için CCSID dönüştürme ve CCSID ' nin ayrıntıları.

<i>Çizelge 648. Desteklenen platformlarda İngiltere İngilizcesi/Galce için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	285, 924, 1146
 z/OS	
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i

IBM i

Kod sayfası:

285

858, 923, 924, 1275 kod sayfalarına dönüştürülmez

924






285, 437, 858, 1051, 1146, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1146

924, 1051, 1275 kod sayfalarına dönüştürülmez

Fransızca

Fransızca için CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 649. Desteklenen altyapılarda Fransızca için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	297, 924, 1147
 z/OS	
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i

IBM i

Kod sayfası:

297

Kod sayfalarına dönüştürülmez: 858, 923, 924, 1275, 5348

924



297, 437, 858, 1051, 1147, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1147

924, 1051, 1275 kod sayfalarına dönüştürülmez

Çok dilli

Çok Dilli için CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 650. Desteklenen altyapılarda çok dilli dönüştürme için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	500, 924, 1148
 z/OS	

Çizelge 650. Desteklenen altyapılarda çok dilli dönüştürme için yerel CCSID ' ler (devamı var)

Hizmet olarak sunulan	Yerel CCSID ' ler
 AIX	819, 923, 5348
 Windows	437, 850, 858, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayfası:

500

858, 923 kod sayfalarına dönüştürülmez

924

437, 858, 1051, 1148, 1252, 1275, 5348 kod sayfalarına dönüştürülmez






1148

924, 1051, 1275 kod sayfalarına dönüştürülmez

Portekizce

Portekizce için CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 651. Desteklenen altyapılarda Portekizce için yerel CCSID ' ler

Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	37, 500, 924, 1140
 z/OS	500, 924, 1140
 AIX	819, 923, 5348
 Windows	850, 858, 860, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayfası:

37

858, 923, 1275 kod sayfalarına dönüştürülmez

500

858, 923, 1275 kod sayfalarına dönüştürülmez

924

858, 860, 1051, 1140, 1252, 1275, 5348 kod sayfalarına dönüştürülmez

1140

860, 924, 1051, 1275 kod sayfalarına dönüştürülmez

Windows

Windows






Kod sayfası:

860

1051, 1275 kod sayfalarına dönüştürülmez

İzlandaca

İzlandaca CCSID ve CCSID dönüşümünün ayrıntıları.

Çizelge 652. Desteklenen platformlarda İzlandaca için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	871, 924, 1149
 z/OS	
 AIX	819, 923, 5348
 Windows	850, 858, 861, 1252, 5348
 Linux	819, 923
Apple istemcisi	1275

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i

IBM i

Kod sayfası:

871

Kod sayfalarına dönüştürülmez: 858, 923, 924, 1275, 5348

924

Kod sayfalarına dönüştürülmez: 858, 861, 871, 1051, 1149, 1252, 1275, 5348

1149

924, 1051, 1275 kod sayfalarına dönüştürülmez

Windows

Windows






Kod sayfası:

861

1051, 1275 kod sayfalarına dönüştürülmez

Doğu Avrupa dilleri

Doğu Avrupa Dillerine ilişkin CCSID ve CCSID dönüştürmesinin ayrıntıları. Bu CCSID 'leri kullanan tipik diller arasında Arnavutça, Hırvatça, Çekçe, Macarca, Lehçe, Rumence, Sırpça, Slovakça ve Slovence yer alır.

Çizelge 653. Desteklenen platformlarda Doğu Avrupa dilleri için yerel CCSID 'ler	
Hizmet olarak sunulan	Yerel CCSID 'ler
 IBM i	870, 1153
 z/OS	
 Windows	852, 1250, 5346, 9044
 AIX	912
 Linux	
Doğu Avrupa Apple istemcisi	1282
Romen Apple istemcisi	1285
Hırvat Apple müşterisi	1284

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS



Kod sayfası:

870

1284, 1285 kod sayfalarına dönüştürülmez

1153

1250, 1284, 1285 kod sayfalarına dönüştürülmez

IBM i



Kod sayfası:

870

1284, 1285, 5346, 9044 kod sayfalarına dönüştürülmez

1153

1282, 1284, 1285, 5346, 9044 kod sayfalarına dönüştürülmez

, Linux



Kod sayfası:

912

1284, 1285 kod sayfalarına dönüştürülmez

Windows

Windows

Kod sayfası:

852

1284, 1285 kod sayfalarına dönüştürülmez

1250






1284, 1285 kod sayfalarına dönüştürülmez

9044

912, 1282, 1284, 1285 kod sayfalarına dönüştürülmez

Kiril

Kiril için CCSID ve CCSID dönüştürmesinin ayrıntıları. Bu CCSID 'leri kullanan tipik diller arasında Beyaz Rusya, Bulgarca, Makedonca, Rusça ve Sırpça yer alır.

<i>Çizelge 654. Desteklenen altyapılarda Kiril için yerel CCSID 'ler</i>	
Hizmet olarak sunulan	Yerel CCSID 'ler
 z/OS	1025
 IBM i	880, 1025
 Windows	855, 866, 1131, 1251, 5347
 AIX	915
 Linux	
Apple istemcisi	1283

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i

IBM i

Kod sayfası:

880

Kod sayfalarına dönüştürülmez: 855, 866, 878, 1131, 5347

1025

Kod sayfalarına dönüştürülmez: 878, 5347

Windows

Windows

Kod sayfası:

855

1131 kod sayfasına dönüştürülmez

866






1131 kod sayfasına dönüştürülmez

1131

855, 866, 880, 1283 kod sayfalarına dönüştürülmez

Estonyaca

Estonca için CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 655. Desteklenen altyapılarda Estonca için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	1122, 1157
 z/OS	
 Windows	902, 922, 1257, 5353, 9449
 AIX	902, 922
 Linux	

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS



Kod sayfası:

1122

902, 1157, 9449 kod sayfalarına dönüştürülmez

1157

922, 1122, 1257, 9449 kod sayfalarına dönüştürülmez

IBM i



Kod sayfası:

1122

902, 5353, 9449 kod sayfalarına dönüştürülmez

1157

922, 5353, 9449 kod sayfalarına dönüştürülmez

Linux



Kod sayfası:

902

922, 1122, 9449 kod sayfalarına dönüştürülmez

922

902, 1157, 9449 kod sayfalarına dönüştürülmez

Windows



Kod sayfası:

5353

Kod sayfası 9449 'a dönüştürülmez

9449






902, 922, 1122, 1157, 1257, 5353 kod sayfalarına dönüştürülmez

902

922, 1122, 9449 kod sayfalarına dönüştürülmez

Letonca ve Litvanca

Letonya ve Litvanca için CCSID ve CCSID dönüşümünün ayrıntıları.

Çizelge 656. Desteklenen platformlarda Letonca ve Litvanca için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	1112, 1156
 z/OS	
 Windows	901, 921, 1257, 5353, 9449
 AIX	901, 921
 Linux	

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS



Kod sayfası:

1112

901, 1156, 9449 kod sayfalarına dönüştürülmez

1156

901, 1156, 9449 kod sayfalarına dönüştürülmez

IBM i



Kod sayfası:

1112

5353 kod sayfasına dönüştürülmez

1153

921, 5353, 9449 kod sayfalarına dönüştürülmez

Linux



Kod sayfası:

902

921, 1112, 1257, 9449 kod sayfalarına dönüştürülmez

921

901, 1156, 9449 kod sayfalarına dönüştürülmez

Windows

Windows

Kod sayfası:

901

921, 1112, 1257, 9449 kod sayfalarına dönüştürülmez

5355




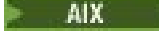

Kod sayfası 9449 'a dönüştürülmez

9449

901, 921, 1112, 1156, 1257 kod sayfalarına dönüştürülmez

Ukraynaca

Ukraynaca için CCSID ve CCSID dönüşümünün ayrıntıları.

<i>Çizelge 657. Desteklenen platformlarda Ukraynca için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	1123
 z/OS	
 Windows	1124, 1125, 1251, 5347.
 AIX	1124
 Linux	

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i

IBM i

Kod sayfası:

1123

5347 kod sayfasına dönüştürülmez

Windows

Windows



Kod sayfası:

1125




1123 kod sayfasına dönüştürülmez

Yunanca

Yunanca için CCSID ve CCSID dönüşümünün ayrıntıları.

<i>Çizelge 658. Desteklenen altyapılarda Yunanca için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	875
 z/OS	

Çizelge 658. Desteklenen altyapılarda Yunanca için yerel CCSID ' ler (devamı var)

Hizmet olarak sunulan	Yerel CCSID ' ler
 Windows	869, 1253, 5349
 AIX  Linux NCR	813
Apple istemcisi	1280
DOS istemcisi	737

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, diğer altyapıların yerel CCSID 'leri olan yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayfası:

875

5349 kod sayfasına dönüştürülmez

Windows



Kod sayfası:

1253

Kod sayfası 737 'ye dönüştürülmez






5349

Kod sayfası 737 'ye dönüştürülmez

Türkçe

Türkçe için CCSID ve CCSID dönüşümünün ayrıntıları.

Çizelge 659. Desteklenen altyapılarda Türkçe için yerel CCSID ' ler

Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i  z/OS	1026
 Windows	857, 1254, 5350
 AIX  Linux	920
Apple istemcisi	1281

İstemci dışındaki tüm altyapılar, aşağıdaki kural dışı durumlar dışında, kendi yerel CCSID 'leri ile diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayfası:

1026

Kod sayfası 5350 'ye dönüştürülmez

İbranice

İbranice için CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 660. Desteklenen altyapılarda İbranice için yerel CCSID 'ler	
Hizmet olarak sunulan	Yerel CCSID 'ler
z/OS	424, 803, 4899, 12712
IBM i	424
AIX	916, 9048
Windows	1255, 5351
Linux	916

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS



Kod sayfası:

424

Kod sayfalarına dönüştürülmez: 867, 4899, 9048, 12712

803

867, 4899, 5351, 9048, 12712 kod sayfalarına dönüştürülmez

4899

424, 803, 856, 862, 916, 1255 kod sayfalarına dönüştürülmez

12712

424, 803, 856, 916, 1255 kod sayfalarına dönüştürülmez

IBM i



Kod sayfası:

424

803, 867, 4899, 5351, 9048, 12712 kod sayfalarına dönüştürülmez

Kod sayfası 424, 856 çeşitlemesi olan CCSID 4952 'ye ve CCSID 4952 'den de dönüştürülür.

AIX



Kod sayfası:

916

Kod sayfalarına dönüştürülmez: 867, 4899, 9048, 12712

9048

424, 803, 856, 862, 916, 1255 kod sayfalarına dönüştürülmez

Windows



Kod sayısı:

1255

Kod sayfalarına dönüştürülmez: 867, 4899, 9048, 12712

5351

803 kod sayfasına dönüştürülmez

Arapça

Arapça için CCSID ve CCSID dönüştürme ayrıntıları

<i>Çizelge 661. Desteklenen altyapılarda Arapça için yerel CCSID 'ler</i>	
Hizmet olarak sunulan	Yerel CCSID 'ler
IBM i	420
z/OS	
AIX	1046, 1089
	1089 (bkz. not)
Windows	720, 864, 1256, 5352
Linux	1089

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID'leri arasında dönüştürmeyi destekler.

IBM i



Kod sayısı:

420

5352 kod sayfasına dönüştürülmez

Linux, Tru64



Kod sayısı:

1089

Kod sayısı 720 'ye dönüştürülmez

Windows



Kod sayısı:

720






1089, 5352 kod sayfalarına dönüştürülmez

5352

Kod sayısı 720 'ye dönüştürülmez

Farsça

Farsça için CCSID ve CCSID dönüştürmesinin ayrıntıları.






Çizelge 662. Desteklenen platformlarda Farsça için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i  z/OS	1097
 AIX  Linux  Windows	1098 (bkz. not)

Not: Bu altyapılara ilişkin yerel CCSID standartlaştırılmadı ve değişebilir.

Tüm platformlar, yerel CCSID 'leri ile diğer platformların yerel CCSID' leri arasında dönüştürmeyi destekler.

Urduca

Urduya ilişkin CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 663. Desteklenen altyapılarda Urduca için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i  z/OS	918
 Windows	868
 AIX  Linux	1006

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayısı:






918

1006 kod sayfasına dönüştürülmez

Tay dili

Tay dili için CCSID ve CCSID dönüştürmesi ayrıntıları.

Çizelge 664. Desteklenen altyapılarda Tay dili için yerel CCSID ' ler

Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i  z/OS	838
 AIX  Linux  Windows	874 (bkz. not)






Not: Bu altyapılara ilişkin yerel CCSID standartlaştırılmadı ve değişebilir.

Tüm platformlar, yerel CCSID 'leri ile diğer platformların yerel CCSID' leri arasında dönüştürmeyi destekler.

Lao

Lao için CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 665. Desteklenen altyapılarda Lao için yerel CCSID ' ler






Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i  z/OS	1132
 AIX  Linux  Windows	1133

Tüm platformlar, yerel CCSID 'leri ile diğer platformların yerel CCSID' leri arasında dönüştürmeyi destekler.

Vietnam dili

Vietnamca için CCSID dönüştürme ve CCSID dönüştürme ayrıntıları.

Çizelge 666. Desteklenen altyapılarda Vietnamca için yerel CCSID ' ler

Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i  z/OS	1130
 Windows	1258, 5354
 AIX  Linux	1129

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

IBM i



Kod sayfası:

1130

1129, 5354 kod sayfalarına dönüştürülmez

Japonca-Latin SBCS

Japonca Latin SBCS için CCSID ve CCSID dönüştürmesinin ayrıntıları.

<i>Çizelge 667. Desteklenen altyapılarda Japonca Latin SBCS için yerel CCSID ' ler</i>	
Hizmet olarak sunulan	Yerel CCSID ' ler
IBM i z/OS	1027
AIX	932, 5050, 33722 (bkz. Not 1)
Windows	932, 943 (bkz. Not 2)
Linux	943, 5050

Not:

- 5050 ve 33722, AIX üzerinde temel kod sayfası 954 ile ilgili CCSID ' lerdir. İşletim sistemi tarafından bildirilen CCSID 33722 'dir.
- Windows NT , 932 kod sayfasını kullanır, ancak bu, en iyi 943 CCSID 'si ile gösterilir. Ancak, IBM MQ 'in tüm platformları bu CCSID' yi desteklemez.
IBM MQ for Windows CCSID 932 'de, kod sayfası 932 'yi göstermek için kullanılır, ancak . . /conv /table/ccsid . tbl dosyasında yapılan bir değişiklik, kullanılan CCSID ' yi 943 olarak değiştirir.

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS



Kod sayfası:

1027

Kod sayfalarına dönüştürülmez: 932, 942, 943, 954, 5050, 33722

IBM i



Kod sayfası:

1027

Kod sayfası 932 'ye dönüştürülmez

AIX



Kod sayfası:

932

1027 kod sayfasına dönüştürülmez

5050

1027 kod sayfasına dönüştürülmez

33722

1027 kod sayfasına dönüştürülmez

Linux

Kod sayfası:

943

1027 kod sayfasına dönüştürülmez

5050

1027 kod sayfasına dönüştürülmez

Japonca Katakana SBCS

Japonca Katakana SBCS için CCSID ve CCSID dönüştürmesi ayrıntıları.

Hizmet olarak sunulan	Yerel CCSID ' ler
IBM i z/OS	290
AIX	932, 5050, 33722 (bkz. Not 1)
Windows	932, 943 (bkz. Not 2)
Linux	943, 5050

Not:

- AIX 5050 ve 33722, AIX üzerinde temel kod sayfası 954 ile ilgili CCSID ' lerdir. İşletim sistemi tarafından bildirilen CCSID 33722 'dir.
- Windows NT , 932 kod sayfasını kullanır, ancak bu, en iyi 943 CCSID 'si ile gösterilir. Ancak, IBM MQ 'in tüm platformları bu CCSID' yi desteklemez.
IBM MQ for Windows CCSID 932 'de, kod sayfası 932 'yi göstermek için kullanılır, ancak . . /conv/ table/ccsid .tbl dosyasında yapılan bir değişiklik, kullanılan CCSID ' yi 943 olarak değiştirir.
- Önceki dönüştürmelere ek olarak IBM MQ , aşağıdaki altyapılarda CCSID 897 'den CCSID 37, 273, 277, 278, 280, 284, 285, 290, 297, 437, 500, 819, 850, 1027 ve 1252 'ye dönüştürmeyi destekler:
 - AIX
 - Linux

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS

Kod sayfası:

290

Kod sayfalarına dönüştürülmez: 932, 943, 954, 5050, 33722

IBM i

IBM i

Kod sayfası:

290

Kod sayfası 932 'ye dönüştürülmez

AIX

AIX

Kod sayfası:

932

290, 897 kod sayfalarına dönüştürülmez

5050

290, 897 kod sayfalarına dönüştürülmez

33722

290, 897 kod sayfalarına dönüştürülmez

Linux

Linux

Kod sayfası:

943






290, 897 kod sayfalarına dönüştürülmez

5050




290, 897 kod sayfalarına dönüştürülmez

Japonca Kanji/Latince Karışık

Japonca Kanji/Latin Karışık için CCSID ve CCSID dönüşümünün ayrıntıları.

Çizelge 669. Desteklenen platformlarda karıştırılmış Japonca Kanji/Latin için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 IBM i	1399, 5035 (bkz. Not 1)
 z/OS	
 AIX	932, 5050, 33722 (bkz. Not 2)
 Windows	932, 943 (bkz. Not 4)
 Linux	943, 5050

Not:

-   5035, kod sayfası 939 ile ilişkili bir CCSID 'dir.
-  5050 ve 33722, AIX üzerinde temel kod sayfası 954 ile ilgili CCSID ' lerdir. İşletim sistemi tarafından bildirilen CCSID 33722 'dir.

3. **Windows** Windows NT , 932 kod sayfasını kullanır, ancak bu, en iyi 943 CCSID 'si ile gösterilir. Ancak, IBM MQ 'in tüm platformları bu CCSID' yi desteklemez.

IBM MQ for Windows CCSID 932 'de, kod sayfası 932 'yi göstermek için kullanılır, ancak . . /conv/table/ccsid.tbl dosyasında yapılan bir değişiklik, kullanılan CCSID ' yi 943 olarak değiştirir.

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS

z/OS

Kod sayfası:

1399

954, 5035, 5050, 33722 kod sayfalarına dönüştürülmez

5035

954, 1399, 5050, 33722 kod sayfalarına dönüştürülmez

IBM i

IBM i

Kod sayfası:

1399

5039 kod sayfasına dönüştürülmez

5035

5039 kod sayfasına dönüştürülmez

Japonca Kanji/Katakana Karışık

Japonca Kanji/Katakana için CCSID ve CCSID dönüşümünün ayrıntıları Karışık.

Çizelge 670. Desteklenen platformlarda karıştırılmış Japonca Kanji/Katakana için yerel CCSID ' ler

Hizmet olarak sunulan	Yerel CCSID ' ler
z/OS z/OS	1390, 5026 (bkz. Not "1" sayfa 949)
IBM i IBM i	5026 (bkz. Not "1" sayfa 949)
AIX AIX	932, 5050, 33722 (bkz. Not "2" sayfa 950)
Windows Windows	932, 943 (bkz. Not "3" sayfa 950)
Linux Linux	943, 5050

Not:

1. **IBM i** **z/OS** EBCDIC ' de CCSID 1390 ve 5026 'nın tek baytlık kipi, temel Latince için tipik ya da değişmez yerleşim düzenine göre farklı konumlarda küçük harfli karakterler içerir. Bu nedenle, ileti verileri diğer CCSID ' lere dönüştürüldüğünde verilerin kaybolmadığından emin olun. Ayrıca, bu CCSID ' lerin kuyruk yöneticisinin varsayılan CCSID 'si olarak kullanılması, diğer kuyruk yöneticileriyle iletişim kurulurken sorunlara neden olabilir. Örneğin, küçük harfli karakterler kullanan kanal adları uzak sistemde doğru şekilde yorumlanmayabilir. 5026, kod sayfası 930 ile ilişkili bir CCSID 'dir. CCSID 5026, Japonca Katakana (DBCS) özelliği seçildiğinde IBM i üzerinde bildirilen CCSID 'dir.

2. **AIX** 5050 ve 33722, AIX üzerinde temel kod sayfası 954 ile ilgili CCSID ' lerdir. İşletim sistemi tarafından bildirilen CCSID 33722 'dir.
3. **Windows** Windows NT , 932 kod sayfasını kullanır, ancak bu, en iyi 943 CCSID 'si ile gösterilir. Ancak, IBM MQ 'in tüm platformları bu CCSID' yi desteklemez.

IBM MQ for Windows üzerinde, CCSID 932, kod sayfası 932 'yi göstermek için kullanılır, ancak . . / conv/table/ccsid.tbl dosyasında, kullanılan CCSID ' yi 943 olarak değiştiren bir değişiklik yapılabilir.

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS

z/OS

Kod sayfası:

1390

954, 5026, 5050, 33722 kod sayfalarına dönüştürülmez

Küçük harf karakterleri kabul etmez.

5026

954, 1390, 5050, 33722 kod sayfalarına dönüştürülmez

IBM i

IBM i

Kod sayfası:

5026

1390, 5039 kod sayfalarına dönüştürülmez

Korece

Korece için CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 671. Desteklenen altyapılarda Korece için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
IBM i IBM i z/OS z/OS	933, 1364
AIX AIX Linux Linux	970
Windows Windows	949, 1363

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS

z/OS

Kod sayfası:

933






Kod sayfası 970 'e dönüştürülmez

1364


Kod sayfası 970 'e dönüştürülmez

Yalınlaştırılmış Çince

Basitleştirilmiş Çince için CCSID ve CCSID dönüştürmesinin ayrıntıları.




Çizelge 672. Desteklenen altyapılarda Basitleştirilmiş Çince için yerel CCSID ' ler	
Hizmet olarak sunulan	Yerel CCSID ' ler
 z/OS	935, 1388
 IBM i	935, 1388
 AIX	1383, 1386
 Windows	1381, 1386 (bkz. Not 2)
 Linux	1383

Not:


1.  Windows , 936 kod sayfasını kullanır, ancak bu, en iyi 1386 CCSID 'si ile gösterilir. Ancak, IBM MQ 'in tüm platformları bu CCSID' yi desteklemez.

IBM MQ for Windows CCSID 1381 'de, kod sayfası 936 'yı göstermek için kullanılır, ancak . . /conv/table/ccsid.tbl dosyasında değişiklik yapılarak, kullanılan CCSID 1386 'ya çevrilir.

2. IBM MQ , Çince GB18030 standardını destekler.

   z/OS, Windows ve Linux sistemlerinde, dönüştürme desteği Unicode (UTF-8 ve UTF-16) ile CCSID 1388 (GB18030 uzantılı EBCDIC), Unicode (UTF-8 ve UTF-16) ve CCSID 5488 (GB18030) ile CCSID 1388 ile CCSID 5488 arasında sağlanır.

Not: GB18030 karakterlerini kullanabilmek için CCSID 5488 olarak ayarlanmalıdır. Ancak, IBM MQ Explorer ya da IBM MQ Console ile yaratılan bir kuyruk yöneticisinde CCSID ayarlanamaz. Bunun yerine, kuyruk yöneticisini 5488 CCSID değeriyle CLI kullanarak yaratmanız ya da kuyruk yöneticisini yarattıktan sonra CCSID ' yi değiştirmek için CLI komut satırını kullanmanız gerekir.

 IBM i işletim sisteminde, işletim sistemi tarafından Unicode (UTF-8 ve UTF-16) ile CCSID 1388 (GB18030 uzantılı EBCDIC) arasında dönüştürme için destek sağlanır.

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID' leri arasında dönüştürmeyi destekler.

z/OS



Kod sayfası:

935

1383 kod sayfasına dönüştürülmez






1388

1383 kod sayfasına dönüştürülmez

Geleneksel Çince

Geleneksel Çince için CCSID ve CCSID dönüştürmesinin ayrıntıları.

Çizelge 673. Desteklenen altyapılarda Geleneksel Çince için yerel CCSID 'ler

Hizmet olarak sunulan	Yerel CCSID 'ler
 IBM i  z/OS	937
 Windows	950
 AIX  Linux	950, 964

Tüm altyapılar, aşağıdaki kural dışı durumlar dışında, yerel CCSID 'leri ve diğer altyapıların yerel CCSID'leri arasında dönüştürmeyi destekler.

z/OS



Kod sayfası:

937

Kod sayfası 964 'e dönüştürülmez

1388

1383 kod sayfasına dönüştürülmez

Linux



Kod sayfası:

964

Kod sayfası 938 'e dönüştürülmez



z/OS dönüştürme desteği

Desteklenen CCSID dönüştürmelerinin listesi.

Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği

CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 1011-924, 920-924, 914-921, 914-914-921 9025, 9030, 9044, 9049, 9056, 9061, 9066, 13121, 13488, 16804, 17248, 17584, 25473, 25479, 25480, 25617, 25619, 25664, 28709
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, 4706, 836, 4929, 449, 4946, 4934, 4934
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, 5348, 9044, 9049, 9056, 9061, 9066, 13488, 17248, 17584

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 916, 920, 923-924, 1009, 1025-1027, 1040-1043, 1047, 1047, 1047, 1047
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, 1043, 1043, 1043
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, 1025-1071, 1043, 1040-1043, 1043, 1043, 1043
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, 1043, 10440
281	1047
282	500, 1047, 1200, 1208, 13488, 17584
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, 1043, 1040-1043, 1043, 1043
285	37, 256, 273, 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, 1047, 1051, 1040-1047
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, 855, 857, 860-865, 895-897, 897, 1009, 825-1040-1047, 1047, 1112, 1112, 1112, 1139, 1139, 1139, 1138, 1138, 1138, 1138, 1138, 113213213-10213-10213-10213-10213-9-108-10213213213213-10213 2, 82, 9-26-102, 9-102268, 9-102, 268, 268, 268, 92268, 268, 268, 268, 268, 268, 82, 92, 268, 265, 268, 911, 26-26-262, 262, 265, 265, 268, 265, 262, 262, 26-26-262, 265, 265, 265, 2, 2, 265, 2, 2, 265, 262, 265, 2, 262, 2, 262, 2, 26-2, 2, 2, 268
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, 10443, 1040-1043, 1047, 10440
300	301, 941, 1200, 1208, 1351, 4396, 8492, 13488, 16684, 17584
301	300, 941, 1200, 1208, 1351, 4396, 8492, 13488, 16684, 17584

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 709
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, 248
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, 912, 916, 920, 1009, 1025-1043, 7012, 1122, 1200, 1208, 1252-1253, 1280, 897, 903, 903, 903, 912, 912, 449
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, 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, 84-875, 897, 903, 905, 905, 912, 914-916, 10-940, 10-940, 10-940, 940 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, 895, 897, 903-905, 9145, 775, 775, 833, 836, 838, 850-832, 850-858, 850-871 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, 8121, 8229, 8482
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, 8121, 8482, 8612, 1212
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, 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, 49-953, 449, 449

Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)

CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 9547, 4953, 5123
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, 1043, 1040-1043, 863, 1097, 499
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, 850, 857-861, 871, 871
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, 852, 1041-1043, 1497
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
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, 909, 1051, 1097
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, 4946, 4934, 4946, 880, 4971, 1027, 1027
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, 850, 858, 860, 4929, 4934, 905, 905, 905, 905, 1025-1025-1097, 1097
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

Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)

CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 857, 860-861, 863, 880, 880, 880, 880, 880, 880, 880, 980, 880, 880, 916, 916, 897, 920, 920, 920, 920, 920, 920, 920, 920, 920, 920, 920, 949
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, 915 -916, 920, 1009, 1025-1027, 840-1043, 871, 871, 871, 871, 8746
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, 870, 874-875, 880, 897, 903, 912, 916, 920, 923-924, 10033, 852, 1011-1027, 10-10409, 9, 914-10496
872	259, 808, 858-859, 923-924, 1140-1149, 1153-1155, 1200, 1208, 5347, 5348, 13488, 17584
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, 10441-1043, 1200, 1208, 1252, 4909, 4934, 4946, 897, 875, 903, 70-49123, 449
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, 1012, 1088, 852, 857, 880, 8609, 871, 880, 880, 897, 871, 897, 897, 897, 903, 903, 930, 897, 897, 930, 930, 897, 930, 930, 930, 897
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, 1025-10488, 1041-1043, 1112, 1122, 875, 878, 878, 897, 978, 897, 947, 947, 947
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, 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, 10441-1043, 1200, 1208, 1252, 4386, 4909, 4934, 4946, 9048, 8483, 4946, 49489, 4948, 4953
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

Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
903	37, 273, 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, 897, 912, 912, 916, 916, 916
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
912	37, 273, 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, 4959, 4948, 4934, 916, 916, 916, 916, 916, 916, 920, 930, 920, 10-1041-1043, 10440, 10440, 10440, 10440, 1, 8, 8, 108, 8, 8, 8, 8, 8, 8, 8, 8, 940
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, 84-875, 897, 903, 903, 912, 912, 4954, 4954
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, 949, 9-4971, 12
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, 709
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

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
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
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

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
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
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, 1051, 1051, 1051
1026	37, 256, 273, 273, 277-278, 280, 284-285, 290, 297, 367, 423, 437, 500, 737, 775, 813, 819, 833, 836, 838, 850, 852, 857, 860-865, 869-871, 874-875, 880, 897, 903, 905, 912, 916, 913, 1009, 1025, 1027, 1040, 1043, 1047, 1047, 10161
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, 1047, 909, 494149, 1112
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, 17684, 17684
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, 1052, 855, 857, 860-871, 84-871, 84-871, 84-871, 8749, 8749, 84168, 84-8449, 8449, 8449, 8449, 8449, 8449, 8449, 8449, 8449, 499,449, 8449, 449, 41,449, 8449, 449

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
1042	37, 273, 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, 1041, 1043, 1088, 1200, 852, 857, 871, 860-861, 871, 871, 880
1043	37, 273, 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, 1041, 1042, 1088, 1114, 1200, 871, 860-84-875, 875, 880, 894, 909, 904, 909
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, 1208, 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, 9049, 644
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
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, 5483, 8353
1114	37, 437, 500, 819, 836, 850, 904, 1043, 1115, 1200, 1208, 4932, 4946, 5210-5211, 8229, 13488, 17584, 25480, 25619, 28709

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 880, 5123, 8353
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
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

Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)

CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 10-1162, 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
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
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

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
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
1280	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
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

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 122, 1139, 1252, 4929, 4932, 852, 4946, 4948, 4951
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, 903, 903, 54949, 44948, 4953
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, 857, 857, 860-871, 891, 1009, 89, 4951
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, 903, 1009, 1025-1027, 1040-1043, 1088, 1112, 1114-1115, 1122, 1252, 4386, 4929, 4946, 852, 855, 857, 89, 870, 870, 871, 871, 104954, 4951
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, 997, 903, 903, 4953

<i>Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)</i>	
CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 903, 4953, 4934, 448
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
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, 10-1149, 1153-1158, 1160-1162, 1164, 1200, 1208, 1252, 1275, 4946, 8229, 13488, 17584, 289

Çizelge 674. IBM MQ for z/OS CCSID dönüştürme desteği (devamı var)

CCSID	CCSID 'ye ve CCSID' den dönüştürür
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, 912, 914-90-916, 20-940, 912, 9-914-921, 912 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, 852, 855, 860-897, 897, 895-897, 897, 897
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
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, 4959, 4951
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, 1208, 850, 852, 860-865, 897, 897, 903, 903, 903, 903, 903, 449, 903, 449

IBM tarafından desteklenen CCSID ' lerin ve dönüştürmelerinin tam listesini uygun IBM i yayınında bulabilirsiniz.

Desteklenen kod sayfaları [Desteklenen CCSID eşlemeleri](#)' de listelenir.

Unicode dönüştürme desteği

Bazı platformlar, kullanıcı verilerinin Unicode kodlamasına ya da Unicode kodlamasından dönüştürülmesini destekler. Desteklenen iki Unicode kodlaması biçimi UTF-16 (CCSID 1200, 13488 ve 17584) ve UTF-8 (CCSID 1208) biçimleridir. Desteklenen en son Unicode sürümünü gösterdikleri için CCSID 1200 ya da 1208 kullanmalısınız.

UTF-16 vekil çiftleri (U + FFFF üzerinde bir Unicode kod noktasını gösteren X'D800' aralığındaki 2 baytlık UTF-16 karakter çifti) desteklenir. Hedef CCSID, UTF-16 yedek çifti tarafından gösterilen bir kod noktası için eşleme içermiyorsa, karakter çifti tek bir yerine koyma karakterine dönüştürülür.

IBM MQ, karakter sıralarını birleştirmeyi destekler. Bu, bazı durumlarda kaynak CCSID 'de önceden oluşturulmuş bir karakterin hedef CCSID' de ya da diğer bir şekilde bir araya getirilmiş karakter sırasına dönüştürüleceği anlamına gelir.

Not: IBM MQ , UTF-16 kuyruk yöneticisi CCSID ' leri desteklemediğinden ileti üstbilgisi verileri UTF-16biçiminde kodlanamıyor.

Unicode için IBM MQ AIX desteği

AIX

IBM MQ for AIX dönüşümünde, aşağıdaki listede Unicode CCSID 'leri olmayan CCSID' ler için desteklenen Unicode CCSID ' leri (tercihen 1200 ya da 1208) desteklenir:

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, 1147, 1148, 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 ve Linux Unicode desteği

Linux

Windows

IBM MQ for Windows ve IBM MQ for Linux üzerinde, aşağıdaki listede Unicode CCSID 'leri olmayan CCSID' ler için desteklenen Unicode CCSID ' leri (tercihen 1200 ya da 1208) desteklenir:

037.

277, 278, 280, 284, 285, 290, 297
300, 301
420, 424, 437
500
813, 819, 833, 835, 836, 837, 838, 850, 852, 855, 856, 857, 858, 860, 861, 862, 863, 864, 865, 866,
867, 868, 869, 870, 871, 874, 875, 878, 880, 891, 897
901, 902, 903, 904, 912, 913^{"5" sayfa 975}, 915, 916, 918, 920, 921, 922, 923, 924, 927, 928, 930,
931^{"1" sayfa 975}, 932^{"2" sayfa 975}, 933, 935, 937, 938^{"3" sayfa 975}, 939, 941, 942, 943, 947, 948, 949,
950, 951, 954^{"4" sayfa 975}, 964, 970
1006, 1025, 1026, 1027, 1040, 1041, 1042, 1043, 1046, 1047, 1051, 1088, 1089, 1097, 1098
1112, 1114, 1115, 1122, 1123, 1124, 1129, 1130, 1132, 1133, 1140, 1141, 1142, 1143, 1144,
1145, 1146, 1147, 1148, 1149, 1153, 1156, 1157
1200, 1208, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1275, 1280, 1281, 1282,
1283
1363, 1364, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381, 1383, 1386, 1388
4899
5050, 5346, 5347, 5348, 5349, 5350, 5351, 5352, 5353, 5354, 5488^{"5" sayfa 975}
9044, 9048, 9449
12712
13488
17584
33722^{"4" sayfa 975}

Notlar:

1. 931, dönüştürme için 939 kullanır.
2. 932, dönüştürme için 942 kullanır.
3. 938, dönüştürme için 948 kullanır.
4. 954 ve 33722, dönüştürme için 5050 'yi kullanır.
5. Yalnızca Windows ve Linux üzerinde.

Unicode için IBM i desteği



UNICODE desteğine ilişkin ayrıntılar için işletim sisteminizle ilgili uygun IBM i yayınına bakın.

Unicode için IBM MQ for z/OS desteği



IBM MQ for z/OS dönüşümünde, aşağıdaki listede Unicode CCSID 'leri olmayan CCSID' ler için desteklenen Unicode CCSID 'leri (tercihen 1200 ya da 1208) desteklenir:

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, 834, 835, 836, 837, 838, 848, 849, 850, 851, 852, 855, 856, 857, 858,
859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 874, 875, 878, 880, 891, 895,
896, 897
901, 902, 903, 904, 905, 912, 914, 915, 916, 918, 920, 921, 922, 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, 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, 1140,
1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1153, 1154, 1155, 1156, 1157, 1158,
1159, 1160, 1161, 1161, 1164
1200, 1208, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1275, 1276, 1277, 1280,
1281, 1282, 1283, 1284, 1285
1351, 1362, 1363, 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 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

64 bitlik platformlarda kodlama standartları

64 bitlik platformlarda ve tercih edilen veri tiplerinde kodlama standartları hakkında bilgi edinmek için bu bilgileri kullanın.

Tercih edilen veri tipleri

Bu tipler hiçbir zaman boyutu deęiřtirmmez ve hem 32 bit hem de 64 bit IBM MQ platformlarında kullanılabilir:

Çizelge 675. Veri tipi adları ve uzunlukları

Ad	Uzunluk
MQLONG	4 Byte
MQULONG	4 Byte
MQINT32	4 Byte
MQUINT32	4 Byte
MQINT64	8 Byte
MQUINT64	8 Byte

ALW AIX, Linux, and Windows üzerinde standart veri tipleri

32 bit AIX and Linux ve 64 bit AIX, Linux, and Windows uygulamalarındaki standart veri tipleri hakkında bilgi edinin.

32 bit AIX and Linux uygulamaları



Çizelge 676. 32 bit AIX and Linux uygulamaları için veri tipi adları ve uzunlukları

Ad	Uzunluk
DAMGA	1 Byte

Çizelge 676. 32 bit AIX and Linux uygulamaları için veri tipi adları ve uzunlukları (devamı var)

Ad	Uzunluk
kısa	2 Byte
int	4 Byte
uzun	4 Byte
kayan nokta	4 Byte
çift	8 Byte
uzun çift	8 Byte
işaretçi	4 Byte
ptrdiff_t	4 Byte
büyükük_t	4 Byte
saat_t	4 Byte
saat_t	4 Byte
wchar_t	4 Byte

AIX

AIX üzerinde bir wchar_t değerinin 2 bayt olduğunu unutmayın.

64 bit AIX and Linux uygulamaları

Linux

AIX

Çizelge 677. 64 bit AIX and Linux uygulamaları için veri tipi adları ve uzunlukları

Ad	Uzunluk
DAMGA	1 Byte
kısa	2 Byte
int	4 Byte
uzun	8 Byte
kayan nokta	4 Byte
çift	8 Byte
uzun çift	8 Byte
işaretçi	8 Byte
ptrdiff_t	8 Byte
büyükük_t	8 Byte
saat_t	8 Byte
saat_t	4 Byte
wchar_t	4 Byte

AIX

AIX üzerinde bir wchar_t değerinin 2 bayt olduğunu unutmayın.

Windows 64 bitlik uygulamalar

Windows

Çizelge 678. Windows 64 bit uygulamaları için veri tipi adları ve uzunlukları

Ad	Uzunluk
DAMGA	1 Byte
kısa	2 Byte
int	4 Byte
uzun	4 Byte
kayan nokta	4 Byte
çift	8 Byte
uzun çift	8 Byte
işaretçi	8 Byte
	Tüm göstergelerin 8 bayt olduğunu unutmayın.
ptrdiff_t	8 Byte
büyükük_t	8 Byte
saat_t	8 Byte
saat_t	4 Byte
wchar_t	2 Byte
Word	2 Byte
DWORD	4 Byte
Tnt	8 Byte
DOSYASI	4 Byte

Windows üzerinde kodlamala ilgili önemli noktalar

Windows

İŞLEYEN HF;

Kullanım

```
hf = CreateFile((LPCTSTR) FileName,  
               Access,  
               ShareMode,  
               xihSecAttsNTRestrict,  
               Create,  
               AttrAndFlags,  
               NULL);
```

Kullanma

```
HFILE hf;  
hf = (HFILE) CreateFile((LPCTSTR) FileName,  
                       Access,  
                       ShareMode,  
                       xihSecAttsNTRestrict,  
                       Create,  
                       AttrAndFlags,  
                       NULL);
```

bir hata oluşması gibi.

boyut_t uzrn fişler

Kullanım

```
size_t len
while (fgets(string1, (int) len, fp) != NULL)
len = strlen(buffer);
```

Kullanma

```
int len;
while (fgets(string1, len, fp) != NULL)
len = strlen(buffer);
```

yazıcı

Kullanım

```
printf("My struc pointer: %p", pMyStruc);
```

Kullanma

```
printf("My struc pointer: %x", pMyStruc);
```

Onaltılı çıkışa gereksinim duyarsanız, üst ve alt 4 baytı ayrı olarak yazdırmanız gerekir.

char * ptr

Kullanım

```
char * ptr1;
char * ptr2;
size_t bufLen;

bufLen = ptr2 - ptr1;
```

Kullanma

```
char *ptr1;
char *ptr2;
UINT32 bufLen;

bufLen = ptr2 - ptr1;
```

alignBytes

Kullanım

```
alignBytes = (unsigned short) ((size_t) address % 16);
```

Kullanma

```
void *address;
unsigned short alignBytes;

alignBytes = (unsigned short) ((UINT32) address % 16);
```

UZUNLUK

Kullanım

```
len = (UINT32) ((char *) address2 - (char *) address1);
```

Kullanma

```
void *address1;
void *address2;
UINT32 len;

len = (UINT32) ((char *) address2 - (char *) address1);
```

sscanf

Kullanım

```
MQLONG SBCSprt;

sscanf(line, "%d", &SBCSprt);
```

Kullanma

```
MQLONG SBCSprt;

sscanf(line, "%1d", &SBCSprt);
```

%1d , 8 baytlık bir tipi 4 baytlık bir tipe yerleştirmeyi dener; yalnızca gerçek bir long veri tipiyle çalışıyorsanız %1 kullanın. MQLONG, UINT32 ve INT32 dört byte olarak tanımlanır; bu, tüm IBM MQ altyapılarında int ile aynıdır:

IBM i

IBM i Uygulama Programlama Başvurusu (ILE/RPG)

IBM için uygulama programlama.

IBM için uygulamalar geliştirmenize yardımcı olması için bu bilgileri kullanın.

- [“IBM i üzerindeki veri tipi açıklamaları” sayfa 981](#)
- [“IBM i üzerinde işlev çağrıları” sayfa 1230](#)
- [“IBM i üzerindeki nesnelerin öznitelikleri” sayfa 1344](#)
- [“Uygulamalar” sayfa 1390](#)
- [“IBM i \(ILE RPG\) için dönüş kodları” sayfa 1402](#)
- [“IBM i \(ILE RPG\) için MQI seçeneklerini doğrulama kuralları” sayfa 1403](#)
- [“IBM i üzerinde makine kodlamaları” sayfa 1406](#)
- [“IBM i üzerinde rapor seçenekleri ve ileti işaretleri” sayfa 1409](#)

IBM i üzerinde RPG ve COBOL uygulamaları için uyumluluk kipinin kullanımdan kaldırılması

IBM i

IBM MQ for IBM i 9.0' den ürün, uyumluluk kipi olarak bilinen dinamik bağlantıyı kullanan RPG ya da COBOL uygulamaları için artık destek sağlamaz. Bu işletim kipi, MQSeries 5.1' den önce yazılan uygulamalar için gerekliydi ve ürünün sonraki sürümleri, bunları derlemek için gereken kopya defterleri IBM WebSphere MQ 6.0'içinde kaldırılmış olsa da, bu uygulamalar için uyumlu bir çalıştırma zamanı ortamı sağladı. Dinamik bağlantı (uyumluluk kipi), IBM MQ for IBM i 9.0adresindeki QMQM kitaplığında kaldırılan aşağıdaki programlar tarafından sağlanmıştır:

- AMQVSTUB
- AMQZSTUB
- QMQM
- MQCLOSE
- MQCONN
- MQDISC

- MQGet
- MQINQ
- MQOPEN
- MQPUT
- MQPUT1
- MQSET

IBM MQ for IBM i 9.0' den, bu uyumluluk kipini kullanan uygulamaların LIBMQM ve LIBMQM_R hizmet programları tarafından sağlanan statik bağlı MQ çağrılarını kullanmak için yeniden derlenmesi gerekir. AMQ3PUT4 ve AMQ3GET4 gibi örnek programlar, bu programlama modelini nasıl kullanacağını gösterir. Bu MQ çağrılarının kullanılmasıyla ilgili ek bilgi için bkz. [IBM i Application Programming Reference \(ILE/RPG\)](#).

Notlar:

- LIBMQM hizmet programını kullanmak için, şu anda CALL 'QMQM' arabirimini kullanarak uygulamaları yeniden kodlemeniz gerekir.

Önceki listedeki program nesnelere ve hizmet programları (örneğin, QMQM, MQCONN, MQPUT, AMQVSTUB ve AMQZSTUB) IBM MQ for IBM i 9.0 içinde kaldırılır ve uyumluluk kipini kullanacak şekilde kodlanmış uygulamalar çalışmayı durdurdu.

- Uygulamalar IBM MQ for IBM i 8.0 adresindeki LIBMQM hizmet programına bağlıysa, bu uygulamaları IBM MQ for IBM i 9.0 ya da daha sonra yeniden derlemeniz ya da yeniden bağlamanız gerekmez.
- Aynı bölüme birden çok IBM MQ for IBM i sürümü kurulamaz.

RPG ya da COBOL programınızın uyumluluk kipini kullanıp kullanmadığını öğrenmek için, uygulama programı tarafından çağrılan dış programları görüntülemek üzere **DSPPGMREF** (Program Başvurularını Görüntüle) komutunu kullanın. Bu bölümde listelenen programlara ilişkin göndermeler varsa, program IBM MQ for IBM i 9.0 ya da daha sonraki bir yayın düzeyiyle çalışmaz. Aşağıdaki **DSPPGMREF** çıkış örneği, kullanımdan kaldırılan üç program nesnesini göstermektedir: MQCONN, MQOPEN, 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

```

Bu tür programlar, [IBM i ' da COBOL programlarını hazırlama](#) içinde açıklanan Bound Procedural Call yöntemi kullanılarak yeniden derlenmelidir.

Uyumluluk kipini kullanan bir uygulama programını IBM MQ for IBM i 9.0 ya da daha sonra çalıştırmayı denerseniz, en sık görülen ilk hata, MQCONN ya da QMQM programını çağırmaya çalışan bir MCH3401 hatasıdır.

İlgili görevler

[Uygulama geliştirilmesi](#)

IBM i üzerindeki veri tipi açıklamaları

Bu konu derlemi, IBM i programlamasında kullanılan veri tiplerinin açıklamalarını içerir.

Veri tipleri tanımında kullanılan kurallar

Bu bilgiler, her temel veri tipi için, programlama dilinden bağımsız bir biçimde, kullanımıyla ilgili bir açıklama sağlar. Bunu, RPG programlama dilinin ILE sürümündeki tipik bildirimler izler. Tutarlılık

sağlamak için temel veri tiplerinin tanımları buraya eklenmiştir. RPG, gereksinim duyduğunuz öznitelikler kullanılarak çalışma alanlarının bildirilebileceği D' belirtimlerini kullanır. Ancak, bunu alanın kullanıldığı hesaplama belirtimlerinde yapabilirsiniz.

Temel veri tiplerini kullanmak için aşağıdaki işlemleri yaratırsınız:

- Tüm veri tiplerini içeren bir /COPY üyesi ya da
- Tüm veri tiplerini içeren bir dış veri yapısı (PF). Daha sonra, çalışma alanlarınızı uygun veri tipi alanı olan 'LIKE' öznitelikleriyle belirtmeniz gerekir.

İkinci seçeneğin yararları, tanımların diğer IBM nesnelere için 'FIELD REFERENCE FILE' olarak kullanılabilirliği. Bir IBM MQ veri tipi tanımlaması değişirse, bu nesnelere yeniden yaratmak nispeten basit bir işlemdir.

Temel veri tipleri

Bu bölümde açıklanan diğer tüm veri tipleri, doğrudan bu temel veri tiplerine ya da bu temel veri türlerinin (diziler ya da yapılar) toplamalarına eşittir.

<i>Çizelge 679. Temel veri tipleri</i>	
Veri tipi	Gösterim
MQBOOL	10 basamaklı işaretli tamsayı
MQBYTE	1 baytlık alfasayısal alan
MQBYTE16	16 baytlık alfasayısal alan
MQBYTE24	24 baytlık alfasayısal alan
MQBYTE32	32 baytlık alfasayısal alan
MQBYTE64	64 baytlık alfasayısal alan
MQCHAR	1 baytlık alfasayısal alan
MQCHAR4	4 baytlık alfasayısal alan
MQCHAR8	8 baytlık alfasayısal alan
MQCHAR12	12 baytlık alfasayısal alan
MQCHAR16	16 baytlık alfasayısal alan
MQCHAR20	20 baytlık alfasayısal alan
MQCHAR28	28 baytlık alfasayısal alan
MQCHAR32	32 baytlık alfasayısal alan
MQCHAR48	48 baytlık alfasayısal alan
MQCHAR64	64 baytlık alfasayısal alan
MQCHAR128	128 baytlık alfasayısal alan
MQCHAR256	256 baytlık alfasayısal alan
MQFLOAT32	4 baytlık kayan noktalı sayı
MQFLOAT64	8 baytlık kayan noktalı sayı
MQHCONFIG	Yapılandırma tanıtıcısı
MQHCONN	10 basamaklı işaretli tamsayı
MQHMSG	Bir iletiye erişim veren ileti tanıtıcısı

Çizelge 679. Temel veri tipleri (devamı var)

Veri tipi	Gösterim
MQHOBj	10 basamaklı işaretli tamsayı
MQINT8	8 bitlik işaretli tamsayı
MQINT16	16 bitlik işaretli tamsayı
MQINT32	32 bit işaretli tamsayı
MQINT64	64 bit işaretli tamsayı
MQLONG	32 bit işaretli tamsayı
MQPID	Süreç tanıtıcısı
MQPTR	Pointer
MQTID	İş parçacığı tanıtıcısı
MQUINT8	8 bitlik işaretsiz tamsayı
MQUINT16	16 bit işaretsiz tamsayı
MQUINT32	32 bit işaretsiz tamsayı
MQUINT64	64 bit işaretsiz tamsayı
MQULONG	32 bit işaretsiz tamsayı
PMQACH	MQACH tipindeki bir veri yapısına ilişkin gösterge
PMQAIR	MQAIR tipindeki bir veri yapısına ilişkin gösterge
PMQAXC	MQAXC tipindeki bir veri yapısına ilişkin gösterge
PMAXP (MAXP)	MAXP tipindeki bir veri yapısına ilişkin gösterge
PMQBMHO	MQBMHO tipindeki bir veri yapısına ilişkin gösterge
PMQBO	MQBO tipindeki bir veri yapısına ilişkin gösterge
PMQBOOL	MQBOOL tipindeki verilere ilişkin gösterge
PMQBYTE	MQBYTE tipindeki verilere ilişkin gösterge
PMQBYTE _n	MQBYTE _n tipindeki verilere ilişkin gösterge
PMQCBC	MQCBC tipindeki bir veri yapısına ilişkin gösterge
PMQCBD	MQCBD tipindeki bir veri yapısına ilişkin gösterge
PMQCHAR	MQCHAR tipindeki bir veri yapısına ilişkin gösterge
PMQCHARV	MQCHARV tipindeki bir veri yapısına ilişkin gösterge
PMQCHARN	MQCHAR _n tipindeki verilere ilişkin gösterge
PMQCIH	MQCIH tipindeki bir veri yapısına ilişkin gösterge
PMQCMHO	MQCMHO tipindeki bir veri yapısına ilişkin gösterge
PMQCNO	MQCNO tipindeki bir veri yapısına ilişkin gösterge
PMQCSP	MQCSP tipindeki bir veri yapısına ilişkin gösterge
PMQCTLO	MQCTLO tipindeki bir veri yapısına ilişkin gösterge
PMQDH	MQDH tipindeki bir veri yapısına ilişkin gösterge

Çizelge 679. Temel veri tipleri (devamı var)

Veri tipi	Gösterim
PMQDHO	MQDHO tipindeki bir veri yapısına ilişkin gösterge
PMQDLH	MQDLH tipindeki bir veri yapısına ilişkin gösterge
PMQDMHO	MQDMHO tipindeki bir veri yapısına ilişkin gösterge
PMQDMPO	MQDMPO tipindeki bir veri yapısına ilişkin gösterge
PMQEPH	MQEPH tipindeki bir veri yapısına ilişkin gösterge
PMQFLOAT32	MQFLOAT32 tipindeki verilere ilişkin gösterge
PMQFLOAT64	MQFLOAT64 tipindeki verilere ilişkin gösterge
PMQFUNC	Bir işleve ilişkin gösterge
PMQGMO	MQGMO tipindeki bir veri yapısına ilişkin gösterge
PMQHCONFIG	MQHCONFIG tipindeki verilere ilişkin gösterge
PMQHCONN	MQHCONN tipindeki verilere ilişkin gösterge
PMQHMSG	MQHMSG tipindeki verileri gösteren gösterge
PMQHOBJ	MQHOBJ tipindeki verilere ilişkin gösterge
PMQIIH	MQIIH tipindeki bir veri yapısına ilişkin gösterge
PMQIMPO	MQIMPO tipindeki bir veri yapısına ilişkin gösterge
PMQINT8	MQINT8 tipindeki verileri gösteren gösterge
PMQINT16	MQINT16 tipindeki verileri gösteren gösterge
PMQINT32	MQINT32 tipindeki verilere ilişkin gösterge
PMQINT64	MQINT64 veri tipine ilişkin gösterge
PMQLONG	MQLONG tipindeki verilere ilişkin gösterge
PMQMD	MQMD tipindeki bir veri yapısına ilişkin gösterge
PMQMDE	MQMDE tipindeki bir veri yapısına ilişkin gösterge
PMQMD1	MQMD1 tipindeki bir veri yapısına ilişkin gösterge
PMQMD2	MQMD2 tipindeki bir veri yapısına ilişkin gösterge
PMQMHBO	MQMHBO tipindeki bir veri yapısına ilişkin gösterge
PMQOD	MQOD tipindeki bir veri yapısına ilişkin gösterge
PMQOR	MQOR tipindeki bir veri yapısına ilişkin gösterge
PMQPD	MQPD tipindeki bir veri yapısına ilişkin gösterge
PMQPID	Bir süreç tanıtıcısı MQPID göstergesi
PMQPMO	MQPMO tipindeki bir veri yapısına ilişkin gösterge
PMQPTR	MQPTR tipindeki verilere ilişkin gösterge
PMQRFH	MQRFH tipindeki bir veri yapısına ilişkin gösterge
PMQRFH2	MQRFH2 tipindeki bir veri yapısına ilişkin gösterge
PMQRMH	MQRMH tipindeki bir veri yapısına ilişkin gösterge

Çizelge 679. Temel veri tipleri (devamı var)

Veri tipi	Gösterim
PMQRR	MQRR tipindeki bir veri yapısına ilişkin gösterge
PMQSCO	MQSCO tipindeki bir veri yapısına ilişkin gösterge
PMQSD	MQSD tipindeki bir veri yapısına ilişkin gösterge
PMQSMPO	MQSMPO tipindeki bir veri yapısına ilişkin gösterge
PMQSRO	MQSRO tipindeki bir veri yapısına ilişkin gösterge
PMQSTS	MQSTS tipindeki bir veri yapısına ilişkin gösterge
PMQTID	Bir iş parçacığı tanıtıcısı MQTID ' ye ilişkin gösterge
PMQTM	MQTM tipindeki bir veri yapısına ilişkin gösterge
PMQTM2	MQTM2 tipindeki bir veri yapısına ilişkin gösterge
PMQUINT8	MQUINT8 veri tipine ilişkin gösterge
PMQUINT16	MQUINT16 veri tipine ilişkin gösterge
PMQUINT32	MQUINT32 tipindeki verileri gösteren gösterge
PMQUINT64	MQUINT64 veri tipine ilişkin gösterge
PMQULONG	MQULONG tipindeki verilere ilişkin gösterge
PMQVOID	Pointer
PMQWIH	MQWIH tipindeki bir veri yapısına ilişkin gösterge
PMQXQH	MQXQH tipindeki bir veri yapısına ilişkin gösterge

IBM i **IBM i üzerinde MQBOOL**

MQBOOL veri tipi bir Boole değerini gösterir. 0 değeri false değerini gösterir. Diğer herhangi bir değer true değerini temsil eder.

Bir MQBOOL, MQLONG veri tipine göre hizalanmalıdır.

IBM i **IBM i üzerinde MQBYTE**

MQBYTE veri tipi, tek byte veri tipini gösterir.

Bayta belirli bir yorum konmaz-ikili sayı ya da karakter olarak değil, bir bit dizgisi olarak işlenir. Özel hizalama gerekmez.

MQBYTE dizisi bazen, kuyruk yöneticisinin bilmediği bir nitelikte ana saklama alanını göstermek için kullanılır. Örneğin, alan uygulama iletisi verilerini ya da bir yapıyı içerebilir. Bu alanın sınır hizalaması, içindeki verilerin niteliğiyle uyumlu olmalıdır.

IBM i **IBM i üzerinde MQBYTEn (n byte dizgisi)**

Her MQBYTEn veri tipi n byte 'lık bir dizgiyi temsil eder.

Burada n aşağıdaki değerlerden birini alabilir:

- 16, 24, 32 ya da 64.

Her byte, MQBYTE veri tipi tarafından tanımlanır. Özel hizalama gerekmez.

Dizedeki veriler, dizginin tanımlı uzunluğundan kısaysa, dizgiyi doldurmak için veriler boş karakterlerle doldurulmalıdır.

Kuyruk yöneticisi uygulamaya byte dizgileri döndürdüğünde (örneğin, MQGET çağrısında), kuyruk yöneticisi her zaman dizginin tanımlı uzunluğuna boş değerler ile doldurur.

Bayt dizilimi alanlarının uzunluklarını tanımlayan sabitler vardır.

IBM i IBM i üzerinde MQCHAR (karakter)

MQCHAR veri tipi tek bir karakteri temsil eder.

Karakterin kodlanmış karakter takımı tanıtıcısı kuyruk yöneticisinin tanıtıcısıdır ([CodedCharSetId](#) konusundaki **CodedCharSetId** özniteliğine bakın). Özel hizalama gerekmez.

Not: MQGET, MQPUT ve MQPUT1 çağrılarında belirtilen uygulama iletisi verileri, MQCHAR veri tipi değil, MQBYTE veri tipi tarafından tanımlanır.

IBM i IBM i üzerinde MQCHARn (n karakterlik dizgi)

Her MQCHARn veri tipi *n* karakterlik bir dizgiyi temsil eder.

Burada *n* aşağıdaki değerlerden birini alabilir:

- 4, 8, 12, 16, 20, 28, 32, 48, 64, 128 ya da 256

Her karakter MQCHAR veri tipi tarafından tanımlanır. Özel hizalama gerekmez.

Dizilimdeki veriler, dizilimin tanımlı uzunluğundan kısaysa, dizilimi doldurmak için veriler boşluklarla doldurulmalıdır. Bazı durumlarda, dizgiyi boşluklarla doldurmak yerine, zamanından önce sonlandırmak için boş karakter kullanılabilir; boş karakter ve onu izleyen karakterler, dizginin tanımlı uzunluğuna kadar boşluk olarak değerlendirilir. Boş değer kullanılabileceği yerler, çağrı ve veri tipi açıklamalarında tanımlanır.

Kuyruk yöneticisi uygulamaya karakter dizgileri döndürdüğünde (örneğin, MQGET çağrısında), kuyruk yöneticisi her zaman dizginin tanımlı uzunluğuna kadar boşluklarla doldurur; kuyruk yöneticisi dizgiyi sınırlamak için boş karakteri kullanmaz.

Karakter dizilimi alanlarının uzunluklarını tanımlayan sabitler vardır.

IBM i IBM i üzerinde MQFLOAT32

MQFLOAT32 veri tipi, standart IEEE kayan nokta biçimi kullanılarak gösterilen 32 bitlik bir kayan noktalı sayıdır.

MQFLOAT32 , 4 baytlık bir sınırdan hizalanmalıdır.

IBM i IBM i üzerinde MQFLOAT64

MQFLOAT64 veri tipi, standart IEEE kayan nokta biçimi kullanılarak gösterilen 64 bitlik bir kayan noktalı sayıdır.

MQFLOAT64 8 baytlık bir sınırdan hizalanmalıdır.

MQHCONFIG-yapılanış tanıtıcısı

MQHCONFIG veri tipi, belirli bir kurulabilir hizmet için yapılandırılmakta olan bir yapılanış tanıtıcısını gösterir. Bir yapılandırma tanıtıcısı, doğal sınırında hizalanmalıdır.

Not: Uygulamalar, yalnızca eşitlik için bu tipteki değişkenleri test etmek zorundadır.

IBM i IBM i üzerinde MQHCONN (Bağlantı tanıtıcısı)

MQHCONN veri tipi, belirli bir kuyruk yöneticisine yönelik bağlantı tanıtıcısını gösterir.

Bir bağlantı tanıtıcısı, doğal sınırında hizalanmalıdır.

Not: Uygulamalar, yalnızca eşitlik için bu tipteki değişkenleri test etmek zorundadır.

IBM i IBM i üzerinde MQHMSG (İleti tanıtıcısı)

MQHMSG veri tipi, bir iletiye erişim veren ileti tanıtıcısını gösterir.

İleti tanıtıcısı 8 baytlık bir sınırdaki hizalanmalıdır.

Not: Uygulamalar, yalnızca eşitlik için bu tipteki değişkenleri test etmek zorundadır.

IBM i M̄HOBJ (Nesne tanıtıcısı)

M̄HOBJ veri tipi, bir nesneye erişim veren bir nesne tanıtıcısını gösterir.

Bir nesne tanıtıcısı, doğal sınırında hizalanmalıdır.

Not: Uygulamalar, yalnızca eşitlik için bu tipteki değişkenleri test etmek zorundadır.

IBM i M̄INT8 (8 bit işaretli tamsayı)- IBM i

M̄INT8 veri tipi, bağlam tarafından tersi belirtilmedikçe, -128-+127 aralığındaki herhangi bir değeri alabilen 8 bitlik imzalı bir tamsayıdır.

IBM i M̄INT16 (16 bit işaretli tamsayı)- IBM i

M̄INT16 veri tipi, bağlam tarafından tersi belirtilmedikçe, -32 768-+32 767 aralığındaki herhangi bir değeri alabilen 16 bitlik imzalı bir tamsayıdır.

M̄INT16 , 2 baytlık bir sınırdaki hizalanmalıdır.

IBM i M̄INT32 (32 bit tamsayı) açık IBM i

M̄INT32 veri tipi 32 bitlik imzalı bir tamsayıdır.

M̄QLONG ile eşdeğerdir.

IBM i M̄INT64 (64 bitlik tamsayı)- IBM i

M̄INT64 veri tipi, -9 223 372 036 854 775 808-+ 9 223 372 036 854 775 807 aralığındaki herhangi bir değeri alabilen 64 bitlik imzalı bir tamsayıdır.

COBOL için geçerli değer aralığı, -999 999 999 999 999 ile 999 999 999 arasında bir değerle sınırlıdır. M̄INT64 8 baytlık bir sınırdaki hizalanmalıdır.

IBM i IBM i üzerinde M̄QLONG (Uzun tamsayı)

M̄QLONG veri tipi, bağlamla sınırlanmadığı sürece, doğal sınırıyla hizalanmadığı sürece, -2 147 483 648-+ 2 147 483 647 aralığındaki herhangi bir değeri alabilen 32 bitlik imzalı bir ikili tamsayıdır.

M̄PID-süreç tanıtıcısı

IBM MQ işlem tanıtıcısı.

Bu, IBM MQ izleme ve FFST dökümlerinde kullanılan tanıtıcıyla aynıdır, ancak işletim sistemi işlem tanıtıcısından farklı olabilir.

M̄PTR göstergesi

M̄PTR veri tipi, herhangi bir tipteki verilerin adresidir. Bir işaretçinin doğal sınırıyla hizalanması gerekir; bu, IBM üzerinde 16 baytlık bir sınırdaki.

Bazı programlama dilleri tip atanmış göstergeleri destekler; MQI bunları birkaç durumda da kullanır.

M̄TID-iş parçacığı tanıtıcısı

MQ iş parçacığı tanıtıcısı.

Bu, MQ izleme ve FFST dökümlerinde kullanılan tanıtıcıyla aynıdır, ancak işletim sistemi iş parçacığı tanıtıcısından farklı olabilir.

IBM i MQUINT8 (8 bit işaretli tamsayı) IBM i

MQUINT8 veri tipi, bağlam tarafından tersi belirtilmedikçe, 0 ile +255 aralığındaki herhangi bir değeri alabilen, 8 bitlik işaretli bir tamsayıdır.

MQUINT16 -16 bitlik işaretli tamsayı

MQUINT16 veri tipi, bağlam tarafından tersi belirtilmedikçe, 0-+65 535 aralığındaki herhangi bir değeri alabilen, 16 bitlik işaretli bir tamsayıdır.

MQUINT16 , 2 baytlık bir sınırdan hizalanmalıdır.

IBM i MQUINT32 (32 bit işaretli tamsayı) IBM i

MQUINT32 veri tipi, 32 bit işaretli bir tamsayıdır. MQULONG ile eşdeğerdir.

MQUINT64 -64 bit işaretli tamsayı

MQUINT64 veri tipi, bağlam tarafından tersi belirtilmedikçe, 0 ile +18 446 744 073 709 551 615 aralığındaki herhangi bir değeri alabilen 64 bitlik bir işaretli tamsayıdır.

COBOL için, geçerli aralık 0 ile +999 999 999 999 999 arasında bir değerle sınırlıdır. MQUINT64 , 8 baytlık bir sınırdan hizalanmalıdır.

MQULONG-32 bit işaretli tamsayı

MQULONG veri tipi, bağlamla kısıtlanmadıkça, 0-+ 4 294 967 294 aralığındaki herhangi bir değeri alabilen 32 bitlik imzalanmamış bir ikili tamsayıdır.

MQULONG, 4 baytlık bir sınırdan hizalanmalıdır.

PMQACH-MQACH tipindeki bir veri yapısına ilişkin gösterge

MQACH tipindeki bir veri yapısına ilişkin gösterge.

PMQAIR-MQAIR tipindeki bir veri yapısına ilişkin gösterge

MQAIR tipindeki bir veri yapısına ilişkin gösterge.

PMQAXC-MQAXC tipindeki bir veri yapısına ilişkin gösterge

MQAXC tipindeki bir veri yapısına ilişkin gösterge.

PMQAXP-MQAXP tipindeki bir veri yapısına ilişkin gösterge

MQAXP tipindeki bir veri yapısına ilişkin gösterge.

PMQBMHO-MQBMHO tipindeki bir veri yapısına ilişkin gösterge

MQBMHO tipindeki bir veri yapısına ilişkin gösterge.

PMQBO-MQBO tipindeki bir veri yapısına ilişkin gösterge

MQBO tipindeki bir veri yapısına ilişkin gösterge.

PMQBOOL-MQBOOL tipindeki verilere ilişkin gösterge

MQBOOL tipindeki verilere ilişkin gösterge.

MQBOOL tipindeki verilere ilişkin gösterge.

PMQBYTE-MQBYTE veri tipine ilişkin gösterge

MQBYTE veri tipine ilişkin gösterge.

PMQBYTEN-MQBYTEN tipinde bir veri yapısına ilişkin gösterge

MQBYTEN tipindeki bir veri yapısına ilişkin gösterge; burada n, 8, 12, 16, 24, 32, 40, 48 ya da 128 olabilir.

PMQCBC-MQCBC tipindeki bir veri yapısına ilişkin gösterge

MQCBC tipindeki bir veri yapısına ilişkin gösterge.

PMQCBD-MQCBD tipindeki bir veri yapısına ilişkin gösterge

MQCBD tipindeki bir veri yapısına ilişkin gösterge.

PMQCHAR-MQCHAR tipindeki verilere ilişkin gösterge

MQCHAR tipindeki verilere ilişkin gösterge.

PMQCHARV-MQCHARV tipindeki bir veri yapısına ilişkin gösterge

MQCHARV tipindeki bir veri yapısına ilişkin gösterge.

PMQCHARn-MQCHARn veri tipine ilişkin gösterge

n, 4, 8, 12, 20, 28, 32, 64, 128, 256, 264 olabilir.

PMQCIH-MQCIH tipinde bir veri yapısına ilişkin gösterge

MQCIH tipindeki bir veri yapısına ilişkin gösterge.

PMQCMHO-MQCMHO tipindeki bir veri yapısına ilişkin gösterge

MQCMHO tipindeki bir veri yapısına ilişkin gösterge.

PMQCNO-MQCNO tipindeki bir veri yapısına ilişkin gösterge

MQCNO tipindeki bir veri yapısına ilişkin gösterge.

PMQCSP-MQCSP tipindeki bir veri yapısına ilişkin gösterge

MQCSP tipindeki bir veri yapısına ilişkin gösterge.

PMQCTLO-MQCTLO tipindeki bir veri yapısına ilişkin gösterge

MQCTLO tipindeki bir veri yapısına ilişkin gösterge.

PMQDH-MQDH tipindeki bir veri yapısına ilişkin gösterge
MQDH tipindeki bir veri yapısına ilişkin gösterge.

PMQDHO-MQDHO tipinde bir veri yapısına ilişkin gösterge
MQDHO tipindeki bir veri yapısına ilişkin gösterge.

PMQDLH-MQDLH tipinde bir veri yapısına ilişkin gösterge
MQDLH tipinde bir veri yapısına ilişkin gösterge.

PMQDMHO-MQDMHO tipindeki bir veri yapısına ilişkin gösterge
MQDMHO tipindeki bir veri yapısına ilişkin gösterge.

PMQDMPO-MQDMPO tipindeki bir veri yapısına ilişkin gösterge
MQDMPO tipindeki bir veri yapısına ilişkin gösterge.
MQDMPO tipindeki bir veri yapısına ilişkin gösterge.

PMQEPH-MQEPH tipindeki bir veri yapısına ilişkin gösterge
MQEPH tipindeki bir veri yapısına ilişkin gösterge.

PMQFLOAT32 - MQFLOAT32 tipindeki verileri gösteren gösterge
MQFLOAT32tipindeki verilere ilişkin gösterge.

PMQFLOAT64 - MQFLOAT64 veri tipine ilişkin gösterge
MQFLOAT64tipindeki verilere ilişkin gösterge.

PMQFUNC-bir işleve ilişkin gösterge
Bir işleve ilişkin gösterge.

PMQGMO-MQGMO tipindeki bir veri yapısına ilişkin gösterge
MQGMO tipindeki bir veri yapısına ilişkin gösterge.

PMQHCONFIG-MQHCONFIG veri tipine ilişkin gösterge
MQHCONFIG veri tipine ilişkin gösterge.

PMQHCONN-MQHCONN veri tipine ilişkin gösterge
MQHCONN veri tipine ilişkin gösterge.

PMQHMSG-MQHMSG veri tipine ilişkin gösterge

MQHMSG veri tipine ilişkin gösterge.

PMQHOBJ-MQHOBJ tipindeki verilere ilişkin gösterge

MQSMPO tipindeki verilere ilişkin gösterge.

PMQIIH-MQIIH tipindeki bir veri yapısına ilişkin gösterge

MQIIH tipindeki bir veri yapısına ilişkin gösterge.

PMQIMPO-MQIMPO tipindeki bir veri yapısına ilişkin gösterge

MQIMPO tipindeki bir veri yapısına ilişkin gösterge.

PMQINT8 - MQINT8 veri tipi göstergesi

MQINT8tipindeki verilere ilişkin gösterge.

PMQINT16 - MQINT16 tipinde veri göstergesi

MQINT16tipindeki verilere ilişkin gösterge.

IBM i PMQINT32 (IBM i üzerinde MQINT32tipindeki veriler için İşaretçi)

PMQINT32 veri tipi, MQINT32veri tipine ilişkin bir işaretçidir. Bu, PMQLONG ile eşdeğerdir.

IBM i PMQINT64 (IBM i üzerinde MQINT64tipinde veriler için İşaretçi)

PMQINT64 veri tipi, MQINT64veri tipine ilişkin bir işaretçidir.

PMQLONG-MQLONG tipindeki verilere ilişkin gösterge

MQLONG tipindeki verilere ilişkin gösterge.

PMQMD-MQMD tipindeki yapıya ilişkin gösterge

MQMD tipi yapıya ilişkin gösterge.

PMQMDE-MQMDE tipindeki bir veri yapısına ilişkin gösterge

MQMDE tipindeki bir veri yapısına ilişkin gösterge.

PMQMDE-MQMDE tipindeki bir veri yapısına ilişkin gösterge

MQMDI tipindeki bir veri yapısına ilişkin gösterge.

PMQMD2 - MQMD2 tipindeki bir veri yapısına ilişkin gösterge

MQMD2 tipindeki bir veri yapısına ilişkin gösterge

PMQMHBO-MQMHBO tipinde bir veri yapısına ilişkin gösterge
MQMHBO tipindeki bir veri yapısına ilişkin gösterge.

PMQOD-MQOD tipindeki bir veri yapısına ilişkin gösterge
MQOD tipindeki bir veri yapısına ilişkin gösterge.

PMQOR-MQOR tipindeki bir veri yapısına ilişkin gösterge
MQOR tipindeki bir veri yapısına ilişkin gösterge.

PMQPD-MQPD tipindeki bir veri yapısına ilişkin gösterge
MQPD tipindeki bir veri yapısına ilişkin gösterge.

PMQPID-bir süreç tanıtıcısına ilişkin gösterge
Bir işlem tanıtıcısına ilişkin gösterge.

PMQPMO-MQPMO tipindeki bir veri yapısına ilişkin gösterge
MQPMO tipindeki bir veri yapısına ilişkin gösterge.

PMQPTR-MQPTR tipindeki verilere ilişkin gösterge
MQPTR tipindeki verilere ilişkin gösterge.

PMQRFH-MQRFH tipindeki bir veri yapısına ilişkin gösterge
MQRFH tipindeki bir veri yapısına ilişkin gösterge.

PMQRFH2 - MQRFH2 tipindeki bir veri yapısına ilişkin gösterge
MQRFH2 tipindeki bir veri yapısına ilişkin gösterge.

PMQRMH-MQRMH tipindeki bir veri yapısına ilişkin gösterge
MQRMH tipindeki bir veri yapısına ilişkin gösterge.

PMQRR-MQRR tipindeki bir veri yapısına ilişkin gösterge
MQRR tipindeki bir veri yapısına ilişkin gösterge.

PMQSCO-MQSCO tipindeki bir veri yapısına ilişkin gösterge
MQSCO tipindeki bir veri yapısına ilişkin gösterge.

PMQSD-MQSD tipindeki bir veri yapısına ilişkin gösterge

MQSD tipindeki bir veri yapısına ilişkin gösterge.

PMQSMPO-MQSMPO tipindeki bir veri yapısına ilişkin gösterge

MQSMPO tipinde bir veri yapısına ilişkin gösterge.

PMQSRO-MQSRO tipindeki bir veri yapısına ilişkin gösterge

MQSRO tipindeki bir veri yapısına ilişkin gösterge.

PMQSTS-MQSTS tipindeki bir veri yapısına ilişkin gösterge

MQSTS tipindeki bir veri yapısına ilişkin gösterge.

PMQTID-MQTID tipindeki bir veri yapısına ilişkin gösterge

MQTID tipindeki bir veri yapısına ilişkin gösterge.

PMQTM-MQTM tipindeki bir veri yapısına ilişkin gösterge

MQTM tipindeki bir veri yapısına ilişkin gösterge.

PMQTM2 - MQTM2 tipindeki bir veri yapısına ilişkin gösterge

MQTM2tipindeki bir veri yapısına ilişkin gösterge.

PMQUINT8 - MQUINT8 veri tipine ilişkin gösterge

MQUINT8tipindeki verilere ilişkin gösterge.

PMQUINT16 - MQUINT16 veri tipi göstergesi

MQUINT16tipindeki verilere ilişkin gösterge.

IBM i PMQUINT32 (IBM i üzerinde MQUINT32tipindeki veriler için İşaretçi)

PMQUINT32 veri tipi, MQUINT32veri tipinde bir işaretçidir. Bu, PMQULONG ile eşdeğerdir.

IBM i PMQUINT64 (IBM i üzerinde MQUINT64tipindeki veriler için İşaretçi)

PMQUINT64 veri tipi, MQUINT64veri tipine ilişkin bir işaretçidir.

PMQULONG-MQULONG tipindeki verilere ilişkin gösterge

MQULONG tipindeki verilere ilişkin gösterge.

PMQVOID-gösterge

Bir işaretçi.

PMQWIH-MQWIH tipindeki bir veri yapısına ilişkin gösterge

MQWIH tipindeki bir veri yapısına ilişkin gösterge.

PMQXQH-MQXQH tipindeki bir veri yapısına ilişkin gösterge

MQXQH tipindeki bir veri yapısına ilişkin gösterge.

Dille ilgili dikkat edilmesi gereken noktalar

Bu konu, RPG programlama dilindeki MQI ' ı kullanmanıza yardımcı olacak bilgileri içerir.

Bu dille ilgili dikkat edilecek noktalardan bazıları şunlardır:

- [“Dosyaları kopyala” sayfa 994](#)
- [“Çağrılar” sayfa 996](#)
- [“Çağrı parametreleri” sayfa 996](#)
- [“Yapılar” sayfa 996](#)
- [“Adlandırılmış sabitler” sayfa 996](#)
- [“MQI yordamları” sayfa 997](#)
- [“İş threading ile ilgili dikkat edilecek noktalar” sayfa 997](#)
- [“Kesinleştirme denetimi” sayfa 997](#)
- [“Bağlı çağrılar kodlanması” sayfa 997](#)
- [“Notasyonel kurallar” sayfa 999](#)

Dosyaları kopyala

İleti kuyruklama kullanan RPG uygulama programlarının yazılmasına yardımcı olmak için çeşitli COPY dosyaları sağlanır. Üç kopya dosyası kümesi vardır:

- Adları *G* harfiyle biten COPY dosyaları, statik bağlantı kullanan programlarla kullanılır. Bu dosyalar, [“Yapılar” sayfa 996](#) içinde belirtilen özel durumlarla başlatılır.
- Adları *H* harfiyle biten COPY dosyaları, statik bağlantı kullanan, ancak **başlatılmayan** programlarla birlikte kullanılır.
- Adları *R* harfiyle biten COPY dosyaları, dinamik bağ kullanan programlarla birlikte kullanılır. Bu dosyalar, [“Yapılar” sayfa 996](#) içinde belirtilen özel durumlarla başlatılır.

COPY dosyaları, QMQM kitaplığındaki QRPGLSRC ' de bulunur.

Her bir COPY dosyası kümesi için, adlandırılmış değişmezleri içeren iki dosya ve her bir yapı için bir dosya vardır. COPY dosyaları [Çizelge 680 sayfa 994](#) içinde özetlenir.

<i>Çizelge 680. RPG kopyalama dosyaları</i>			
Dosya adı (statik bağ, ilk kullanıma hazırlandı, CMQ* G)	Dosya adı (statik bağlantı, başlatılmadı, CMQ* H)	Dosya adı (dinamik bağ, ilk kullanıma hazırlandı, CMQ* R)	İçerik
CMQBOG	CMQBOH	-	Başlangıç seçenekleri yapısı
CMQCDG	CMQCDH	CMQCDR	Kanal tanımlama yapısı
CMQCFBG	CMQCFBH	-	PCF bit süzgeci değiştirgesi

Çizelge 680. RPG kopyalama dosyaları (devamı var)

Dosya adı (statik bağ, ilk kullanıma hazırlandı, CMQ* G)	Dosya adı (statik bağlantı, başlatılmadı, CMQ* H)	Dosya adı (dinamik bağ, ilk kullanıma hazırlandı, CMQ* R)	İçerik
CMQCFG	-	-	PCF ve olaylara ilişkin değişmezler
CMQCFBSG	CMQCFBSH	-	PCF byte dizgisi
CMQCFGRG	CMQCFGRH	-	PCF grup parametresi
CMQCFIFG	CMQCFIFH	-	PCF tamsayı süzgeci değiştirgesi
CMQCFHG	CMQCFSS	-	PCF üstbilgisi
CMQCFILG	CMQCFILH	-	PCF tamsayı listesi değiştirge yapısı
CMQCFING	CMQCFINH	-	PCF tamsayı değiştirgesi yapısı
CMQCFSG	CMQCFSH	-	PCF dizgi süzgeci değiştirgesi
CMQCFSLG	CMQCFSLH	-	PCF dizgi listesi değiştirgesi yapısı
CMQCFSTG	CMQCFSTH	-	PCF dizgi değiştirgesi yapısı
CMQCFXLG	CMQCFXLH	-	CFIL64 için PCF kısa adı
CMQCFXNG	CMQCFXNH	-	CFIN64 için PCF kısa adı
CMQCIHG	CMQCISS	-	CICS bilgi üstbilgisi yapısı
CMQCNQG	CMQCNQH	-	Bağlantı seçenekleri yapısı
CMQCSQG	CMQCSQH	-	Güvenlik parametreleri
CMQCXQG	CMQCXQH	CMQCXPR	Kanal çıkış parametresi yapısı
CMQDHG	CMQDHH	CMQDHR	Dağıtım üstbilgisi yapısı
CMQDLHG	CMQDLSS	CMQDLHR	Ölü harf üstbilgisi yapısı
CMQDXQG	CMQDXQH	CMQDXPR	Veri dönüştürme çıkış parametresi yapısı
CMQEPHG	CMQEPHH	-	Yerleşik PCF üstbilgi yapısı
CMQGG	-	CMQR	Ana MQI için adlandırılmış sabitler
CMQGMQG	CMQGMQH	CMQGMOR	İleti alma seçenekleri yapısı
CMQIIHG	CMQIISS	CMQIIHR	IMS bilgi üstbilgisi yapısı
CMQMDEG	CMQMDEH	CMQMDER	İleti tanımlayıcı uzantısı yapısı
CMQMDG	CMQMDH	CMQMDR	İleti tanımlayıcı yapısı
CMQMD1G	CMQMD1H	CMQMD1R	İleti tanımlayıcı yapısı sürüm 1
CMQMD2G	CMQMD2H	-	İleti tanımlayıcı yapısı sürüm 2
CMQODG	CMQODH	CMQODR	Nesne tanımlayıcı yapısı
CMQORG	CMQORH	CMQORR	Nesne kaydı yapısı
CMQPMQG	CMQPMQH	CMQPMOR	Koyma iletisi seçenekleri yapısı
CMQPSG	-	-	Yayınlama/abone olma için sabitler
CMQRFHG	CMQRFSS	-	Kurallar ve biçimlendirme üstbilgi yapısı

Çizelge 680. RPG kopyalama dosyaları (devamı var)

Dosya adı (statik bağ, ilk kullanıma hazırlandı, CMQ* G)	Dosya adı (statik bağlantı, başlatılmadı, CMQ* H)	Dosya adı (dinamik bağ, ilk kullanıma hazırlandı, CMQ* R)	İçerik
CMQRFH2G	CMQRFH2H	-	Kurallar ve biçimlendirme üstbilgisi 2 yapısı
CMQRMHG	CMQRMSS	CMQRMHR	Başvuru iletisi üstbilgi yapısı
CMQRRG	CMQRRH	CMQRRR	Yanıt kaydı yapısı
CMQTMCG	CMQTMCH	CMQTMCR	Tetikleyici ileti yapısı (karakter biçimi)
CMQTMCG	CMQTMCH	CMQTMCR	Tetikleyici ileti yapısı (karakter biçimi) sürüm 2
CMQTMG	CMQTMH	CMQTMR	Tetikleyici ileti yapısı
CMQWIHG	CMQWIHH	-	İş bilgileri üstbilgi yapısı
CMQXG	-	CMQXR	Veri dönüştürme çıkışı için adlandırılmış sabitler
CMQXQHG	CMQXQSS	CMQXQHR	İletim kuyruğu üstbilgi yapısı

Çağrılar

Çağrılar, bireysel adları kullanılarak açıklanır.

Çağrı parametreleri

MQI 'a geçirilen bazı değıştirmelerin birden çok kořutzamanlı işlevi olabilir. Bunun nedeni, geçirilen tamsayı değerin, toplam değeri üzerinde değil, genellikle alan içindeki tek tek bitlerin ayarında sınanmasıdır. Bu, birkaç işlevi birlikte 'eklemenizi' ve bunları tek bir parametre olarak geçirmenizi sağlar.

Yapılar

Tüm IBM MQ yapıları, aşağıdaki kural dışı durumlar dışında, alanlara ilişkin ilk değerlerle tanımlanır:

- H ekine sahip herhangi bir yapı.
- MQTMC
- MQTMC2

Bu başlangıç değeri, her yapı için ilgili tabloda tanımlanır.

Yapı bildirimleri DS deyimleri içermiyor. Bu, uygulamanın DS deyimini kodlayarak ve bildirim geri kalan kısmında kopyalamak için /COPY deyimini kullanarak tek bir veri yapısını ya da çoklu geçiş veri yapısını bildirmesini sağlar:

```
D* .1.....2.....3.....4.....5.....6.....7
D* Declare an MQMD data structure with 5 occurrences
DMYMD          DS          5
D/COPY CMQMDR
```

Adlandırılmış sabitler

Uygulama programınız ile kuyruk yöneticisi arasında veri değışimi sağlayan birçok tamsayı ve karakter değeri vardır. Bu değeri kullanılmaya ilişkin daha okunabilir ve tutarlı bir yaklaşım sağlamak için, bu

değerler için adlandırılmış sabitler tanımlanır. Bu, program kaynak kodunun okunabilirliğini artırdığından, gösterdikleri değerleri değil, bu adlandırılmış değişmezleri kullanabilirsiniz.

CCMQG kopya dosyası değişmezleri tanımlamak için bir programa eklendiğinde, RPG derleyicisi program tarafından kullanılmayan sabitler için birçok önem düzeyi sıfır iletisi verir; bu iletiler iyi huyludur ve güvenle yoksayılabilir.

MQI yordamları

ILE ile bağlı çağrılar kullanırken, programınızı yaratırken MQI yordamlarına bağlanmanız gerekir. Bu yordamlar, uygun olduğu şekilde aşağıdaki hizmet programlarından dışa aktarılır:

QMQM/LIBMQM

Bu hizmet programı, sürüm 5.1 ve üstü için tek iş parçacıklı bağ tanımlarını içerir. İş parçacıklı uygulamalar yazılırken dikkat edilmesi gereken özel noktalar için aşağıdaki bölüme bakın.

QMQM/LIBMQM_R

Bu hizmet programı, sürüm 5.1 ve üstü için çok iş parçacıklı bağ tanımlarını içerir. İş parçacıklı uygulamalar yazılırken dikkat edilmesi gereken özel noktalar için aşağıdaki bölüme bakın.

QMQM/LIBMQIC

Bu hizmet programı, iş parçacığı kullanmayan istemci uygulamaları için bağ tanımlanmasını sağlar.

QMQM/LIBMQIC_R

Bu hizmet programı, iş parçacıklı istemci uygulamaları için bağ tanımlanmasını sağlar.

Programlarınızı oluşturmak için CRTPGM komutunu kullanın. Örneğin, aşağıdaki komut ILE bağlı çağrılar kullanan tek iş parçacıklı bir program yaratır:

```
CRTPGM PGM(MYPROGRAM) BNDSRVPGM(QMQM/LIBMQM)
```

İş threading ile ilgili dikkat edilecek noktalar

IBM i için kullanılan RPG derleyicisi, WebSphere Development Toolset ve WebSphere Development Studio for IBM i ' un bir parçasıdır ve ILE RPG IV Compiler olarak bilinir.

Genel olarak RPG programları, çok iş parçacıklı hizmet programlarını kullanmamalıdır. Kural dışı durumlar, ILE RPG IV Compiler kullanılarak oluşturulan ve denetim belirtiminde THREAD (*SERIALIZE) anahtar sözcüğünü içeren RPG programlarıdır. Ancak, bu programlar iş parçacığı korumalı olsa da, THREAD (*SERIALIZE) , RPG yordamlarının modül düzeyinde diziselleştirilmesini zorladığından ve bu durum genel performans üzerinde olumsuz bir etkiye sahip olabileceğinden, genel uygulama tasarımına dikkat edilmelidir.

RPG programları veri dönüştürme çıktıları olarak kullanıldığında, bunlar iş parçacığı korumalı yapılmalı ve denetim belirtiminde THREAD (*SERIALIZE) ile birlikte sürüm 4.4 ILE RPG derleyicisi ya da üstü kullanılarak yeniden derlenmelidir.

İş hakkında daha fazla bilgi için bkz. *IBM i IBM MQ Development Studio: ILE RPG Reference* ve *IBM i IBM MQ Development Studio: ILE RPG Programmer's Guide*.

Kesinleştirme denetimi

MQI syncpoint işlevleri MQCMIT ve MQBACK, olağan kipte çalışan ILE RPG programları için kullanılabilir; bu çağrılar, programın MQ kaynaklarındaki değişiklikleri kesinleştirmesini ve geri vermesini sağlar.

Bağlı çağrılar kodlanması

MQI ILE yordamları [Çizelge 681 sayfa 998](#) içinde listelenir.

Çizelge 681. Her hizmet programı tarafından desteklenen ILE RPG bağlı çağruları

Aramanın adı	LIBMQM ve LIBMQM_R	LIBMQIC ve LIBMQIC_R
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

Bu yordamları kullanmak için aşağıdakileri yapmanız gerekir:

1. 'D' belirtilerinizdeki dış yordamları tanımlayın. Bunların tümü, adı belirtilen değişmezleri içeren COPY dosya üyesi CMQG içinde kullanılabilir.
2. Yordamı parametreleriyle birlikte çağırarak için CALLP işlem kodunu kullanın.

Örneğin, MQOPEN çağrısı aşağıdaki kodun eklenmesini gerektirir:

```

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*
    
```

Yordamı çağırarak için, çeşitli parametreleri başlattıktan sonra aşağıdaki koda gereksinim duyarsınız:

```

...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
C CALLP MQOPEN(HCONN : MQOD : OPTS : HOBJ :
C CMPCOD : REASON)
    
```

Burada MQOD yapısı, kopya üyesi CMQODG kullanılarak tanımlanır ve bileşenlerine ayrılır.

Notasyonel kurallar

Bu bölümdeki ikinci konular, aşağıda belirtilenlerin nasıl olduğunu gösterir:

- Çağrılar çağrılmalıdır
- Değiştirgeler bildirilmelidir
- Çeşitli veri tipleri bildirilmelidir

Bazı durumlarda, parametreler değişmez büyüklükte olmayan diziler ya da karakter dizgileridir. Bunlar için, sayısal bir sabiti göstermek için küçük harf "n" kullanılır. Bu parametreye ilişkin bildirim kodlandığında, "n" , gerekli sayısal değerle değiştirilmelidir.

IBM i IBM i üzerinde MQAIR (Kimlik doğrulama bilgileri kaydı)

MQAIR yapısı, kimlik doğrulama bilgileri kaydını temsil eder.

Genel Bakış

Amaç: MQAIR yapısı, IBM MQ istemcisi olarak çalışan bir uygulamanın istemci bağlantısı için kullanılacak bir kimlik doğrulayıcıya ilişkin bilgileri belirtmesine olanak sağlar. Yapı, MQCONNX çağrısındaki bir giriş değiştirgesidir.

Karakter kümesi ve kodlama: MQAIR içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır.

- [“Alanlar” sayfa 999](#)
- [“Başlangıçtaki değerler” sayfa 1001](#)
- [“RPG bildirim” sayfa 1001](#)

Alanlar

MQAIR yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

AICN (10 basamaklı işaretli tamsayı)

Bu, LDAP sunucusunun çalıştığı anasistemin anasistem adı ya da ağ adresidir. Bunu, parantez içinde yer alan isteğe bağlı bir kapı numarası izleyebilir.

Değer alanın uzunluğundan kısaysa, değeri boş bir karakterle sonlandırın ya da alanın uzunluğuna kadar boşluklarla doldurun. Değer geçerli değilse, çağrı RC2387neden koduyla başarısız olur.

Varsayılan kapı numarası 389'dur.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNAICN tarafından verilir. Bu alanın ilk değeri boş karakterdir.

AITYP (10 basamaklı işaretli tamsayı)

Bu, kayıttaki kimlik doğrulama bilgilerinin tipidir.

Değer şöyle olmalıdır:

AITLDP

LDAP sunucusu kullanılarak sertifika iptal.

Değer geçerli değilse, çağrı RC2386neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri AITLDP 'dir.

AIPW (10 basamaklı işaretli tamsayı)

Bu, LDAP CRL sunucusuna erişmek için gereken paroladır.

Değer alanın uzunluğundan kısaysa, değeri boş bir karakterle sonlandırın ya da alanın uzunluğuna kadar boşluklarla doldurun. LDAP sunucusu parola gerektirmiyorsa ya da LDAP kullanıcı adını

atlırsanız, *AIPW* boş değerli ya da boş olmalıdır. LDAP kullanıcı adını atlırsanız ve *AIPW* boş değerli ya da boş değilde, arama RC2390neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNLDPW tarafından verilir. Bu alanın boş karakterlerinin ilk değeri.

AILUL (10 basamaklı işaretli tamsayı)

Bu, *AILUP* ya da *AILUO* alanı tarafından adreslenen LDAP kullanıcı adının bayt cinsinden uzunluğudur. Değer, LNDISN ile sıfır aralığında olmalıdır. Değer geçerli değilde, çağrı RC2389neden koduyla başarısız olur.

Dahil olan LDAP sunucusu bir kullanıcı adı gerektirmiyorsa, bu alanı sıfır olarak ayarlayın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

AILUO (10 basamaklı işaretli tamsayı)

Bu, LDAP kullanıcı adının MQAIR yapısının başlangıcındaki bayt cinsinden görelî konumudur.

Görelî konum pozitif ya da negatif olabilir. *LDAPUserNameLength* sıfırsa alan yoksayılr.

LDAP kullanıcı adını belirtmek için *LDAPUserNamePtr* ya da *LDAPUserNameOffset* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız; ayrıntılar için *LDAPUserNamePtr* alanının tanımına bakın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

AILUP (10 basamaklı işaretli tamsayı)

Bu, LDAP kullanıcı adıdır.

LDAP CRL sunucusuna erişmeye çalışan kullanıcının Ayırt Edici Adından oluşur. Değer *AILUL* ile belirtilen uzunluktan kısaysa, değeri boş bir karakterle sonlandırın ya da *AILUL* uzunluğuna kadar boşluklarla doldurun. *AILUL* sıfırsa alan yoksayılr.

LDAP kullanıcı adını şu iki yoldan biriyle sağlayabilirsiniz:

- *AILUP* gösterge alanını kullanarak

Bu durumda, uygulama MQAIR yapısından ayrı bir dizgi bildirebilir ve *AILUP* dizgisini dizginin adresine ayarlayabilir.

İşaretçi veri tipini destekleyen programlama dilleri için farklı ortamlarda (örneğin, C programlama dili) taşınabilir bir şekilde *AILUP* ' i kullanmayı düşünün.

- *AILUO* görelî konum alanını kullanarak

Bu durumda, uygulama MQSCO yapısını ve ardından MQAIR kayıtları dizisini ve ardından LDAP kullanıcı adı dizgilerini içeren bir bileşik yapıyı bildirmeli ve *AILUO* dizgisini MQAIR yapısının başlangıcından itibaren uygun ad dizgisinin görelî konumuna ayarlamalıdır. Bu değerin doğru olduğundan ve MQLONG içinde tutulabilecek bir değere sahip olduğundan emin olun (en kısıtlayıcı programlama dili COBOL 'dur; geçerli aralık -999 999 999 ile 999 999 999 999 arasındadır).

Gösterge veri tipini desteklemeyen programlama dilleri ya da gösterge veri tipini farklı ortamlara (örneğin, COBOL programlama dili) taşınmayacak şekilde uygulayan programlama dilleri için *AILUO* ' i kullanmayı düşünün.

Hangi teknik seçilirse seçilsin, *AILUP* ve *AILUO* yöntemlerinden yalnızca birini kullanın; Arama RC2388neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değerli gösterge ve tersi durumda tümü boş değerli bir bayt dizgisidir.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir.

AISID (10 basamaklı işaretli tamsayı)

Değer şöyle olmalıdır:

AISIDV (ISIDV)

Kimlik doğrulama bilgileri kaydının tanıtıcısı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri AISIDV 'dir.

AIVER (10 basamaklı işaretli tamsayı)

Değer şöyle olmalıdır:

AIVER1

Version-1 kimlik doğrulama bilgileri kaydı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

AIVERC.

Kimlik doğrulama bilgileri kaydının geçerli sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri AIVER1' dir.

Başlangıçtaki değerler

Çizelge 682. MQAIR için MQAIR içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
AISID	AISIDV (ISIDV)	'AIR↵'
AIVER	AIVERC	1
AITYP	AITLDP	1
AICN	Yok	Boş değerli dizilim ya da boşluklar
AILUP	Yok	Boş değerli gösterge ya da boş değerli byte
AILUO	Yok	0
AILUL	Yok	0
AIPW	Yok	Boş değerli dizilim ya da boşluklar

Notlar:

1. ↵ simgesi, tek bir boş karakteri temsil eder.

RPG bildirimini

```
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
```

İleti işleme seçeneklerine ilişkin arabelleği tanımlayan yapı.

Genel Bakış

Amaç: MQBMHO yapısı, uygulamaların ileti tanıtıcılarının arabelleklerden nasıl üretildiğini denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQBUFMH çağrısındaki bir giriş değişirgesidir.

Karakter kümesi ve kodlama: MQBMHO içindeki veriler, uygulamanın (ENNAT) karakter kümesinde ve kodlamasında olmalıdır.

- [“Alanlar” sayfa 1002](#)
- [“Başlangıçtaki değerler” sayfa 1003](#)
- [“RPG bildirimi” sayfa 1003](#)

Alanlar

MQBMHO yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

BMSID (10 basamaklı işaretli tamsayı)

İleti işleme yapısı için arabellek- StrucId alanı.

Bu, yapı tanıtıcısıdır. Değer şöyle olmalıdır:

BMSIDV

İleti işleme yapısına ilişkin arabellek tanıtıcısı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri BMSIDV 'dir.

BMVER (10 basamaklı işaretli tamsayı)

İleti tanıtıcısı yapısı-Sürüm alanı için arabellek.

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

BMVER1

İleti işleme yapısına ilişkin arabelleğin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

BMVERVC

İleti işleme yapısına ilişkin arabelleğin yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri BMVER1' dir.

BMOPT (10 basamaklı işaretli tamsayı)

İleti işleme yapısı-Seçenekler alanı için arabellek.

Değer şöyle olabilir:

BMDLPR

İleti tanıtıcısına eklenen özellikler arabellekten silinir. Arama başarısız olursa, hiçbir özellik silinmez.

Varsayılan seçenekler: Açıklanan seçeneğe gerek duymuyorsanız, aşağıdaki seçeneği kullanın:

BMNONE

Seçenek belirtilmedi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri BMDLPR 'dir.

Başlangıçtaki değerler

Çizelge 683. MQBMHO içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
BMSID	BMSIDV	'BMHO'
BMVER	BMVER1	1
BMOPT	BMNONE	0

RPG bildirim

```
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 üzerinde MQBO (Başlangıç seçenekleri)

MQBO yapısı, uygulamanın bir iş birimi yaratılmasıyla ilgili seçenekleri belirtmesine olanak sağlar.

Genel Bakış

Amaç: Yapı, MQBEGIN çağrısında bir giriş/çıkış değiştirgesidir.

Karakter kümesi ve kodlama: MQBO 'daki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından verilen karakter kümesinde olmalıdır.

- “Alanlar” sayfa 1003
- “Başlangıçtaki değerler” sayfa 1004
- “RPG bildirim” sayfa 1004

Alanlar

MQBO yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

BOOPT (10 basamaklı işaretli tamsayı)

MQBEGIN işlemini denetleyen seçenekler.

Değer şöyle olmalıdır:

BONONE.

Seçenek belirtilmedi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri BONONE.

BOSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

BOSIDV

Başlangıç seçenekleri yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri BOSIDV 'dir.

BOVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

BOVER1

Başlangıç seçenekleri yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

BOVERC.

Başlangıç seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri BOVER1' dir.

Başlangıçtaki değerler

Çizelge 684. MQBO 'daki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
BOSID	BOSIDV	'BO- -'
BOVER	BOVER1	1
BOOPT	BONONE.	0

Notlar:

1. - simgesi, tek bir boş karakteri temsil eder.

RPG bildirim

```
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 üzerinde MQCBC (Geri çağırma bağlamı)

Geri çağırma yordamını tanımlayan yapı.

Genel Bakış

Amaç

MQCBC yapısı, bir geri çağırma işlevine geçirilen bağlam bilgilerini belirtmek için kullanılır.

Yapı, bir ileti tüketicisi yordamına yapılan çağrıdaki bir giriş/çıkış değiştirgesidir.

Sürüm

MQCBC 'nin yürürlükteki sürümü CBCV2.

Karakter kümesi ve kodlama

MQCBC içindeki veriler, ENNAT tarafından verilen yerel kuyruk yöneticisinin **CodedCharSetId** kuyruk yöneticisi özneliği ve kodlaması tarafından belirtilen karakter kümesinde yer almaktadır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde yer alması için kodlamasında yer alıyor.

- “Alanlar” sayfa 1005
- “Başlangıçtaki değerler” sayfa 1010
- “RPG bildirimi” sayfa 1010

Alanlar

MQCBC yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

CBCBUFFLEN (10 basamaklı işaretli tamsayı)

Arabellek, tüketici için tanımlanan MaxMsgUzunluğu değerinden ve MQGMO ' da ReturnedLength değerinden büyük olabilir.

Geri çağırma bağlamı yapısı- BufferLength alanı.

Bu, bu işleve geçirilen ileti arabelleğinin bayt cinsinden uzunluğudur.

Gerçek ileti uzunluğu DataLength alanında sağlanır.

Uygulama, geri çağırma işlevi süresince tüm arabelleği kendi amaçları için kullanabilir.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; kural dışı durum işleyici işleviyle ilgili değildir.

CBCCALLBA (10 basamaklı işaretli tamsayı)

Geri çağırma bağlamı yapısı- CallbackArea alanı.

Bu, geri çağırma işlevinin kullanabileceği bir alandır.

Kuyruk yöneticisi bu alanın içeriğine dayalı olarak karar vermez ve geri çağırma işlevini tanımlamak için kullanılan MQCB çağrısında bir parametre olan MQCBD yapısındaki CBDCALLBA alanından değiştirilmeden geçirilir.

CBCCALLBA üzerinde yapılan değişiklikler, bir *CBCHOBJ* için geri çağırma işlevinin çağrılmaları boyunca korunur. Bu alan, diğer tutamaçlar için geri çağırma işlevleriyle paylaşılmaz.

Bu, geri çağırma işlevine ilişkin bir giriş/çıkış alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

CBCCALLT (10 basamaklı işaretli tamsayı)

Geri Çağırma Bağlamı yapısı- CallType alanı.

Bu işlevin çağrılmasıyla ilgili bilgileri içeren alan. Aşağıdaki arama tipleri tanımlandı.

İleti teslim çağrısı tipleri: Bu çağrı tipleri bir iletiyle ilgili bilgi içerir. **CBCCLEN** ve **CBCBUFFLEN** parametreleri bu çağrı tipleri için geçerlidir.

CBCTMR

İleti tüketicisi işlevi, nesne tanıtıcısından yıkıcı bir şekilde kaldırılmış bir iletiyle çağrıldı.

CBCCC değeri CCWARN ise, *Reason* alanının değeri RC2079 ya da bir veri dönüştürme sorununu gösteren kodlardan biridir.

CBCTMN

İleti tüketicisi işlevi, henüz nesne tanıtıcısından yıkıcı bir şekilde kaldırılmamış bir iletiyle çağrıldı.

İleti, *MsgToken* kullanılarak nesne tanıtıcısından yıkıcı bir şekilde kaldırılabilir.

İleti aşağıdaki nedenden ötürü kaldırılmamış olabilir:

- MQGMO seçenekleri bir göz atma işlemi istedi, GMBR*
- İleti, kullanılabilir arabellekten büyük ve MQGMO seçenekleri gmatm belirtmiyor

CBCCC değeri CCWARN ise, *Reason* alanının değeri RC2080 ya da bir veri dönüştürme sorununu gösteren kodlardan biridir.

Geri çağırma denetimi arama tipleri: Bu çağrı tipleri, geri çağırma denetimine ilişkin bilgileri içerir ve bir iletiyle ilgili ayrıntıları içermez. Bu çağrı tipleri, MQCBD yapısında CBDOPT kullanılarak isteniyor.

CBCCLEN ve **CBCCBUFFLEN** parametreleri bu çağrı tipleri için geçerli değildir.

CBCTRC

Bu çağrı tipinin amacı, geri çağırma işlevinin bazı ilk ayarları gerçekleştirmesine izin vermektir.

Geri çağırma işlevi, geri çağırma kaydedildikten hemen sonra çağrılır; yani, CBREG ' nin *Operation* alanı için bir değer kullanılarak bir MQCB çağrısından döndükten sonra çağrılır.

Bu çağrı tipi hem ileti tüketicileri hem de olay işleyiciler için kullanılır.

İstenirse, bu, geri çağırma işlevinin ilk çağrısıdır.

CBCCREA alanının değeri RCNONE 'dir.

CBCTSC

Bu çağrı tipinin amacı, geri çağırma işlevinin başlatıldığında bazı ayarları gerçekleştirmesine izin vermektir; örneğin, önceden durdurulduğunda temizlenen kaynakları yeniden sağlamak.

Geri çağırma işlevi, bağlantı CTLSR ya da CTLSW kullanılarak başlatıldığında çağrılır.

Bir geri çağırma işlevi başka bir geri çağırma işlevinde kayıtlıysa, geri çağırma döndüğünde bu çağrı tipi çağrılır.

Bu çağrı tipi yalnızca ileti tüketicileri için kullanılır.

CBCCREA alanının değeri RCNONE 'dir.

CBCTTC

Bu çağrı tipinin amacı, geri çağırma işlevinin bir süre durdurulduğunda (örneğin, iletilerin tüketimi sırasında edinilen ek kaynakların temizlenmesi) bir temizleme işlemi gerçekleştirmesini sağlamaktır.

Geri çağırma işlevi, CTLSP ' nin *Operation* alanı için bir değer kullanılarak bir MQCTL çağrısı verildiğinde çağrılır.

Bu çağrı tipi yalnızca ileti tüketicileri için kullanılır.

CBCCREA alanının değeri, durdurma nedenini gösterecek şekilde ayarlanır.

CBCTDC

Bu çağrı tipinin amacı, geri çağırma işlevinin, tüketme işleminin sonunda son temizlemeyi gerçekleştirmesine izin vermektir. Geri çağırma işlevi aşağıdaki durumda çağrılır:

- Geri çağırma işlevinin kaydı BCUNR ile bir MQCB çağrısı kullanılarak kaldırılmıştır.
- Kuyruk kapatıldı, örtük bir kayıttan çıkarıcıya neden oluyor. Bu örnekte, geri çağırma işlevine nesne tanıtıcısı olarak HOUNUH geçilir.
- MQDISC çağrısı tamamlandı-örtük bir kapatmaya ve dolayısıyla kayıttan kaldırılmasına neden oldu. Bu durumda, bağlantı hemen kesilmez ve devam eden hareketler henüz kesinleştirilmez.

Geri çağırma işlevinin kendisinde bu işlemlerden herhangi biri gerçekleştirilirse, geri çağırma işlemi geri döndürdükten sonra işlem çağrılır.

Bu çağrı tipi hem ileti tüketicileri hem de olay işleyiciler için kullanılır.

İstenirse, bu, geri çağırma işlevinin son çağrısıdır.

CBCCREA alanının değeri, durdurma nedenini gösterecek şekilde ayarlanır.

CBCTEC

Olay işleyici işlevi

Olay işleyici işlevi aşağıdaki durumlarda ileti olmadan çağrıldı:

- CTLSP ' nin *Operation* alanı için bir değer içeren bir MQCTL çağrısı yayınlandı ya da
- Kuyruk yöneticisi ya da bağlantı durdurulur ya da susturulur.

Bu çağrı, tüm geri çağırma işlevleri için uygun işlemi gerçekleştirmek üzere kullanılabilir.

- **İleti tüketici işlevi**

Nesne tanıtıcısına özgü bir hata (CBCCC = CCFAIL) saptandığında ileti tüketici işlevi bir ileti olmadan çağrıldı; örneğin, CBCREA kod = RC2016 .

CBCREA alanının değeri, aramanın nedenini gösterecek şekilde ayarlanır.

Bu bir giriş alanıdır. CBCTMR ve CMCTMN yalnızca ileti tüketici işlevleri için geçerlidir.

CBCCC (10 basamaklı işaretli tamsayı)

Geri çağırma bağlamı yapısı- CompCode alanı.

Bu tamamlama kodudur. Bu, iletiyi tüketen herhangi bir sorun olup olmadığını gösterir; aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı

CCWARN

Uyarı (kısmi tamamlama)

CCFAIL

Arama başarısız oldu

Bu bir giriş alanıdır. Bu alanın ilk değeri CCOK ' dur.

CBCCONNAREA (10 basamaklı işaretli tamsayı)

Geri çağırma bağlamı yapısı- ConnectionArea alanı.

Bu, geri çağırma işlevinin kullanabileceği bir alandır.

Kuyruk yöneticisi bu alanın içeriğine dayalı olarak karar vermez ve geri çağırma işlevini denetlemek için kullanılan MQCTL çağrısındaki bir parametre olan MQCTLO yapısındaki ConnectionArea alanından değiştirilmeden geçirilir.

Geri çağırma işlevleri tarafından bu alanda yapılan değişiklikler, geri çağırma işlevinin çağrılmaları boyunca korunur. Bu alan, tüm geri çağırma işlevleri tarafından paylaşılacak bilgileri iletmek için kullanılabilir. *CallbackArea* ' in tersine, bu alan bir bağlantı tanıtıcısı için tüm geri çağırma arasında yaygındır.

Bu bir giriş ve çıkış alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

CBCLLEN (10 basamaklı işaretli tamsayı)

Bu, iletideki uygulama verilerinin bayt cinsinden uzunluğudur. Değer sıfırsa, iletinin uygulama verisi içermediği anlamına gelir.

CBCLLEN alanı, iletinin uzunluğunu içerir, ancak tüketiciye iletilen ileti verilerinin uzunluğunu içermez. İleti kesilmiş olabilir. Tüketiciye ne kadar veri aktarıldığını saptamak için MQGMO ' daki GMRL alanını kullanın.

Neden kodu iletinin kesildiğini gösteriyorsa, gerçek iletinin ne kadar büyük olduğunu saptamak için CBCLLEN alanını kullanabilirsiniz. Bu, ileti verilerini barındırmak için gereken arabellek büyüklüğünü saptamanızı ve MQCBD 'de CBDMML ' yi uygun bir değerle güncellemek için bir MQCB çağrısı yürütmenizi sağlar.

GMCONV seçeneği belirtilirse, dönüştürülen ileti DataLength için döndürülen değerden büyük olabilir. Bu tür durumlarda, uygulamanın MQCBD 'deki CBDMML ' yi DataLength kuyruk yöneticisinin döndürdüğü değerden büyük olacak şekilde güncellemek için bir MQCB çağrısı yayınlaması gerekir.

İleti kesme sorunlarını önlemek için, MaxMsgUzunluğunu CBDFM olarak belirtin. Bu, kuyruk yöneticisinin veri dönüştürme işleminden sonra tam ileti uzunluğu için bir arabellek ayırmasına neden olur. Ancak, bu seçenek belirtilse bile, isteği doğru olarak işlemek için yeterli saklama alanı bulunmadığını unutmayın. Uygulamalar her zaman döndürülen neden kodunu kontrol etmelidir. Örneğin, iletiyi dönüştürmek için yeterli bellek ayrılmadıysa, iletiler uygulamaya dönüştürülmemiş olarak döndürülür.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; bir olay işleyici işleviyle ilgili değildir.

CBCFLG (10 basamaklı işaretli tamsayı)

Bu tüketici hakkında bilgi içeren işaretler.

Aşağıdaki seçenek tanımlandı:

CBCFBE

COQSC seçeneğini kullanan önceki bir MQCLOSE çağrısı RC2458neden koduyla başarısız olursa bu işaret döndürülebilir.

Bu kod, son önden okuma iletilerinin döndürüldüğünü ve arabelleğin artık boş olduğunu belirtti. Uygulama COQSC seçeneğini kullanarak başka bir MQCLOSE çağrısı yaparsa başarılı olur.

Önden okuma arabelleğinde yürürlükteki seçim ölçütleriyle eşleşmeyen iletiler olabileceği için, bir uygulamaya bu işaret kümesiyle bir ileti verileceği garanti edilmez. Bu örnekte, tüketici işlevi RC2019 neden koduyla çağrılır.

Önden okuma arabelleği boşsa, tüketici CBCFBE işaretiyle ve RC2518neden koduyla çağrılır.

Bu, ileti tüketici işlevine ilişkin bir giriş alanıdır; bir olay işleyici işleviyle ilgili değildir.

CBCHOBJ (10 basamaklı işaretli tamsayı)

Geri çağırma bağlam yapısı-CBCHOBJ alanı.

Bir ileti tüketicisine yapılan çağrı için bu, ileti tüketicisine ilişkin nesnenin tanıtıcısıdır.

Olay işleyici için bu değer HONONE 'tur

Uygulama, kuyruktan bir ileti kaldırılmamışsa, iletiyi almak için İleti Alma Seçenekleri bloğundaki ileti simgesini ve bu tanıtıcıyı kullanabilir.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri HOUNUH

CBCRCD (10 basamaklı imzalı tamsayı)

CBCRCD , kuyruk yöneticisinin yeniden bağlanmayı denemeden önce ne kadar bekleyeceğini gösterir. Alan, gecikmeyi değiştirmek ya da yeniden bağlantıyı tamamen durdurmak için bir olay işleyici tarafından değiştirilebilir.

Geri Çağırma Bağlamındaki **Reason** alanının değeri RC2545ise **CBCRCD** alanını kullanın.

Olay işleyiciye girişte **CBCRCD** değeri, kuyruk yöneticisinin yeniden bağlantı girişiminde bulunmadan önce bekleyeceği milisaniye sayısıdır. [Çizelge 685 sayfa 1008](#) içinde, olay işleyiciden dönüşte kuyruk yöneticisinin davranışını değiştirmek için ayarlayabileceğiniz değerler listelenir.

<i>Çizelge 685. CBCRCD değerler</i>	
Değer	Açıklama
-1	Yeniden bağlanma girişiminde bulunma. Uygulamaya bir hata döndürüldü.
0	Hemen yeniden bağlanmayı deneyin.
>0	Bağlantıyı yeniden denemeden önce bu kadar milisaniye bekleyin.

CBCREA (10 basamaklı işaretli tamsayı)

Geri çağırma bağlamı yapısı-Neden alanı.

CBCCC kodunu niteleyen neden kodudur.

Bu bir giriş alanıdır. Bu alanın ilk değeri RCNONE.

CBCSTATE (10 basamaklı işaretli tamsayı)

Geçerli tüketicinin durumunun bir göstergesi. Bu alan, tüketici işlevine sıfır dışında bir neden kodu iletildiğinde bir uygulamaya en çok değer verir.

Her neden kodu için davranış kodlamanız gerekmediğinden uygulama programlamayı basitleştirmek için bu alanı kullanabilirsiniz.

Bu bir giriş alanıdır. Bu alanın ilk değeri CSNONE.

Çizelge 686. CBCSTATE değerleri ve sonuçtaki işlemler		
İl	Kuyruk yöneticisi işlemi	Değişmezin değeri
<i>CSNONE</i> Bu neden kodu, ek neden bilgisi olmadan olağan bir çağrıyı gösterir	Yok, bu normal bir işlem.	0
<i>CSSUST</i> Bu neden kodları geçici koşulları gösterir.	Koşulu bildirmek için geri çağırma yordamı çağrılır ve sonra askıya alınır. Bir süre sonra sistem işlemi yeniden deneyebilir, bu da aynı koşulun yeniden oluşmasına neden olabilir.	1
<i>CSSUSU</i> Bu neden kodları, geri çağırmanın koşulu çözmek için işlem yapması gereken durumları gösterir.	Tüketici askıya alındı ve koşulu bildirmek için geri çağırma yordamı çağrıldı. Geri çağırma yordamı olanağı varsa koşulu çözmeli ve RESUME ya da bağlantıyı kapatmalıdır.	2
<i>CSSUS</i> Bu neden kodları, ileti geri çağırmasını önleyen başarısızlıkları gösterir.	Kuyruk yöneticisi, geri çağırma işlevini otomatik olarak askıya alır. Geri çağırma işlevi sürdürülürse, aynı neden kodunu yeniden alma olasılığı yüksektir.	3
<i>CSSTOP</i> Bu neden kodları, ileti tüketiminin sonunu gösterir.	Kural dışı durum işleyicisine ve belirtilen CBDTC 'yi geri çağırmaya teslim edildi. Başka ileti tüketilemez.	4

CBCSID (10 basamaklı imzalı tamsayı)

Geri çağırma bağlamı yapısı- StrucId alanı.

Bu yapı tanıttırıcıdır; değer şu olmalıdır:

CBCSI

Geri çağırma bağlamı yapısına ilişkin tanıttırıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CBCSI 'dir.

CBCVER (10 basamaklı imzalı tamsayı)

Geri çağırma bağlamı yapısı-Sürüm alanı.

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

CBCV1

Version-1 geri çağırma bağlamı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

CBCCV

Geri çağırma bağlamı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CBCV1' dir.

Başlangıçtaki değerler

Çizelge 687. MQCBC içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
CBCSID	CBCSI	'CBC¬'
CBCVER	CBCV1	1
CBCCALLT	Yok	0
CBCHOBJ	HOUNUH.	-1
CBCCALLBA	Yok	Boş değerli gösterge ya da boş değerli byte
CBCCONNAREA	Yok	Boş değerli gösterge ya da boş değerli byte
CBCCC	CCOK	0
CBCREA	RCNONE	0
CBCSTATE	CSNONE	0
CBCLLEN	Yok	0
CBCBUFFLEN	Yok	0
CBCFLG	Yok	0
CBCRCD	yok	0

Not:

1. ¬ simgesi, tek bir boş karakteri temsil eder.

RPG bildirimi

```
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 CBCLLEN       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 IBM i üzerinde MQCBD (Callback descriptor)

Geri çağırma işlevini belirten yapı.

Genel Bakış

Amaç: MQCBD yapısı, bir geri çağırma işlevini ve kuyruk yöneticisi tarafından kullanımını denetleyen seçenekleri belirtmek için kullanılır.

Yapı, MQCB çağrısındaki bir giriş değiştirgesidir.

Sürüm: MQCBD ' nin yürürlükteki sürümü CBDV1.

Karakter kümesi ve kodlama: MQCBD ' deki veriler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır; bunlar **CodedCharSetId** kuyruk yöneticisi özneliği ve ENNAT tarafından verilir. Ancak, uygulama IBM MQ MQI clientolarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

- [“Alanlar” sayfa 1011](#)
- [“Başlangıçtaki değerler” sayfa 1015](#)
- [“RPG bildirim” sayfa 1015](#)

Alanlar

MQCBD yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

CBDCALLBA (10 basamaklı işaretli tamsayı)

Bu, geri çağırma işlevinin kullanabileceği bir alandır.

Kuyruk yöneticisi bu alanın içeriğine dayalı olarak karar vermez ve geri çağırma işlevi bildirimindeki bir değiştirge olan MQCBD yapısındaki **CBDCALLBA** alanından değiştirilmeden geçirilir.

Değer yalnızca CBREG değerine sahip bir *Operation* için kullanılır; tanımlı geri çağırma yok, önceki bir tanımlamanın yerine geçmez.

Bu, geri çağırma işlevine ilişkin bir giriş ve çıkış alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

CBDCALLBF (10 basamaklı işaretli tamsayı)

Geri çağırma işlevi bir işlev çağırısı olarak çağrılır.

Geri çağırma işlevine ilişkin bir gösterge belirtmek için bu alanı kullanın.

CallbackFunction ya da *CallbackName* belirtmeniz gerekir. Her ikisini de belirtirseniz, RC2486 neden kodu döndürülür.

CallbackName ya da *CallbackFunction* ayarlanmazsa, arama RC2486neden koduyla başarısız olur.

Bu seçenek aşağıdaki ortamlarda desteklenmez:

- CICS Değiştirildiği tarih/saat: z/OS
- İşlev işaretçisi başvurularını desteklemeyen programlama dilleri ve derleyiciler

Böyle durumlarda, çağrı RC2486neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

CBDSCALLBN (10 basamaklı işaretli tamsayı)

Geri çağırma işlevi dinamik olarak bağlı bir program olarak çağrılır.

CallbackFunction ya da *CallbackName* belirtmeniz gerekir. Her ikisini de belirtirseniz, RC2486 neden kodu döndürülür.

CallbackName ya da *CallbackFunction* doğru değilse, arama RC2486neden koduyla başarısız olur.

Kullanılacak ilk geri çağırma yordamı kaydedildiğinde birim yüklenir ve bu yordamı kullanan son geri çağırma yordamı kayıttan kaldırıldığında birim kaldırılır.

Aşağıdaki metinde belirtilenler dışında, ad alan içinde sola hizalanır ve boşluk bırakılmaz; adın kendisi alan uzunluğuna kadar boşluklarla doldurulur. Aşağıdaki açıklamalarda, köşeli ayraç ([]) isteğe bağlı bilgileri gösterir:

IBMi

Geri çağırma adı aşağıdaki biçimlerden biri olabilir:

- Kitaplık "/" Programı
- Kitaplık "/" ServiceProgram ("FunctionName")

Örneğin, MyLibrary/MyProgram(MyFunction).

Kitaplık adı *LIBL olabilir. Hem kitaplık hem de program adları en çok 10 karakterle sınırlıdır.

AIX and Linux

Geri çağırma adı, dinamik olarak yüklenebilir bir modülün ya da kitaplığın adıdır; soneki o kitaplıkta bulunan bir işlevin adıdır. İşlev adı ayraç içine alınmalıdır. Kitaplık adının başına isteğe bağlı olarak bir dizin yolu konabilir:

```
[path]library(function)
```

Yol belirtilmezse, sistem arama yolu kullanılır.

Ad en çok 128 karakterle sınırlıdır.

Windows

Geri çağırma adı, bu kitaplıkta bulunan bir işlevin adının eklendiği dinamik bağlantı kitaplığının adıdır. İşlev adı ayraç içine alınmalıdır. Kitaplık adının başına isteğe bağlı olarak bir dizin yolu ve sürücü öneki konabilir:

```
[d:][path]library(function)
```

Sürücü ve yol belirtilmezse, sistem arama yolu kullanılır.

Ad en çok 128 karakterle sınırlıdır.

z/OS

Geri çağırma adı, LINK ya da LOAD makrosunun EP parametresinde belirtim için geçerli olan bir yükleme modülünün adıdır.

Ad en çok 8 karakterle sınırlıdır.

z/OS CICS

Geri çağırma adı, EXEC CICS LINK komut makrosunun PROGRAM parametresinde belirtim için geçerli olan bir yükleme modülünün adıdır.

Ad en çok 8 karakterle sınırlıdır.

Program, kurulu PROGRAM tanımının REMOTESYTEM seçeneği ya da dinamik yöneltme programı kullanılarak uzak olarak tanımlanabilir.

Program IBM MQ API çağrılarını kullanacaksa, uzak CICS bölgesi IBM MQ ' e bağlanmalıdır. Ancak, MQCBC yapısındaki CBCHOBJ alanının uzak bir sistemde geçerli olmadığını unutmayın.

CallbackName yüklenirken bir hata oluşursa, uygulamaya aşağıdaki hata kodlarından biri döndürülür:

- RC2495
- RC2496
- RC2497

Ayrıca, yüklemenin denendiği birimin adını ve işletim sisteminden gelen neden kodunu içeren hata günlüğüne bir ileti yazılır.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş bir dizilim ya da boşluktur.

CBDSCALLBT (10 basamaklı işaretli tamsayı)

Bu, geri çağırma işlevinin tipidir. Değer şunlardan biri olmalıdır:

CBTMC

Bu geri çağırma ileti tüketici işlevi olarak tanımlar.

Belirtilen seçim ölçütlerini karşılayan bir ileti bir nesne tanıtıcısında kullanılabilir olduğunda ve bağlantı başlatıldığında, ileti tüketicisi geri çağırma işlevi çağrılır.

CBTEH

Bu geri çağırma zamanuysuz olay yordamı olarak tanımlar; bir tanıtıcı için iletileri tüketmeye yönlendirilmez.

Olay işleyiciyi tanımlayan MQCB çağrısında *Hobj* gerekli değildir ve belirtildiyse yoksayılır.

Olay işleyici, tüm ileti tüketici ortamını etkileyen koşullar için çağrılır. Bir olay (örneğin, kuyruk yöneticisi ya da bağlantı durdurma ya da susturma) oluştuğunda, tüketici işlevi ileti olmadan çağrılır. Tek bir ileti tüketicisine özgü koşullar için çağrılmaz; örneğin, RC2016.

Olaylar, aşağıdaki ortamlar dışında, bağlantının başlatılıp başlatılmamasından ya da durdurulmasından bağımsız olarak uygulamaya teslim edilir:

- z/OS ortamında CICS
- iş parçacıklı olmayan uygulamalar

Çağırma bu değerlerden birini geçirmezse, çağrı RC2483 neden koduyla başarısız olur.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CBTMC 'dir.

CBDMML (10 basamaklı işaretli tamsayı)

Bu, tanıtıcıdan okunabilen ve geri çağırma yordamına verilen en uzun iletinin bayt cinsinden uzunluğudur. Bir iletinin uzunluğu daha uzunsa, geri çağırma yordamı iletinin *MaxMsgLength* byte 'ını alır ve neden kodu:

- RC2080 ya da
- RC2079 GMATM belirlediyseniz.

Gerçek ileti uzunluğu, MQCBC yapısının "CBCLLEN (10 basamaklı işaretli tamsayı)" sayfa 1007 alanında sağlanır.

Aşağıdaki özel değer tanımlandı:

CBDFM

Arabellek uzunluğu, sistem tarafından kesilmeden ileti döndürecek şekilde ayarlanır.

İletiyi almak üzere bir arabellek ayırmak için yeterli bellek yoksa, sistem geri çağırma işlevini RC2071 neden koduyla çağırır.

Örneğin, veri dönüştürme isteğinde bulunursanız ve ileti verilerini dönüştürmek için yeterli bellek yoksa, dönüştürülemeyen ileti geri çağırma işlevine geçirilir.

Bu bir giriş alanıdır. *MaxMsgLength* alanının ilk değeri CBDFM 'dir.

CBDOPT (10 basamaklı işaretli tamsayı)

Geri çağırma tanımlayıcısı yapısı-Seçenekler alanı.

Aşağıdakilerden herhangi biri ya da tümü belirtilebilir. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyorsa). Geçerli olmayan birleşimler belirtilir; diğer birleşimler geçerlidir.

CBDFQ

Kuyruk yöneticisi susturma durumundaysa MOCB çağrısı başarısız olur.

z/OS sistemlerinde bu seçenek, bağlantı (bir CICS ya da IMS uygulaması için) susturma durumundaysa MOCB çağrısına da başarısız olmasına neden olur.

MOCB çağrısında geçirilen MQGMO seçeneklerinde, ileti tüketicilerine susturulurken bildirim gönderilmesini sağlamak için GMFIQ değerini belirtin.

Denetim seçenekleri: Aşağıdaki seçenekler, tüketicinin durumu değiştiğinde, ileti olmadan geri çağırma işlevinin çağrılıp çağrılmayacağını denetler:

CBDRC

Geri çağırma işlevi CBCTRC çağrı tipiyle çağrıldı

CBDESC

Geri çağırma işlevi CBCTSC çağrı tipiyle çağrılır.

CBDTC

Geri çağırma işlevi CBCTTC çağrı tipiyle çağrılır.

CBDDC

Geri çağırma işlevi CBCTDC çağrı tipiyle çağrılır.

Bu arama tipleriyle ilgili daha fazla ayrıntı için bkz. [“CBCCALLT \(10 basamaklı işaretli tamsayı\)” sayfa 1005](#).

Varsayılan seçenek: Açıklanan seçeneklerden herhangi birine gerek duymuyorsanız, aşağıdaki seçeneği kullanın:

CDNO

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar.

CDNO, program belgelerine yardımcı olmak üzere tanımlanır; bu seçeneğin başka bir değerle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım saptanamaz.

Bu bir giriş alanıdır. *Options* alanının ilk değeri CBDNO değeridir.

CBDSID (10 basamaklı işaretli tamsayı)

Geri çağırma tanımlayıcısı yapısı- StrucId alanı.

Bu yapı tanıtcısıdır; değer şu olmalıdır:

CBDSI

Geri çağırma tanımlayıcısı yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CBDSI 'dir.

CBDVER (10 basamaklı işaretli tamsayı)

Geri çağırma tanımlayıcı yapısı-Sürüm alanı.

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

CBDV1

Version-1 geri çağırma tanımlayıcı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

CBDCV

Geri çağırma tanımlayıcısı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CBDV1' dir.

Başlangıçtaki değerler

Alan adı	Değişmezin adı	Değişmezin değeri
<i>StrucId</i>	CBDSI	'CBD↵'
<i>Version</i>	CBDV1	1
<i>CallbackType</i>	CBTMC	1
<i>Options</i>	CDNO	0
<i>CallbackArea</i>	Yok	Boş değerli byte
<i>CallbackFunction</i>	Yok	Boş değerli byte
<i>CallbackName</i>	Yok	Boşluklar
<i>MaxMsgLength</i>	CBD FM	-1

Not:

1. ↵ simgesi, tek bir boş karakteri temsil eder.

RPG bildirimi

```
D* MÇCBD 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*
D* Callback name
D  CBDCALLBN       49     176   INZ('\0')
D*
D* Maximum message length
D  CBDMML          177    180I 0 INZ(-1)
```

IBM i üzerinde MÇCHARV (Değişken Uzunluk Dizisi)

Değişken uzunluklu bir diziyi tanımlamak için MÇCHARV yapısını kullanın.

Genel Bakış

Karakter kümesi ve kodlama: MQCHARV içindeki veriler, ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlamasında ve yapı içindeki VCHRC alanının karakter kümesinde olmalıdır. Uygulama IBM MQ MQI clientolarak çalışıyorsa, yapı istemcinin kodlamasında olmalıdır. Bazı karakter kümelerinin kodlamaya bağlı bir gösterimi vardır. VCHRC bu karakter kümelerinden biriye, kullanılan kodlama, MQCHARV ' deki diğer alanlarla aynı kodlamadır. VSCSID ile tanıtilan karakter takımı çift baytlık karakter takımı (DBCS) olabilir.

Kullanım: MQCHARV yapısı, içerdiği yapıyla bitişik olmayabilecek verileri adresler. Bu verileri ele almak için, işaretçi veri tipiyle bildirilen alanlar kullanılabilir.

- “Alanlar” sayfa 1016
- “Başlangıçtaki değerler” sayfa 1017
- “RPG bildirim” sayfa 1017
- “CSAPL ' nin yeniden tanımlanması” sayfa 1018

Alanlar

MQCHARV yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

VCHRC (10 basamaklı işaretli tamsayı)

Bu, VCHRP ya da VCHRO alanı tarafından adreslenen değişken uzunluklu dizilimin karakter takımı tanıtıcısıdır.

Bu alanın ilk değeri CSAPL 'dir. Bu, kuyruk yöneticisi tarafından kuyruk yöneticisinin gerçek karakter kümesi tanıtıcısı olarak değiştirilmesi gerektiğini belirtmek için IBM MQ tarafından tanımlanır. Bu, CSQM ' nin davranışlarının aynısı. Sonuç olarak, CSAPL değeri hiçbir zaman değişken uzunluklu bir dizgiyle ilişkilendirilmez. Bu alanın ilk değeri, uygulamanızın programlama dili için uygun yöntemlerle derleme biriminize ilişkin değişmez CSAPL için farklı bir değer tanımlanarak değiştirilebilir.

VCHRL (10 basamaklı işaretli tamsayı)

VCHRP ya da VCHRO alanı tarafından adreslenen değişken uzunluklu dizilimin bayt cinsinden uzunluğu.

Bu alanın ilk değeri 0 'dır. Değer sıfırdan büyük ya da sıfıra eşit ya da tanınan şu özel değer olmalıdır:

VSNTL

VSNTL belirlenmezse, VCHRL baytları dizilimin bir parçası olarak eklenir. Boş karakterler varsa, dizgiyi sınırlamazlar.

VSNTL belirtilirse, dizgi, dizgide karşılaşılan ilk boş değerle sınırlanmış olur. Boş değer, o dizginin bir parçası olarak dahil edilmez.

Not: VSNTL değeri VCHRC tarafından belirlenen kod takımındaki boş değerse, bir dizgiyi sonlandırmak için kullanılan boş karakter.

Örneğin, UTF-16 (CCSID 1200, 13488 ve 17584) içinde, bu 2 baytlık Unicode kodlamasıdır; burada boş değer, tüm sıfırların 16 bitlik sayısı gösterilir. UTF-16 'da, karakterlerin (örneğin 7 bit ASCII karakterleri) bir parçası olan tüm sıfıra ayarlanmış tek baytları bulmak yaygındır, ancak dizgiler yalnızca çift bayt sınırında iki ' sıfır ' bayt bulunduğunda boş sonlandırılır. Her biri geçerli karakterlerin bir parçası olduğunda tek bir sınırda iki ' sıfır ' bayt almak mümkündür. Örneğin, x '01' x '00' x '00' x '30' geçerli iki Unicode karakteri temsil eder ve dizgiyi boş değerli olarak sonlandırmaz.

VCHRO (10 basamaklı işaretli tamsayı)

Değişken uzunluklu dizginin MQCHARV ' nin başlangıcındaki ya da bunu içeren yapıdaki bayt cinsinden görel konumu.

MQCHARV yapısı başka bir yapıya yerleştirildiğinde, bu değer, bu MQCHARV yapısını içeren yapının başlangıcındaki değişken uzunluklu dizginin bayt cinsinden görel konumudur. MQCHARV yapısı başka

bir yapıya yerleştirilmediğinde, örneğin, bir işlev çağrısında parametre olarak belirtilirse, görelî konum MQCHARV yapısının başlangıçla görelî olur.

Görelî konum pozitif ya da negatif olabilir. Değişken uzunluklu dizilimi belirlemek için VCHRP ya da VCHRO alanını kullanabilirsiniz, ancak ikisini birden belirleyemezsiniz.

Bu alanın ilk değeri 0 'dır.

VCHRP (gösterge)

Bu, değişken uzunluklu diziyi gösteren bir işaretçidir.

Değişken uzunluklu dizilimi belirlemek için VCHRP ya da VCHRO alanını kullanabilirsiniz, ancak ikisini birden belirleyemezsiniz.

Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

VCHRS (10 basamaklı işaretli tamsayı)

VCHRP ya da VCHRO alanı tarafından adreslenen arabelleğin bayt cinsinden büyüklüğü.

MQCHARV yapısı bir işlev çağrısında çıkış alanı olarak kullanıldığında, bu alan sağlanan arabelleğin uzunluğuyla kullanıma hazırlanmalıdır. VCHRL değeri VCHRS değerinden büyükse, arabellekteki çağırana yalnızca VCHRS veri baytları döndürülür.

Değer sıfırdan büyük ya da sıfıra eşit ya da tanınan şu özel değer olmalıdır:

VSUSL

VSUSL belirtilirse, arabelleğin uzunluğu MQCHARV yapısındaki VCHRL alanından alınır. Bu özel değer, yapı bir çıkış alanı olarak kullanıldığında ve bir arabellek sağlandığında uygun değerdir. Bu, bu alanın ilk değeridir.

Başlangıçtaki değerler

Çizelge 689. Değişmezler için MQCHARV başlangıç değerleri		
Alan adı	Değişmezin adı	Değişmezin değeri
VCHRP	Yok	Boş değerli gösterge ya da boş değerli byte 'lar.
VCHRO	Yok	0
VCHRS	VSUSL	-1
VCHRL	Yok	0
VCHRC	CSAPL	-3

RPG bildirimî

```
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
```

CSAPL ' nin yeniden tanımlanması

Diğer platformlarda desteklenen programlama dillerinden farklı olarak, RPG 'nin tanımlı bir sabiti yeniden tanımlama yöntemi yoktur; bu nedenle, CSAPL dışında bir değer kullanmak istiyorsanız, her bir VCHRC' yi özel olarak ayarlamamız gerekir.

IBM i IBM i üzerinde MÖCIH (CICS bridge üstbilgisi)

MÖCIH yapısı, CICS bridge ile IBM MQ for z/OSarasında gönderilen bir iletinin başlangıcında bulunabilecek bilgileri açıklar.

Genel Bakış

Biçim adı: FMCICS.

Sürüm: MÖCIH ' nin yürürlükteki sürümü CIVER2. Yapının yalnızca daha yeni sürümünde var olan alanlar, izleyen açıklamalarda bu şekilde tanımlanır.

Sağlanan COPY dosyası, CIVER alanının ilk değeri CIVER2olarak ayarlanmış MÖCIH ' nin en son sürümünü içerir.

Karakter kümesi ve kodlama: MÖCIH yapısı ve uygulama iletisi verileri için kullanılan karakter kümesi ve kodlama için özel koşullar geçerlidir:

- CICS bridge kuyruğunun sahibi olan kuyruk yöneticisine bağlanan uygulamaların, kuyruk yöneticisinin karakter kümesi ve kodlamasında bulunan bir MÖCIH yapısı sağlaması gerekir. Bunun nedeni, bu durumda MÖCIH yapısının veri dönüştürmesinin gerçekleştirilmemiş olmasıdır.
- Diğer kuyruk yöneticilerine bağlanan uygulamalar, desteklenen karakter kümelerinde ve kodlamasında bulunan bir MÖCIH yapısı sağlayabilir; MÖCIH ' nin dönüştürülmesi, CICS bridge kuyruğunun sahibi olan kuyruk yöneticisine bağlı olan alıcı ileti kanalı aracısı tarafından gerçekleştirilir.

Not: Bunun bir istisnası var. CICS bridge kuyruğunun sahibi olan kuyruk yöneticisi dağıtımlı kuyruğa alma için CICS kullanıyorsa, MÖCIH, CICS bridge kuyruğunun sahibi olan kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır.

- MÖCIH yapısını izleyen uygulama iletisi verileri, MÖCIH yapısıyla aynı karakter kümesinde ve kodlamada olmalıdır. MÖCIH yapısındaki CICS ve CIENC alanları, uygulama ileti verilerinin karakter kümesini ve kodlamasını belirtmek için kullanılamaz.

Veriler, kuyruk yöneticisi tarafından desteklenen yerleşik biçimlerden biri değilse, uygulama iletisi verilerini dönüştürmek için kullanıcı tarafından bir veri dönüştürme çıkışı sağlanmalıdır.

Kullanım: Uygulamanın gerektirdiği değerler Çizelge 691 sayfa 1027içinde gösterilen ilk değerlerle aynıysa ve köprü AUTH=LOCAL ya da AUTH=TANIMLA ile çalışıyorsa, MÖCIH yapısı iletiden çıkarılabilir. Diğer tüm durumlarda, yapı mevcut olmalıdır.

Köprü bir version-1 ya da version-2 MÖCIH yapısını kabul eder, ancak 3270 hareketleri için version-2 yapısı kullanılmalıdır.

Uygulama, "istek" olarak belgelenen alanların köprüye gönderilen iletide uygun değerlere sahip olmasını sağlamalıdır; bu alanlar köprüye girer.

"Yanıt" olarak belgelenen alanlar, köprünün uygulamaya gönderdiği yanıt iletisinde CICS bridge tarafından ayarlanır. CIRET, CIFNC, CICC, CIREAve CIAC alanlarında hata bilgileri döndürülür, ancak bunların tümü her durumda ayarlanmaz. Çizelge 690 sayfa 1018 , farklı CIRETdeğerleri için ayarlanan alanları gösterir.

Çizelge 690. MÖCIH yapısındaki hata bilgisi alanlarının içeriği				
CIRET	CIFNC	CICC	CIREA	CIAC
CRC000	-	-	-	-
CRC003	-	-	FBC*	-
CRC002 CRC008	IBM MQ arama adı	IBM MQ CMPCOD	IBM MQ REASON	-

Çizelge 690. MÇCIH yapısındaki hata bilgisi alanlarının içeriği (devamı var)				
CIRET	CIFNC	CICC	CIREA	CIAC
CRC001 CRC006 CRC007 CRC009	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	-
CRC004 CRC005	-	-	-	CICS KODU

- “Alanlar” sayfa 1019
- “Başlangıçtaki değerler” sayfa 1027
- “RPG bildirim” sayfa 1028

Alanlar

MÇCIH yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

CIAC (4 baytlık karakter dizisi)

Anormal kod.

Bu alanda döndürülen değer yalnızca *CIRET* alanında CRC005 ya da CRC004 değeri varsa önemlidir. Varsa, *CIAC* CICS ABCODE değerini içerir.

Bu bir yanıt alanıdır. Bu alanın uzunluğu LNABNC tarafından verilir. Bu alanın ilk değeri 4 boş karakterdir.

Bu, ADS tanımlayıcılarının SEND ve RECEIVE BMS isteklerinde gönderilip gönderilmeyeceğini belirten bir göstergedir. Aşağıdaki değerler tanımlanır:

ADYOK

ADS tanımlayıcısını göndermeyin ya da almayın.

ADSEND

ADS tanımlayıcısını gönderin.

ADRECV

ADS tanımlayıcısını alın.

ADMSGF

ADS tanımlayıcısı için ileti biçimini kullanın.

Bu, ADS tanımlayıcısının uzun biçimi kullanılarak gönderilmesine ya da alınmasına neden olur. Uzun form, 4 baytlık sınırlara hizalanmış alanlar içerir.

CIADS alanı aşağıdaki gibi ayarlanmalıdır:

- ADS tanımlayıcıları kullanılmıyorsa, alanı ADNONE olarak ayarlayın.
- ADS tanımlayıcıları *kullanılıyorsa* ve her ortamda *aynı* CCSID kullanılıyorsa, alanı ADSEND ve ADRECV toplamı olarak ayarlayın.
- ADS tanımlayıcıları *kullanılıyorsa* , ancak her ortamda *farklı* CCSID ' leri varsa, alanı ADSEND, ADRECV ve ADMSGF toplamına ayarlayın.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri ADNONE olur.

CIADS (10 basamaklı işaretli tamsayı)

ADS tanımlayıcısını gönder/al.

Bu, ADS tanımlayıcılarının SEND ve RECEIVE BMS isteklerinde gönderilip gönderilmeyeceğini belirten bir göstergedir. Aşağıdaki değerler tanımlanır:

ADYOK

ADS tanımlayıcısını göndermeyin ya da almayın.

ADSEND

ADS tanımlayıcısını gönderin.

ADRECV

ADS tanımlayıcısını alın.

ADMSGF

ADS tanımlayıcısı için ileti biçimini kullanın.

Bu, ADS tanımlayıcısının uzun biçimi kullanılarak gönderilmesine ya da alınmasına neden olur. Uzun form, 4 baytlık sınırlara hizalanmış alanlar içerir.

CIADS alanı aşağıdaki gibi ayarlanmalıdır:

- ADS tanımlayıcıları kullanılmıyorsa, alanı ADNONE olarak ayarlayın.
- ADS tanımlayıcıları *kullanılıyorsa* ve her ortamda *aynı* CCSID kullanılıyorsa, alanı ADSEND ve ADRECV toplamı olarak ayarlayın.
- ADS tanımlayıcıları *kullanılıyorsa* , ancak her ortamda *farklı* CCSID ' leri varsa, alanı ADSEND, ADRECV ve ADMSGF toplamına ayarlayın.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri ADNONE olur.

CIAI (4 baytlık karakter dizgisi)

AID anahtarı.

Bu, hareket başlatıldığında AID anahtarının ilk değeridir. 1 baytlık bir değerdir ve sola hizalanır.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın uzunluğu LNATID tarafından verilir. Bu alanın ilk değeri 4 boşluktur.

CIAUT (8 baytlık karakter dizgisi)

Parola ya da geçiş bileti.

Bu bir parola ya da geçiş biletidir. CICS bridge için kullanıcı kimliği kimlik doğrulaması etkinse, *CIAUT* , iletiyi gönderenin kimliğini doğrulamak için MQMD kimlik bağlamındaki kullanıcı kimliğiyle birlikte kullanılır.

Bu bir istek alanıdır. Bu alanın uzunluğu LNAUTH tarafından verilir. Bu alanın ilk değeri 8 boşluktur.

CICC (10 basamaklı işaretli tamsayı)

IBM MQ tamamlama kodu ya da CICS EIBRESP.

Bu alanda döndürülen değer *CIRET* ; bkz. [Çizelge 690 sayfa 1018](#).

Bu bir yanıt alanıdır. Bu alanın ilk değeri CCOK ' dur.

CICNC (4 baytlık karakter dizgisi)

Hareket kodunu olağandışı sonlandır.

Bu, hareketi sonlandırmak için kullanılacak olağandışı bitiş kodudur (normalde daha fazla veri isteyen bir etkileşimli hareket). Ters durumda, bu alan boş olarak ayarlanır.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın uzunluğu LNCNCL tarafından verilir. Bu alanın ilk değeri 4 boşluktur.

CICP (10 basamaklı işaretli tamsayı)

İmleç konumu.

Bu, hareket başlatıldığında ilk imleç konumudur. Daha sonra, etkileşimli hareketler için imleç konumu RECEIVE vektöründe olur.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri 0 'dır. *CIVER* , *CIVER2* değerinden küçükse bu alan yoktur.

CICSI (10 basamaklı işaretli tamsayı)

Ayrılmış.

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri 0 'dır.

CICT (10 basamaklı işaretli tamsayı)

Görevin sohbet amaçlı olup olmayacağını belirler.

Bu, görevin daha fazla bilgi için istekte bulunmasına izin verilip verilmeyeceğini ya da olağandışı sona erdirilip erdirilmeyeceğini belirten bir göstergedir. Değer aşağıdakilerden biri olmalıdır:

CTYES

Görev konuşmaktır.

CTNO

Görev etkileşimli değil.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri CTNO olur.

CIENC (10 basamaklı işaretli tamsayı)

Ayrılmış.

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri 0 'dır.

CIEO (10 basamaklı işaretli tamsayı)

İletideki hatanın görelî konumu.

Bu, köprü çıkışı tarafından saptanan geçersiz verilerin konumudur. Bu alan, iletinin başlangıcından geçersiz verilerin konumuna kadar olan görelî konumu sağlar.

Bu, yalnızca 3270 hareketleri için kullanılan bir yanıt alanıdır. Bu alanın ilk değeri 0 'dır. *CIVER* , *CIVER2* değerinden küçükse bu alan yoktur.

CIFAC (8 baytlık dize)

Köprü tesisi simgesi.

Bu, 8 baytlık bir köprü olanağı simgesidir. Bir köprü olanağı simgesinin amacı, bir sözde etkileşimde birden çok işlemin aynı köprü olanağını (sanal 3270 uçbirimi) kullanmasına izin vermektir. Bir sözde sohbetteki ilk ya da tek iletide FCNONE değeri ayarlanmalıdır; bu, CICS ' e bu ileti için yeni bir köprü olanağı ayırmasını bildirir. Giriş iletisinde sıfır dışında bir *CIFKT* belirtildiğinde yanıt iletilerinde bir köprü olanağı simgesi döndürülür. Sonraki giriş iletileri aynı köprü olanağı simgesini kullanabilir.

Aşağıdaki özel değer tanımlandı:

FCNONE

BVT simgesi belirtilmedi.

Bu alan, yalnızca 3270 hareketleri için kullanılan bir istek ve yanıt alanıdır. Bu alanın uzunluğu LNFAc tarafından verilir. Bu alanın ilk değeri FCNONE ' dır.

CIFKT (10 basamaklı işaretli tamsayı)

Köprü tesisi serbest bırakma zamanı.

Bu, kullanıcı hareketi sona erdikten sonra köprü olanağının alıkonacağı süreyi saniye cinsinden belirtir. Etkileşimli olmayan hareketler için değer sıfır olmalıdır.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın ilk değeri 0 'dır.

CIFL (4 baytlık karakter dizisi)

Uçbirim öykünülen öznelîkleri.

Bu, köprü olanağı için model olarak kullanılacak kurulu bir uçbirimin adıdır. Boşluk değeri, *CIFL* öğesinin köprü hareket profili tanımından alındığı ya da varsayılan bir değer kullanıldığı anlamına gelir.

Bu, yalnızca 3270 hareketleri için kullanılan bir istek alanıdır. Bu alanın uzunluğu LNFAcL tarafından verilir. Bu alanın ilk değeri 4 boşluktur.

CIFLG (10 basamaklı işaretli tamsayı)

Bayraklar.

Değer şöyle olmalıdır:

CIFNON

Bayrak yok.

Bu bir istek alanıdır. Bu alanın ilk değeri CIFNON ' dur.

CIFMT (8 baytlık karakter dizgisi)

IBM MQ MQCIH ' yi izleyen verilerin biçim adı.

Bu, MQCIH yapısını izleyen verilerin IBM MQ biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *MDFMT* alanıyla aynıdır.

CIRFM alanı FMNONE değerine sahipse, yanıt iletisi için bu biçim adı da kullanılır.

- DPL istekleri için, *CIFMT* COMMAREA ' nın biçim adı olmalıdır.
- 3270 istekleri için *CIFMT* CSQCBDCIve *CIRFM* CSQCBDCOolmalıdır.

Bu biçimlere ilişkin veri dönüştürme çıkışları, çalışacakları kuyruk yöneticisine kurulmalıdır.

İstek iletisi bir hata yanıt iletisi oluşturulmasına neden olursa, hata yanıt iletisinin biçim adı FMSTR olur.

Bu bir istek alanıdır. Bu alanın uzunluğu LNFMT tarafından verilir. Bu alanın ilk değeri FMNONE.

CIFNC (4 baytlık karakter dizgisi)

IBM MQ çağrı adı ya da CICS EIBFN işlevi.

Bu alanda döndürülen değer *CIRET* ; bkz. [Çizelge 690 sayfa 1018](#). *CIFNC* bir IBM MQ arama adı içerdiğinde aşağıdaki değerler kullanılabilir:

CFCONN

MQCONN çağrısı.

CFGET

MQGET çağrısı.

CFINQ (Soru)

MQINQ çağrısı.

CFOPEN

MQOPEN çağrısı.

CFPUT

MQPUT çağrısı.

CFPUT1

MQPUT1 çağrısı.

CFNONE

Telefon yok.

Bu bir yanıt alanıdır. Bu alanın uzunluğu LNFUNC tarafından verilir. Bu alanın ilk değeri CFNONE.

CIGWI (10 basamaklı işaretli tamsayı)

Köprü görevi tarafından verilen MQGET çağrısıyla ilgili bekleme aralığı.

Bu alan yalnızca *CIUOW* CUFIRST değerine sahip olduğunda geçerlidir. Gönderen uygulamanın, köprü tarafından yayınlanan MQGET çağrılarının bu ileti tarafından başlatılan iş birimine ilişkin ikinci ve sonraki istek iletilerini beklemesi gereken yaklaşık süreyi milisaniye cinsinden belirtmesini sağlar. Bu, köprü tarafından kullanılan varsayılan bekleme aralığını geçersiz kılar. Aşağıdaki özel değerler kullanılabilir:

WIDFLT

Varsayılan bekleme aralığı.

Bu, CICS bridge ' in köprünün başlatıldığı zaman belirtilen süreyi beklemesine neden olur.

WIULIM

Sınırsız bekleme aralığı.

Bu bir istek alanıdır. Bu alanın ilk değeri WIDFLT 'dir.

CIII (10 basamaklı işaretli tamsayı)

Ayrılmış.

Bu ayrılmış bir alandır. Değer 0 olmalıdır. *CIVER* , *CIVER2* değerinden küçükse bu alan yoktur.

CILEN (10 basamaklı işaretli tamsayı)

MQCIH yapısının uzunluğu.

Değer aşağıdakilerden biri olmalıdır:

CILEN1

version-1 CICS bilgi üstbilgisi yapısının uzunluğu.

CILEN2

version-2 CICS bilgi üstbilgisi yapısının uzunluğu.

Aşağıdaki değişmez, yürürlükteki sürümün uzunluğunu belirtir:

CILENC

CICS bilgi üstbilgisi yapısının geçerli sürümünün uzunluğu.

Bu bir istek alanıdır. Bu alanın ilk değeri CILEN2' dir.

ELIK (10 basamaklı işaretli tamsayı)

Bağlantı tipi.

Bu, köprünün bağlanmayı denemesi gereken nesne tipini gösterir. Değer aşağıdakilerden biri olmalıdır:

LTPROG

DPL programı.

LTTRAN

3270 hareketi.

Bu bir istek alanıdır. Bu alanın ilk değeri LTPROG değeridir.

CINTI (4 baytlık karakter dizgisi)

Eklenecek sonraki işlem.

Bu, kullanıcı hareketi tarafından döndürülen sonraki hareketin adıdır (genellikle EXEC CICS RETURN TRANSID tarafından). Sonraki hareket yoksa, bu alan boş olarak ayarlanır.

Bu, yalnızca 3270 hareketleri için kullanılan bir yanıt alanıdır. Bu alanın uzunluğu LNTRID tarafından verilir. Bu alanın ilk değeri 4 boşluktur.

CIODL (10 basamaklı işaretli tamsayı)

Çıkış COMMAREA veri uzunluğu.

Bu, yanıt iletilinde istemciye döndürülecek kullanıcı verilerinin uzunluğudur. Bu uzunluk, 8 baytlık program adını içerir. Bağlı programa geçirilen COMMAREA ' nın uzunluğu, bu alanın üst sınırı ve istek iletilisindeki kullanıcı verilerinin uzunluğu eksi 8 'dir.

Not: Bir iletideki kullanıcı verilerinin uzunluğu, MQCIH yapısı *hariç* iletinin uzunluğudur.

İstek iletilisindeki kullanıcı verilerinin uzunluğu *CIODL* değerinden küçükse, LINK komutunun DATALENGTH seçeneği kullanılır; bu, LINK işlevinin başka bir CICS bölgesine verimli bir şekilde gönderilmesini sağlar.

Aşağıdaki özel değer kullanılabilir:

OLINPT

Çıkış uzunluğu, giriş uzunluğuyla aynı.

Bağlantı oluşturulan programa geçirilen COMMAREA 'nın yeterli büyüklükte olmasını sağlamak için herhangi bir yanıt istenmese de bu değer gerekli olabilir.

Bu, yalnızca DPL programları için kullanılan bir istek alanıdır. Bu OLINPT alanının başlangıç değeri.

CIREA (10 basamaklı işaretli tamsayı)

IBM MQ neden ya da geribildirim kodu ya da CICS EIBRESP2.

Bu alanda döndürülen değer *CIRET* ; bkz. [Çizelge 690 sayfa 1018](#).

Bu bir yanıt alanıdır. Bu alanın ilk değeri RCNONE.

CIRET (10 basamaklı işaretli tamsayı)

Köprüden dönüş kodu.

Bu, köprü tarafından gerçekleştirilen işlemlerin sonucunu açıklayan CICS bridge dönüş kodudur. *CIFNC*, *CICC*, *CIREAve CIAC* alanları ek bilgi içerebilir (bkz. [Çizelge 690 sayfa 1018](#)). Değer aşağıdakilerden biridir:

CRC000

(0, X'000 ') Hata yok.

CRC001

(1, X'001 ') EXEC CICS deyimi bir hata saptadı.

CRC002

(2, X'002 ') IBM MQ çağırısı bir hata saptadı.

CRC003

(3, X'003 ') CICS bridge bir hata algıladı.

CRC004

(4, X'004 ') CICS bridge olağandışı bitti.

CRC005

(5, X'005 ') Uygulama olağandışı bitti.

CRC006

(6, X'006 ') Güvenlik hatası oluştu.

CRC007

(7, X'007 ') Program kullanılmıyor.

CRC008

(8, X'008 ') Yürürlükteki iş birimi içindeki ikinci ya da sonraki ileti, belirtilen süre içinde alınmadı.

CRC009

(9, X'009 ') İşlem kullanılmıyor.

Bu bir yanıt alanıdır. Bu alanın ilk değeri CRC000' dir.

CIRFM (8 baytlık karakter dizgisi)

IBM MQ yanıt iletilisinin biçim adı.

Bu, yürürlükteki iletiye yanıt olarak gönderilecek yanıt iletilisinin IBM MQ biçim adıdır. Bu kodu kodlamaya ilişkin kurallar, MQMD 'deki *MDFMT* alanıyla aynıdır.

Bu, yalnızca DPL programları için kullanılan bir istek alanıdır. Bu alanın uzunluğu LNFMT tarafından verilir. Bu alanın ilk değeri FMNONE.

CIRSI (4 baytlık karakter dizgisi)

Ayrılmış.

Bu ayrılmış bir alandır. Değer 4 boşluk olmalıdır. Bu alanın uzunluğu LNRSID tarafından verilir.

CIRS1 (8 baytlık karakter dizgisi)

Ayrılmış.

Bu ayrılmış bir alandır. Değer 8 boşluk olmalıdır.

CIRS2 (8 baytlık karakter dizgisi)

Ayrılmış.

Bu ayrılmış bir alandır. Değer 8 boşluk olmalıdır.

CIRS3 (8 baytlık karakter dizgisi)

Ayrılmış.

Bu ayrılmış bir alandır. Değer 8 boşluk olmalıdır.

CIRS4 (10 basamaklı işaretli tamsayı)

Ayrılmış.

Bu ayrılmış bir alandır. Değer 0 olmalıdır. *CIVER* , *CIVER2* değerinden küçükse bu alan yoktur.

CIRTI (4 baytlık karakter dizgisi)

Ayrılmış.

Bu ayrılmış bir alandır. Değer 4 boşluk olmalıdır. Bu alanın uzunluğu LNTRID tarafından verilir.

CISC (4 baytlık karakter dizilimi)

İşlem başlatma kodu.

Bu, köprünün bir uçbirim hareketine öykünme mi, yoksa bir START hareketine öykünme mi gerçekleştirdiğini belirten bir göstergedir. Değer aşağıdakilerden biri olmalıdır:

SCSTRT

Başlatma.

SCDATA

Verileri başlatın.

SCTERM

Girişi sonlandırın.

SCNONE

Yok.

Köprüden gelen yanıtta bu alan, *CINTI* alanında bulunan sonraki hareket tanıtıcısına uygun başlangıç koduna ayarlanır. Yanıtta aşağıdaki başlatma kodları olabilir:

- SCSTRT
- SCDATA
- SCTERM

CICS Transaction Server 1.2 için bu alan yalnızca bir istek alanıdır; yanıtta değeri tanımsız olur.

CICS Transaction Server 1.3 ve sonraki yayınlar için bu hem bir istek hem de bir yanıt alanıdır.

Bu alan yalnızca 3270 hareketleri için kullanılır. Bu alanın uzunluğu LNSTCO tarafından verilir. Bu alanın ilk değeri SCNONE.

CISID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

CISIDV

CICS bilgi üstbilgisi yapısına ilişkin tanıtıcı.

Bu bir istek alanıdır. Bu alanın ilk değeri CISIDV 'dir.

CITES (10 basamaklı işaretli tamsayı)

Görevin sonundaki durum.

Bu alan, görev sonunda kullanıcı işleminin durumunu gösterir. Aşağıdaki değerlerden biri döndürülür:

TENOSY

Senkronize değil.

Kullanıcı işlemi henüz tamamlanmadı ve eşitlenmedi. MQMD 'deki *MDMT* alanı bu durumda *MTRQST* 'dir.

TECMIT

Kesinleştirme iş birimi.

Kullanıcı hareketi henüz tamamlanmadı, ancak ilk iş birimini eşitlendi. MQMD 'deki *MDMT* alanı *MTDGRM* 'dir.

GERI Dönüş

Geri çekilin.

Kullanıcı hareketi henüz tamamlanmadı. Yürürlükteki iş birimi geriletilecek. MQMD 'deki *MDMT* alanı *MTDGRM* 'dir.

TEDAT

Görevi sona erdir.

Kullanıcı hareketi sona erdi (ya da olağandışı bitti). MQMD 'deki *MDMT* alanı bu durumda *MTRPLY* 'dir.

Bu, yalnızca 3270 hareketleri için kullanılan bir yanıt alanıdır. Bu alanın ilk değeri *TENOSY* 'dir.

CITI (4 baytlık karakter dizgisi)

Eklenecek işlem.

CILT *LTTRAN* değerine sahipse, *CITI* çalıştırılacak kullanıcı işleminin işlem tanıtıcısıdır; bu durumda boş olmayan bir değer belirtilmelidir.

CILT *LTPROG* değerine sahipse, *CITI* iş birimi içindeki tüm programların çalıştırılacağı hareket kodudur. Belirtilen değer boşsa, *CICS* *DPL* köprüsü varsayılan işlem kodu (*CKBP*) kullanılır. Değer boş değilse, *CICS* için *CSQCBP00* başlangıç programıyla birlikte yerel bir *TRANSACTION* (hareket) olarak tanımlanmış olmalıdır. Bu alan yalnızca *CIUOW* , *CUFRST* ya da *CUONLY* değerine sahip olduğunda geçerlidir.

Bu bir istek alanıdır. Bu alanın uzunluğu *LNTRID* tarafından verilir. Bu alanın ilk değeri 4 boşluktur.

CIUOW (10 basamaklı işaretli tamsayı)

İş birimi kontrolü.

Bu, *CICS* bridgetarafından gerçekleştirilen iş birimi işlemlerini denetler. Köprüden tek bir hareket ya da bir iş birimi içinde bir ya da daha çok program çalıştırmasını isteyebilirsiniz. Bu alan, *CICS* *bridge* 'in bir iş birimi başlatması, yürürlükteki iş birimi içinde istenen işlevi gerçekleştirmesi ya da iş birimini kesinleştirerek ya da yedekleyerek sona erdirmesi gerektiğini gösterir. Veri iletim akışlarının eniyilenmesi için çeşitli birleşimler desteklenir.

Değer aşağıdakilerden biri olmalıdır:

YALNIZCA

İş birimini başlatın, işlev gerçekleştirin ve iş birimini kesinleştirin (*DPL* ve 3270).

KüCONT

Yürürlükteki iş birimine ilişkin ek veriler (yalnızca 3270).

CUFRST (KESME)

İş birimini başlat ve işlevi gerçekleştir (yalnızca *DPL*).

CUMIDL

İşlevi yürürlükteki iş birimi içinde gerçekleştir (yalnızca *DPL*).

SON

İşlevi gerçekleştirin ve iş birimini kesinleştirin (yalnızca *DPL*).

CUCMIT.

İş birimini kesinleştirin (yalnızca *DPL*).

KÜME

İş birimini geri çevirin (yalnızca DPL).

Bu bir istek alanıdır. Bu alanın ilk değeri CUONLY 'dir.

CIVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer aşağıdakilerden biri olmalıdır:

CIVER1

Version-1 CICS bilgi üstbilgisi yapısı.

CIVER2

Version-2 CICS bilgi üstbilgisi yapısı.

Yalnızca yapının daha yeni sürümünde var olan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

CIVERC

CICS bilgi üstbilgisi yapısının geçerli sürümü.

Bu bir istek alanıdır. Bu alanın ilk değeri CIVER2' dir.

Başlangıçtaki değerler

Çizelge 691. MÇCIH ' deki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
CISID	CISIDV	'CIH→'
CIVER	CIVER2	2
CILEN	CILEN2	180
CIENC	Yok	0
CICSI	Yok	0
CIFMT	FMNONE	Boşluklar
CIFLG	CIFNON	0
CIRET	CRC000	0
CICC	CCOK	0
CIREA	RCNONE	0
CIUOW	YALNIZCA	273
CIGWI	WIDFLT	-2
CILT	LTPROG	1
CIODL	OLINPT	-1
CIFKT	Yok	0
CIADS	ADYOK	0
CICT	CTNO	0
CITES	TENOSY	0
CIFAC	FCNONE	Boş Değerler
CIFNC	CFNONE	Boşluklar

Çizelge 691. MÇCIH 'deki Alanlar (devamı var)		
Alan adı	Değişmezin adı	Değişmezin değeri
CIAC	Yok	Boşluklar
CIAUT	Yok	Boşluklar
CIRS1	Yok	Boşluklar
CIRFM	FMNONE	Boşluklar
CIRSI	Yok	Boşluklar
CIRTI	Yok	Boşluklar
CITI	Yok	Boşluklar
CIFL	Yok	Boşluklar
CIAI	Yok	Boşluklar
CISC	SCNONE	Boşluklar
CICNC	Yok	Boşluklar
CINTI	Yok	Boşluklar
CIRS2	Yok	Boşluklar
CIRS3	Yok	Boşluklar
CICP	Yok	0
CIE0	Yok	0
CIII	Yok	0
CIRS4	Yok	0

Notlar:

1. - simgesi, tek bir boş karakteri temsil eder.

RPG bildirimi

```

D* .1.....2.....3.....4.....5.....6.....7..
D* MÇCIH 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 MÇCIH 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 followsMÇCIH
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 be conversational
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'0000000000000000-00')
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*

```

IBM i

IBM i üzerinde MQCMHO (İleti tanıtıcı seçenekleri yarat)

MQCMHO yapısı, uygulamaların ileti tanıtıcılarının nasıl yaratılacağını denetleyen seçenekleri belirtmesine olanak sağlar.

Genel Bakış

Amaç

Yapı, **MQCRTMH** çağrısındaki bir giriş parametresidir.

Karakter kümesi ve kodlama

MQCMHO içindeki veriler, uygulamanın karakter kümesinde ve uygulamanın kodlamasında (ENNAT) olmalıdır.

- [“Alanlar” sayfa 1030](#)
- [“Başlangıçtaki değerler” sayfa 1031](#)
- [“RPG bildirimi” sayfa 1031](#)

Alanlar

MQCMHO yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

CMOPT (10 basamaklı işaretli tamsayı)

Aşağıdaki seçeneklerden biri belirtilebilir:

CMVAL

Bu ileti tanıtıcısında bir özellik ayarlamak için **MQSETMP** çağrıldığında, özellik adının geçerliliği denetlenir ve aşağıdakilerden emin olur:

- geçersiz karakter içermiyor.
- Aşağıdakiler dışında "JMS" ya da "usr.JMS" başlamaz:
 - JMSCorrelationID
 - JMSReplyTo
 - JMSType
 - JMSXGroupID
 - JMSXGroupSeq

Bu adlar JMS özellikleri için ayrılmıştır.

- büyük ya da küçük harflerin herhangi bir karışımında aşağıdaki anahtar sözcüklerden biri değildir:
 - "VE"
 - "ARASINDA"
 - "KAÇIŞ"
 - "YANLIŞ"
 - "İÇERİ"
 - "OLDUĞU"
 - "BEĞEN"
 - "DEĞİL"
 - "BOŞ DEĞER"
 - "YA DA"
 - "DOĞRU"
- "Body" başlamaz. Ya da "Kök". ("Root.MQMD." dışında).

Özellik MQ-defined ("mq. *") ise Ad tanınır ve özellik tanımlayıcı alanları, özellik için doğru değerlere ayarlanır. Özellik tanınmazsa, özellik tanımlayıcısının *Support* alanı **PDSUPO** olarak ayarlanır (daha fazla bilgi için bkz. [PDSUP](#)).

CMDEFV

Bu, özellik adlarının varsayılan geçerlilik denetimi düzeyinin oluştuğunu belirtir.

Varsayılan geçerlilik denetimi düzeyi, **CMVAL** tarafından belirtilenle eşdeğerdir.

İleride yayınlanacak bir yayında, **CMDEFV** tanımlandığında oluşacak doğrulama düzeyini değiştirecek bir denetim seçeneği tanımlanabilir.

Bu varsayılan değerdir.

CMNOVA

Özellik adı üzerinde geçerlilik denetimi gerçekleştirilmez. **CMVAL** açıklamasına bakın.

Varsayılan seçenek: Bu bölümde daha önce açıklanan seçeneklerden hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

CMNONE

Tüm seçenekler varsayılan değerlerini varsayar. Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın. **CMNONE** program belgelerine yardımcı olur; bu seçeneğin başka bir değerle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **CMDEFV**.

CMSID (10 basamaklı işaretli tamsayı)

Bu yapı tanıtıcısıdır; değer şu olmalıdır:

CMSIDV

İleti tanıtıcısı yaratma seçenekleri yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **CMSIDV**.

CMVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

CMVER1

Version-1 ileti tanıtıcı seçenekleri yapısı yaratır.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

CMVERC

İleti tanıtıcısı yaratma seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **CMVER1**.

Başlangıçtaki değerler

Çizelge 692. MÖCMHO içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
CMSID	CMSIDV	'CMHO'
CMVER	CMVER1	1
CMOPT	CMDEFV	0

RPG bildirimi

```
D* MÖCMHO 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 MÖCRTMH
D CMOPT          9     12I 0 INZ(0)
```

IBM i üzerinde MÖCNO (Bağlantı seçenekleri)

MÖCNO yapısı, uygulamanın yerel kuyruk yöneticisine yönelik bağlantıyla ilgili seçenekleri belirtmesine olanak sağlar.

Genel Bakış

Amaç: Yapı, MQCONNX çağrısındaki bir giriş/çıkış deęiřtirgesidir.

Sürüm: MQCNO ' nun yürürlükteki sürümü CNVER6. Yapının yalnızca daha yeni sürümlerinde bulunan alanlar, izleyen açıklamalarda bu şekilde tanımlanır.

Saęlanan COPY dosyası, ortam tarafından desteklenen en son MQCNO sürümünü içeriyor; ancak, CNVER alanının ilk deęeri CNVER1olarak ayarlanmış. version-1 yapısında var olmayan alanları kullanmak için uygulamanın CNVER alanını gerekli sürümün sürüm numarasına ayarlaması gerekir.

Karakter kümesi ve kodlama: MQCNO içindeki veriler, **CodedCharSetId** kuyruk yöneticisi öznitelięi ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır.

- “Alanlar” sayfa 1032
- “Bařlangıçtaki deęerler” sayfa 1037
- “RPG bildirimini” sayfa 1038

Alanlar

MQCNO yapısı ařaęıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

CCDTUL (10 basamaklı iřaretli tamsayı)

CCDTUL, baęlantı için kullanılacak istemci baęlantı kanalı çizelgesinin konumunu tanımlayan bir URL içeren CCDTUP ya da CCDTUO tarafından tanımlanan dizginin uzunluęudur.

CCDTUL ' yi yalnızca MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI clientolarak çalıştıęında kullanın.

Bu, [MQCHLLIB](#) ve [MQCHLTAB](#) ortam deęiřkenlerini ayarlamaya yönelik programlı bir alternatiftir.

Uygulama istemci olarak çalışmıyorsa, CCDTUL yoksayılr.

CNVER CNVER6deęerinden küçükse bu alan yoksayılr.

CCDTUO (10 basamaklı iřaretli tamsayı)

CCDTUO, MQCNO yapısının bařlangıcından baęlantı için kullanılacak istemci baęlantı kanalı çizelgesinin konumunu tanımlayan URL içeren bir dizgiye kadar bayt cinsinden görelili konumdur. Görelili konum pozitif ya da negatif olabilir.

CCDTUL ' yi yalnızca MQCONNX çağrılarını yayınlayan uygulama IBM MQ MQI clientolarak çalıştıęında kullanın.

Önemli: CCDTUP ve CCDTUO ' dan yalnızca birini kullanabilirsiniz. Her iki alan da sıfır deęilse, çağrı RC2600 neden koduyla başarısız olur.

Bu, [MQCHLLIB](#) ve [MQCHLTAB](#) ortam deęiřkenlerini ayarlamaya yönelik programlı bir alternatiftir.

Uygulama istemci olarak çalışmıyorsa, CCDTUO yoksayılr.

CNVER CNVER6deęerinden küçükse bu alan yoksayılr.

CCDTUP (gösterge)

CCDTUP, baęlantı için kullanılacak istemci baęlantı kanalı çizelgesinin yerini saptamak için, URLiçeren bir dizgiye iliřkin isteęe baęlı bir göstergedir.

CCDTUP özellięini, MQCONNX çağrısı yayınlayan uygulama IBM MQ MQI clientolarak çalıştıęında kullanın.

Önemli: CCDTUP ve CCDTUO ' dan yalnızca birini kullanabilirsiniz. Her iki alan da sıfır deęilse, çağrı RC2600 neden koduyla başarısız olur.

Bu, [MQCHLLIB](#) ve [MQCHLTAB](#) ortam deęiřkenlerini ayarlamaya yönelik programlı bir alternatiftir.

Uygulama istemci olarak çalışmıyorsa, CCDTUP yoksayılr.

CNVER CNVER6değerinden küçükse bu alan yoksayılır.

CNAN (28 baytlık karakter dizgisi)

Kuyruk yöneticisiyle bağlantıyı tanımlamak için uygulama tarafından ayarlanan ad. Alanın ilk değeri boş karakterdir.

CNVER CNVER7değerinden küçükse bu alan yoksayılır.

CNCCO (10 basamaklı işaretli tamsayı)

MQCNO yapısının başlangıcından itibaren bir MQCD kanal tanımlaması yapısının byte cinsinden görelî konumudur.

CNCCP (gösterge)

Bu, bir MQCD kanal tanımlaması yapısına ilişkin bir işaretçidir.

CNCONID (24 baytlık karakter dizgisi)

Benzersiz bağlantı tanıtıcısı. Bu alan, kuyruk yöneticisinin kuyruk yöneticisine ilk bağlandığında benzersiz bir tanıtıcı atayarak bir uygulama işlemini güvenilir bir şekilde tanımasını sağlar.

Uygulamalar PUT ve GET çağrılarını yaparken ilişkilendirme amacıyla bağlantı tanıtıcısını kullanır. Bağlantı nasıl kurulduğu fark etmez, tüm bağlantılara kuyruk yöneticisi tarafından bir tanıtıcı atanır.

Uzun süreli bir iş biriminin sona ermesini zorlamak için bağlantı tanıtıcısı kullanılabilir. Bunu yapmak için, 'Bağlantıyı Durdur' PCF komutunu ya da STOP CONN MQSC komutunu kullanarak bağlantı tanıtıcısını belirtin. Bu komutların kullanılmasıyla ilgili ek bilgi için ilgili bağlantılara bakın.

Alanın ilk değeri 24 boş bayttır.

CNCT (128 baytlık bit dizgisi)

Bu, kuyruk yöneticisinin bu bağlantı sırasında uygulamadan etkilenen kaynaklarla ilişkilendirdiği bir etikettir.

Kuyruk yöneticisi bağlantı etiketi.

Kuyruk yöneticisinin etkilenen kaynaklara erişimi doğru şekilde diziselleştirebilmesi için, her uygulama ya da uygulama yönetim ortamının etiket için farklı bir değer kullanması gerekir. Daha fazla ayrıntı için CN* CT* seçeneklerinin açıklamalarına bakın. Uygulama sona erdiğinde ya da MQDISC çağrısı verildiğinde etiket geçerli olmaktan çıkar.

Etiket gerekmiyorsa aşağıdaki özel değeri kullanın:

CTNONE.

Bağlantı etiketi belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNCTAG tarafından verilir. Bu alanın ilk değeri CTNONE. CNVER CNVER3değerinden küçükse bu alan yoksayılır.

Bir z/OS kuyruk yöneticisine bağlanırken ConnTag alanını kullanın.

CNNORES2 (4 baytlık karakter dizgisi)

Yapıyı 64 bitlik bir sınıra çıkarmak için ayrılmış bir alan. Alanın ilk değeri, alanın uzunluğu için ikili sıfırdır.

CNVER CNVER7değerinden küçükse bu alan yoksayılır.

CNOPT (10 basamaklı işaretli tamsayı)

MQCONNXylemini denetleyen seçenekler.

Bağ tanımlama seçenekleri

Bağlama seçenekleri, kullanılan IBM MQ bağ tanımının tipini denetler; bu seçeneklerden yalnızca birini belirtin:

CNSBND

Standart bağlama.

Standart bağlama seçeneği, uygulamanın ve yerel kuyruk yöneticisi aracısının genellikle ayrı işlemlerde ayrı yürütme birimlerinde çalışmasına neden olur. Düzenleme, kuyruk yöneticisinin bütünlüğünü korur; yani, kuyruk yöneticisini hatalı programlardan korur.

Uygulamanın tam olarak sinanmamış olabileceği ya da güvenilir ya da güvenilir olmayabileceği durumlarda CNSBND komutunu kullanın. CNSBND varsayılan değerdir.

CNSBND , program belgelerine yardımcı olmak için tanımlanır. Bu seçeneği, kullanılan bağ tanımının tipini denetleyen başka bir seçenikle kullanmayın; ancak değeri sıfır olduğu için, bu tür bir kullanım algılanamaz.

Bu seçenek tüm ortamlarda desteklenir.

CNFBND

Hızlı yol bağlama.

Hızlı yol bağlama seçeneği, uygulamanın ve yerel kuyruk yöneticisi aracısının aynı yürütme biriminin bir parçası olmasına neden olur. Hızlı yol, uygulamanın ve yerel kuyruk yöneticisi aracısının ayrı yürütme birimlerinde çalıştığı standart bağ tanımından farklıdır.

Kuyruk yöneticisi bu tip bir bağ tanımını desteklemiyorsa CNFBND yoksayılr; işlem, seçenek belirtilmemiş gibi devam eder.

CNFBND , birden çok sürecin uygulama tarafından kullanılan genel kaynaktan daha fazla kaynak tükettiği durumlarda avantajlı olabilir.Hızlı yol bağ tanımını kullanan bir uygulama, *güvenilir uygulama* olarak bilinir.

Hızlı yol bağlamanın kullanılıp kullanılmayacağına karar verirken aşağıdaki önemli noktaları göz önünde bulundurun:

- **CNFBND seçeneğinin kullanılması, bir uygulamanın iletileri ve kuyruk yöneticisine ait diğer veri alanlarını değiştirmesini ya da bozmasını önlemez. Bu seçeneği yalnızca bu sorunları tam olarak değerlendirdiğiniz durumlarda kullanın.**
- Uygulama, CNFBND ile zamanuyumsuz sinyaller ya da süreölçer kesintileri (sigkillgibi) kullanmamalıdır. Paylaşılan bellek parçalarının kullanımına ilişkin kısıtlamalar da vardır.
- Uygulamada, kuyruk yöneticisine aynı anda birden çok iş parçacığı bağlı olmamalıdır.
- Uygulamanın kuyruk yöneticisiyle bağlantısını kesmek için MQDISC çağrısı kullanması gerekir.
- Uygulama, kuyruk yöneticisini endmqm komutuyla sona erdirmeden önce bitmelidir.

Aşağıdaki noktalar, belirtilen ortamlarda CNFBND kullanımı için geçerlidir:

- IBM sistemlerinde iş, QMQMADM grubuna ait QMQM kullanıcı tanıtımı altında çalıştırılmalıdır. Ayrıca, program olağandışı sonlanmamalıdır, aksi takdirde beklenmedik sonuçlar ortaya çıkabilir.

Güvenilir uygulamaların kullanılmasının etkilerine ilişkin ek bilgi için [MQCONN](#) çağrısı kullanılarak kuyruk yöneticisine bağlanma ve [Güvenilen uygulamalara ilişkin kısıtlamalar](#) başlıklı konuya bakın.

CNSHBD

Paylaşılan Bağ Tanımları.

Paylaşılan bağ tanımları seçeneği, uygulamanın ve yerel kuyruk yöneticisi aracısının genellikle ayrı işlemlerde ayrı yürütme birimlerinde çalışmasına neden olur. Düzenleme, kuyruk yöneticisinin bütünlüğünü korur; yani, kuyruk yöneticisini hatalı programlardan korur. Ancak, bazı kaynaklar uygulama ile yerel kuyruk yöneticisi aracısı arasında paylaşılır. Kuyruk yöneticisi bu tip bir bağ tanımını desteklemiyorsa CNSHBD yoksayılr. İşlem, seçenek belirtilmemiş gibi devam eder.

CNIBND

Yalıtılmış Bağ Tanımları.

Yalıtılmış bağ tanımları seçeneği, uygulamanın ve yerel kuyruk yöneticisi aracısının genellikle ayrı işlemlerde ayrı yürütme birimlerinde çalışmasına neden olur. Düzenleme, kuyruk yöneticisinin bütünlüğünü korur; yani, kuyruk yöneticisini hatalı programlardan korur. Uygulama süreci ve yerel kuyruk yöneticisi aracı, kaynakları paylaşmadıkları için birbirinden yalıtılır. Kuyruk yöneticisi bu tip bir bağ tanımını desteklemiyorsa, CNIBND yoksayılr. İşlem, seçenek belirtilmemiş gibi devam eder.

İşle paylaşma seçenekleri

Aşağıdaki seçenekler, aynı işlem içindeki farklı iş parçacıkları (koşut işleme birimleri) arasındaki çekme noktalarının paylaşılmasını denetler. Bu seçeneklerden yalnızca biri belirtilebilir.

CNHSN

İş parçacıkları arasında tanıtıcı paylaşımı yok.

İş parçacıkları arasındaki tanıtıcı paylaşımı yok seçeneği, bağlantı ve nesne tanıtıcılarının yalnızca tutamacın ayrılmasına neden olan iş parçacığı tarafından kullanılabilceğini; diğer bir deyişle, MQCONN, MQCONNXya da MQOPEN çağrısını yayınlayan iş parçacığı tarafından kullanılabilceğini belirtir. Tutamaçlar, aynı işleme ait diğer iş parçacıkları tarafından kullanılamaz.

CNHSB

Arama bloklamasıyla, iş parçacıkları arasında dizisel tutamaç paylaşımı.

İş parçacıkları arasındaki dizisel tutamaç paylaşımı (çağrı engelleme ile) seçeneği, bir işlemin bir iş parçacığı tarafından ayrılan bağlantı ve nesne tanıtıcılarının aynı işleme ait diğer iş parçacıkları tarafından kullanılabilceğini gösterir. Ancak, aynı anda yalnızca bir iş parçacığı belirli bir tutamacı kullanabilir; yani, yalnızca bir tutamacın dizisel kullanımına izin verilir. Bir iş parçacığı, başka bir iş parçacığı tarafından kullanılmakta olan bir tanıtıcıyı kullanmayı denerse, çağrı blokları (tutamaç kullanılabilir oluncaya kadar bekler).

CNHSNB

Arama engellemesi olmadan, iş parçacıkları arasında dizisel tutamaç paylaşımı.

Arama engellemesi olmadan, iş parçacıkları arasındaki dizisel tutamaç paylaşımı seçeneği, " ile blocking " seçeneği dışında, tutamaç başka bir iş parçacığı tarafından kullanılırsa, çağrı tutamaç kullanılabilir oluncaya kadar engelleme yerine CCFAIL ve RC2219 ile hemen tamamlanır.

Bir iş parçacığının sıfır ya da bir paylaşılmayan tanıtıcısı ve sıfır ya da daha fazla paylaşılan tanıtıcısı olabilir:

- CNHSN belirten her bir MQCONN ya da MQCONNX çağrısı, ilk çağrıda yeni bir paylaşılmayan tanıtıcı ve sonraki çağrılarda aynı paylaşılmayan tanıtıcı döndürür (MQDISC çağrısını müdahalede bulunmadığı varsayılarak). Neden kodu, ikinci ve sonraki çağrılar için RC2002 ' dir.
- CNHSB ya da CNHSNB belirten her bir MQCONNX çağrısı, her çağrıda yeni bir paylaşılan tanıtıcı döndürür.

Nesne tanıtıcıları, nesne tanıtıcısını yaratan MQOPEN çağrısında belirtilen bağlantı tanıtıcısıyla aynı paylaşım özelliklerini devralır. Ayrıca, iş birimleri, iş birimini başlatmak için kullanılan bağlantı tanıtıcısıyla aynı paylaşım özelliklerini devralır; iş birimi, paylaşılan bir tanıtıcı kullanılarak bir iş parçacığında başlatılırsa, iş birimi aynı tanıtıcı kullanılarak başka bir iş parçacığında güncellenebilir.

Bir tanıtıcı paylaşım seçeneği belirlemezseniz, varsayılan değer ortam tarafından belirlenir:

- Microsoft Transaction Server (MTS) ortamında varsayılan değer CNHSBile aynıdır.
- Diğer ortamlarda, varsayılan değer CNHSNile aynıdır.

Yeniden bağlanma seçenekleri

Yeniden bağlanma seçenekleri, bir bağlantının yeniden bağlanabilir olup olmadığını belirler. Yalnızca istemci bağlantıları yeniden bağlanabilir.

CNRCDF

Yeniden bağlanma seçeneği varsayılan değerine çözülür. Varsayılan değer ayarlanmazsa, bu seçeneğin değeri DEVRE DIŞI olarak çözülür. Seçeneğin değeri sunucuya geçirilir ve **PCF** ve **MQSC** tarafından sorgulanabilir.

CNRC

Uygulama, MQCONNX **QMNAME** değiştirgesinin değeriyle tutarlı herhangi bir kuyruk yöneticisine yeniden bağlanabilir. CNRC seçeneğini yalnızca istemci uygulaması ile başlangıçta bağlantı kurduğu kuyruk yöneticisi arasında benzerlik yoksa kullanın. Seçeneğin değeri sunucuya geçirilir ve **PCF** ve **MQSC** tarafından sorgulanabilir.

CNRC D

Uygulama yeniden bağlanamıyor. Seçeneğin değeri sunucuya iletilmedi.

CNRCQM

Uygulama yalnızca, başlangıçta bağlandığı kuyruk yöneticisine yeniden bağlanabilir. Bir istemci yeniden bağlanabiliyorsa, ancak istemci uygulaması ile başlangıçta bağlantı kurduğu kuyruk yöneticisi arasında bir benzerlik varsa bu değeri kullanın. Bir istemcinin yüksek kullanılabilirlikli bir kuyruk yöneticisinin yedek yönetim ortamına otomatik olarak yeniden bağlanmasını istiyorsanız bu değeri seçin. Seçeneğin değeri sunucuya geçirilir ve **PCF** ve **MQSC** tarafından sorgulanabilir.

Yalnızca istemci bağlantıları için CNRC, CNRC D ve CNRCQM seçeneklerini kullanın. Seçenekler bir bağ tanımlama bağlantısı için kullanılıyorsa, MQCONNX tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_OPTIONS_ERROR ile başarısız olur.

Varsayılan seçenek: Açıklanan seçeneklerin hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

CNNONE

Seçenek belirtilmedi.

CNNONE , program belgelerine yardımcı olmak için tanımlanır. Bu seçeneğin başka bir CN* seçeneğiyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

CNSCO (10 basamaklı işaretli tamsayı)

MQCNO yapısının başlangıcından itibaren bir MQSCO yapısının byte cinsinden görelî konumu.

CNVER CNVER4 değerinden küçükse bu alan yoksayılr.

CNSCP (gösterge)

Bu, bir MQSCO yapısının adresidir.

CNVER CNVER4 değerinden küçükse bu alan yoksayılr.

CNSECP O (10 basamaklı işaretli tamsayı)

Güvenlik parametreleri görelî konumu. Bir kullanıcı kimliği ve parola belirtmek için kullanılan MQCSP yapısının görelî konumu.

Değer pozitif ya da negatif olabilir. Bu alanın ilk değeri 0 'dır.

CNVER CNVER5 değerinden küçükse bu alan yoksayılr.

CNSECPP (gösterge)

Güvenlik değiştirgeleri göstergesi. Kullanıcı kimliği ve parola belirtmek için kullanılan MQCSP yapısının adresi.

Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

CNVER CNVER5 değerinden küçükse bu alan yoksayılr.

CNSID (4 baytlık karakter dizgisi)

MQCNO yapısına ilişkin yapı tanıtıcısı.

Değer şöyle olmalıdır:

CNSIDV

Bağlantı seçenekleri yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CNSIDV 'dir.

CNVER (10 basamaklı işaretli tamsayı)

MQCNO yapısına ilişkin yapı sürümü numarası.

Değer şöyle olmalıdır:

CNVER6

Version-6 bağlantı seçenekleri yapısı.

Bu sürüm tüm ortamlarda desteklenir.

CNVER7

Version-7 bağlantı seçenekleri yapısı.

Bu sürüm tüm ortamlarda desteklenir.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

CNVERC

Bağlantı seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CNVER7' dir.

Başlangıçtaki değerler

<i>Çizelge 693. MQCNO içindeki alanlar</i>		
Alan adı	Değişmezin adı	Değişmezin değeri
CNSID	CNSIDV	' CNO-
CNVER	CNVER5	1
CNOPT	CNNONE	0
CNCCO	Yok	0
CNCCP	Yok	Boş değerli gösterge ya da boş değerli byte
CNCT	CTNONE.	Boş Değerler
CNSCP	Yok	Boş değerli gösterge ya da boş değerli byte
CNSCO	Yok	0
CNCONID	Yok	Boş Değerler
CNSECPO	Yok	0
CNSECPP	Yok	Boş değerli gösterge ya da boş değerli byte
CCDTUL	Yok	0
CCDTUO	Yok	0
CCDTUP	Yok	Boş değerli gösterge ya da boş değerli byte

Notlar:

1. - simgesi tek bir boş karakteri temsil eder.

RPG bildirimi

```
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*.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 manager connection tag
D  CNCT          33         160   INZ(X'0000000000000000-
D                                     00000000000000000000000000-
D                                     00000000000000000000000000-
D                                     00000000000000000000000000-
D                                     00000000000000000000000000-
D                                     00000000000000000000000000-
D                                     00000000000000000000000000-
D                                     00000000000000000000000000-
D                                     00000000000000000000000000-
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'0000000000000000-
D 0000000000000000000000-
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 CNCCDTU0 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

IBM i üzerinde MQCSP (Güvenlik deęiřtirgeleri)

IBM için MQCSP yapısının özeti.

Genel Bakış

Amaç: MQCSP yapısı, yetkilendirme hizmetinin bir kullanıcı kimliğini ve parolayı doğrulamasını sağlar. MQCONNX çağrısında MQCSP bağlantı güvenliği deęiřtirgeleri yapısını belirtirsiniz.

Karakter kümesi ve kodlama: MQCSP ' deki veriler, **CodedCharSetId** kuyruk yöneticisi öznitelięi ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır.

- [“Alanlar” sayfa 1039](#)
- [“Bařlangıçtaki deęerler” sayfa 1041](#)
- [“RPG bildirim” sayfa 1042](#)

Alanlar

MQCSP yapısı ařaęıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

CSAUTH (10 basamaklı iřaretli tamsayı)

Bu, gerçekteřtirilecek kimlik doğrulama türüdür.

Geçerli deęerler řunlardır:

CSAN.

Kullanıcı kimlięi ve parola alanlarını kullanmayın.

YAPIM

Kullanıcı kimlięi ve parola alanlarını doğrulayın.

Bu bir giriş alanıdır. Bu alanın ilk deęeri CSAN ' dır.

CSCPPL (10 basamaklı iřaretli tamsayı)

Bu, kimlik doğrulamasında kullanılacak parolanın uzunluęudur.

Parolanın uzunluk üst sınırı altyapıya baęlı deęil. Parolanın uzunluęu izin verilenden büyükse, kimlik doğrulama isteęi RC2035 ile başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk deęeri 0 'dır.

CSCPPO (10 basamaklı iřaretli tamsayı)

Bu, kimlik doğrulamasında kullanılacak parolanın bayt cinsinden görelidir.

Görelî konum pozitif ya da negatif olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

CSCPPP (gösterge)

Bu, kimlik doğrulamasında kullanılacak parolanın adresidir.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş işaretçidir.

CSCSPUIL (10 basamaklı işaretli tamsayı)

Bu, kimlik doğrulamasında kullanılacak kullanıcı kimliğinin uzunluğudur.

Kullanıcı kimliğinin uzunluk üst sınırı altyapıya bağımlı değil. Kullanıcı kimliğinin uzunluğu izin verileden büyükse, kimlik doğrulama isteği RC2035ile başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

CSCSPUIO (10 basamaklı imzalı tamsayı)

Bu, kimlik doğrulamasında kullanılacak kullanıcı kimliğinin bayt cinsinden görelî konumudur.

Görelî konum pozitif ya da negatif olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

CSCSPUIP (gösterge)

Bu, kimlik doğrulamasında kullanılacak kullanıcı kimliğinin adresidir.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş işaretçidir. CSVER, CSVER5değerinden küçükse bu alan yoksayılır.

V 9.3.0 V 9.3.0 CSINITKL (10 basamaklı işaretli tamsayı)

Bu, parola koruma sistemine ilişkin ilk anahtarın uzunluğudur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

V 9.3.0 V 9.3.0 CSINITKO (10 basamaklı işaretli tamsayı)

Bu, parola koruma sistemine ilişkin ilk anahtarın bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir.

İlk anahtarı belirtmek için *CSINITKO* ya da *CSINITKP* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız. Daha fazla bilgi için *CSINITKP* alanının açıklamasına bakın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

V 9.3.0 V 9.3.0 CSINITKP (işaretçi)

Bu, parola koruma sistemine ilişkin ilk anahtarın bayt cinsinden adresidir.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş işaretçidir.

IBM MQ MQI clients , IBM MQ parola koruma sistemi kullanılarak şifrelenmiş değerler olarak bazı alanların değerini sağlayabilir. Aşağıdaki alanlar şifrelenmiş değerler içerebilir:

- MQSCO yapısındaki anahtar havuzu parolası.

İlk anahtar, şifreleme algoritması tarafından bu değerleri şifrelemek ve bu değerlerin şifresini çözmek için kullanılır. Bu alanların değerleri **runmqicred** yardımcı programı kullanılarak şifrelendiğinde bir başlangıç anahtarı sağlanırsa, istemci kuyruk yöneticisine bağlandığında aynı başlangıç anahtarı belirtilmelidir.

Bu alanı kullanarak belirtilen ilk anahtar, MQS_MQI_KEYFILE ortam değişkeni ya da istemci yapılandırma dosyasının Security (Güvenlik) kısmına ilişkin *MQIInitialKeyFile* özelliği kullanılarak belirtilen ilk anahtarı geçersiz kılar.

İlk anahtarı belirtmek için *CSINITKO* ya da *CSINITKP* kullanabilirsiniz, ancak her ikisini birden kullanamazsınız.

CSRE1 (4 baytlık karakter dizgisi)

IBM üzerinde işaretçi hizalaması için gerekli ayrılmış bir alan.

Bu bir giriş alanıdır. Bu alanın başlangıç değerinin tümü boş.

CSRS2 (8 baytlık karakter dizgisi)

IBM üzerinde işaretçi hizalaması için gerekli ayrılmış bir alan.

Bu bir giriş alanıdır. Bu alanın başlangıç değerinin tümü boş.

CSSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

CSSIDV

Güvenlik parametreleri yapısına ilişkin tanıtıcı.

CSVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

CSVER1

Version-1 güvenlik parametreleri yapısı.

V 9.3.0 **V 9.3.0** **V 9.3.0** **CSVER2**

Version-2 güvenlik parametreleri yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

CVERC







Güvenlik parametreleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CSVER1' dir.

Başlangıçtaki değerler

Çizelge 694. MQCNO içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
CSSID	CSSIDV	' CSP- '
CSVER	CSVER1	1
CSAUTH	Yok	0
CSRE1	Yok	Boş Değerler
CSCSPUIP	Yok	Boş değerli gösterge
CSCSPUIO	Yok	0
CSCSPUIL	Yok	0
CSRS2	Yok	Boş Değerler
CSCPPP	Yok	Boş değerli gösterge
CSCPPO	Yok	0
CSCPPL	Yok	0

Çizelge 694. MQCNO içindeki alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
  CSINITKP	Yok	Boş değerli gösterge
  CSINITKO	Yok	0
  CSINITKL	Yok	0

Not:

1. → simgesi tek bir boş karakteri temsil eder.

RPG bildirimi

```
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 CSCSPUIL 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 CSCPPPO 65 68I 0 INZ(0)
D* Length of password
D CSCPPL 69 72I 0 INZ(0)
```

 IBM i

IBM i üzerinde MQCTLO (Denetim geri çağırma seçenekleri yapısı)

Denetim geri çağırma işlevini belirten yapı.

Genel Bakış

Amaç

MQCTLO yapısı, bir denetim geri çağırma işleviyle ilgili seçenekleri belirtmek için kullanılır.

Yapı, [MQCTL](#) çağırısındaki bir giriş ve çıkış değiştirgesidir.

Sürüm

MQCTLO ' nun yürürlükteki sürümü CTLV1.

Karakter kümesi ve kodlama

MQCTLO içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

- “Alanlar” sayfa 1043
- “Başlangıçtaki değerler” sayfa 1044
- “RPG bildirim” sayfa 1044

Alanlar

MQCTLO yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

COCONNAREA (10 basamaklı işaretli tamsayı)

Denetim seçenekleri yapısı- ConnectionArea alanı.

Bu, geri çağırma işlevinin kullanabileceği bir alandır.

Kuyruk yöneticisi bu alanın içeriğine dayalı olarak karar vermez ve MQCB çağrısındaki bir parametre olan MQCBC yapısındaki CBCCONNAREA alanından değiştirilmeden geçirilir.

Bu alan, CTLSR ve CTLSW dışındaki tüm işlemler için yoksayıdır.

Bu, geri çağırma işlevine ilişkin bir giriş ve çıkış alanıdır. Bu alanın ilk değeri boş değerli bir gösterge ya da boş değerli byte.

COOPT (10 basamaklı işaretli tamsayı)

MQCTLO ' nun işlemini denetleyen seçenekler.

CTLFQ

Kuyruk yöneticisi ya da bağlantı susturma durumundaysa MQCTLO çağırısı başarısız yapmaya zorlayın.

MQCB çağrısında geçirilen MQGMO seçeneklerinde, ileti tüketicilerine susturulurken bildirim gönderilmesini sağlamak için GMFIQ değerini belirtin.

CTLTHR

Bu seçenek, uygulamaya, aynı bağlantı için tüm ileti tüketicilerinin aynı iş parçasığında çağrılmasını gerektirdiğini bildirir.

Varsayılan seçenek: Açıklanan seçeneklerden herhangi birine gerek duymuyorsanız, aşağıdaki seçeneği kullanın:

CTLNO.

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar. CTLNO, program belgelerine yardımcı olmak üzere tanımlanır; bu seçeneğin başka bir değerle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu bir giriş alanıdır. *COOPT* alanının ilk değeri CTLNO değeridir.

CORSV (10 basamaklı işaretli tamsayı)

Bu ayrılmış bir alandır. Bu alanın ilk değeri boş bir karakterdir.

COSID (10 basamaklı işaretli tamsayı)

Denetim seçenekleri yapısı- StrucId alanı.

Bu yapı tanıtıcısıdır; değer şu olmalıdır:

CTLSI

Denetim Seçenekleri yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CTLSI 'dir.

COVER (10 basamaklı işaretli tamsayı)

Denetim seçenekleri yapısı-Sürüm alanı.

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

CTLV1

Version-1 Denetim seçenekleri yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

CTLCV

Denetim seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri CTLV1' dir.

Başlangıçtaki değerler

Çizelge 695. MÖCTLO 'daki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
COSID	CTLSI	' CTLO '
COVER	CTLV1	1
COOPT	CTLNO.	Boş Değerler
CORSV	Ayrılmış alan	
COCONNAREA	Yok	Boş değerli gösterge ya da boş değerli byte

RPG bildirimi

```
D* MÖCTLO 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 MÖCTL
D COOPT          9      12I 0 INZ(0)
D*
D* Reserved
D CORSV         13     16I 0 INZ(-1)
D*
D* MÖCTL Data area passed to the function
D COCONNAREA    17     32*  INZ(*NULL)
```

IBM i IBM i üzerinde MÖDH (Dağıtım üstbilgisi)

MÖDH yapısı, ileti bir iletim kuyruğunda saklanan bir dağıtım listesi ileti olduğunda iletide bulunan ek verileri tanımlar.

Genel Bakış

Amaç: Dağıtım listesi ileti, birden çok hedef kuyruğuna gönderilen bir iletidir. Ek veriler, MÖDH yapısının ardından bir MÖOR kayıtları dizisi ve bir MÖPMR kayıtları dizisinden oluşur.

Bu yapı, iletileri doğrudan iletim kuyruklarına koyan ya da iletim kuyruklarından iletileri kaldiren özel uygulamalar (örneğin, ileti kanalı araçları) tarafından kullanılır.

Bu yapı, iletileri dağıtım listelerine koymak isteyen normal uygulamalar tarafından kullanılmamalıdır. Bu uygulamalar, dağıtım listesindeki hedefleri tanımlamak için MÖOD yapısını ve ileti özelliklerini belirtmek için MÖPMO yapısını kullanmalı ya da tek tek hedeflere gönderilen iletilere ilişkin bilgileri almalıdır.

Karakter kümesi ve kodlama: MQDH ' deki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından belirtilen karakter kümesinde ve C programlama dili için ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlamasında olmalıdır.

MQDH ' nin karakter kümesi ve kodlaması aşağıdaki alanlarda *MDCSI* ve *MDENC* alanlarına ayarlanmalıdır:

- MQMD (MQDH yapısı ileti verilerinin başındadır) ya da
- MQDH yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Kullanım: Bir uygulama bir iletiyi dağıtım listesine koyduğunda ve hedeflerin bazıları ya da tümü uzak olduğunda, kuyruk yöneticisi uygulama ileti verilerini MQXQH ve MQDH yapılarıyla önekler ve iletiyi ilgili iletim kuyruğuna yerleştirir. Bu nedenle, ileti bir iletim kuyruğunda olduğunda veriler aşağıdaki sırada gerçekleşir:

- MQXQH yapısı
- MQDH yapısı ve MQOR ve MQPMR kayıtlarının dizileri
- Uygulama iletisi verileri

Hedeflere bağlı olarak, kuyruk yöneticisi tarafından birden çok ileti oluşturulabilir ve farklı iletim kuyruklarına yerleştirilebilir. Bu durumda, bu iletilerdeki MQDH yapıları, uygulama tarafından açılan dağıtım listesi tarafından tanımlanan hedeflerin farklı alt kümelerini tanımlar.

Dağıtım listesi iletisini doğrudan bir iletim kuyruğuna koyan bir uygulama, daha önce açıklanan sıraya uymalı ve MQDH yapısının doğru olduğundan emin olmalıdır. MQDH yapısı geçerli değilse, kuyruk yöneticisi RC2135neden koduyla MQPUT ya da MQPUT1 çağrısının başarısız olmasını seçebilir.

Messages can be stored on a queue in distribution-list form only if the queue is defined as being able to support distribution list messages (see the **DistLists** queue attribute described in “Kuyruklara ilişkin öznelilikler” sayfa 1345). Bir uygulama dağıtım listesi iletisini dağıtım listelerini desteklemeyen bir kuyruğa doğrudan koyarsa, kuyruk yöneticisi dağıtım listesi iletisini tek tek iletilere böler ve bunun yerine kuyruktaki iletileri yerleştirir.

- “Alanlar” sayfa 1045
- “Başlangıçtaki değerler” sayfa 1048
- “RPG bildirim” sayfa 1048

Alanlar

MQDH yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

DHCNT (10 basamaklı işaretli tamsayı)

Var olan MQOR kayıtlarının sayısı.

Bu, hedef sayısını tanımlar. Bir dağıtım listesi her zaman en az bir hedef içermelidir; bu nedenle *DHCNT* her zaman sıfırdan büyük olmalıdır.

Bu alanın ilk değeri 0 'dır.

DHCSI (10 basamaklı işaretli tamsayı)

MQOR ve MQPMR kayıtlarını izleyen verilerin karakter kümesi tanıtıcısı.

MQOR ve MQPMR kayıtlarının dizilerini izleyen verilerin karakter kümesi tanıtıcısını belirtir; MQDH yapısının kendisindeki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değer kullanılabilir:

CSINHT

Bu yapının karakter kümesi tanıtıcısını edin.

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MÖGET çağrısı CSINHT değerini döndürmez.

MÖMD 'deki *MDPAT* alanının değeri ATBRKR ise, CSINHT kullanılamaz.

Bu alanın ilk değeri CSUNDF 'dir.

DHENC (10 basamaklı işaretli tamsayı)

MÖOR ve MÖPMR kayıtlarını izleyen verilerin sayısal kodlaması.

MÖOR ve MÖPMR kayıtlarının dizilerini izleyen verilerin sayısal kodlamasını belirtir; MÖDH yapısının kendisindeki sayısal veriler için geçerli değildir.

MÖPUT ya da MÖPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri 0 'dır.

DHFLG (10 basamaklı işaretli tamsayı)

Genel bayraklar.

Aşağıdaki işaret belirtilebilir:

DHFYENİ

Yeni ileti tanıtıcıları oluşturur.

Bu işaret, dağıtım listesindeki her hedef için yeni bir ileti tanıtıcısı oluşturulacağını gösterir. Bu, yalnızca bir koyma iletisi kaydı olmadığında ya da kayıtlar varsa ancak *PRMID* alanını içermediğinde ayarlanabilir.

Bu işaretin kullanılması, ileti tanıtıcılarının oluşturulmasını olası son ana, yani dağıtım listesi iletisinin son olarak tek tek iletilere bölünme anına kadar erteler. Bu, dağıtım listesi iletisiyle akması gereken denetim bilgisi miktarını en aza indirir.

Bir uygulama bir dağıtım listesine ileti yerleştirdiğinde, kuyruk yöneticisi MÖDH 'de DHFNEW değerini aşağıdaki deyimlerin her ikisi de doğru olduğunda oluşturur:

- Uygulama tarafından sağlanan bir koyma iletisi kaydı yok ya da sağlanan kayıtlar *PRMID* alanını içermiyor.
- MÖMD 'deki *MDMID* alanı MINONE ya da MÖPMO içindeki *PMOPT* alanı PMNMID' yi içeriyor

İşaret gerekmezse, aşağıdaki belirtilebilir:

DHFNON

Bayrak yok.

Bu değişmez, işaret belirtilmediğini gösterir. DHFNON, program belgelerine yardımcı olmak için tanımlanmıştır. Bu değişmezin başka bir değişmeze birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğundan bu tür bir kullanım algılanamaz.

Bu alanın ilk değeri DHFNON.

DHFMT (8 baytlık karakter dizisi)

MÖOR ve MÖPMR kayıtlarını izleyen verilerin biçim adı.

MÖOD ve MÖPMR kayıtlarının dizilerini izleyen verilerin biçim adını belirtir (hangisi sonsa).

MÖPUT ya da MÖPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MÖMD 'deki *MDFMT* alanıyla aynıdır.

Bu alanın ilk değeri FMNONE.

DHLEN (10 basamaklı işaretli tamsayı)

MÖDH yapısının uzunluğu ve aşağıdaki MÖOR ve MÖPMR kayıtları.

MÖDH yapısının başlangıcından MÖOR ve MÖPMR kayıtlarının dizilerini izleyen ileti verilerinin başlangıçlarına kadar olan byte sayısı. Veriler aşağıdaki sırada oluşur:

- MDH yapısı
- MOR kayıtları dizisi
- MPMR kayıtları dizisi
- İleti Verileri

MOR ve MPMR kayıtlarının dizileri, MDH yapısında bulunan görelı konumlar tarafından ele alınır. Bu görelı konumlar bir ya da daha fazla MDH yapısı, kayıt dizileri ve ileti verileri arasında kullanılmayan byte 'lar ile sonuçlandıysa, kullanılmayan byte 'ların *DHLEN*değerine eklenmesi gerekir, ancak bu byte 'ların içeriđi kuyruk yöneticisi tarafından korunmaz. MPMR kayıtları dizisinin MOR kayıtları dizisinden önce gelmesi için geçerlidir.

Bu alanın ilk değeri 0 'dır.

DHORO (10 basamaklı işaretli tamsayı)

MDH ' nin başlangıcındaki ilk MOR kaydının görelı konumu.

Bu alan, hedef kuyrukların adlarını içeren MOR nesne kayıtları dizisindeki ilk kaydın bayt cinsinden görelı konumunu verir. Bu dizide *DHCNT* kayıt var. Bu kayıtlar (artı ilk nesne kaydı ile önceki alan arasında atlanan bayt), *DHLEN* alanı tarafından belirtilen uzunluđa eklenir.

Bir dağıtım listesi her zaman en az bir hedef içermelidir; bu nedenle *DHORO* her zaman sıfırdan büyük olmalıdır.

Bu alanın ilk değeri 0 'dır.

DHPRF (10 basamaklı işaretli tamsayı)

Hangi MPMR alanlarının var olduğunu gösteren işaretler.

Aşğıdaki işaretlerden sıfır ya da daha fazlası belirtilebilir:

PFMID

İleti tanıtıcısı alanı var.

PFCID

İlinti tanıtıcısı alanı var.

PFGTnt

Grup tanıtıcısı alanı var.

PFFB.

Geribildirim alanı var.

PFACC

Muhasebe simgesi alanı var.

MPMR alanı yoksa, aşğıdaki belirtilebilir:

PFNONE

Bir koyma iletisi kayıt alanı yok.

PFNONE, program belgelerine yardımcı olmak için tanımlanır. Bu değışmezin başka bir değışmele birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu alanın ilk değeri PFNONE.

DHPRO (10 basamaklı işaretli tamsayı)

MDH ' nin başlangıcındaki ilk MPMR kaydının görelı konumu.

Bu alan, ileti özelliklerini içeren MPMR koyma iletisi kayıtları dizisindeki ilk kaydın bayt cinsinden görelı konumunu verir. Varsa, bu dizide *DHCNT* kayıt vardır. Bu kayıtlar (artı ilk koyma iletisi kaydı ile önceki alan arasında atlanan byte 'lar), *DHLEN* alanı tarafından belirtilen uzunluđa eklenir.

Koyma iletisi kayıtları isteđe bađlıdır; kayıt sağlanmazsa, *DHPRO* sıfırdır ve *DHPRF* PFNONE değerine sahiptir.

Bu alanın ilk değeri 0 'dır.

DHSID (4 baytlık karakter dizilimi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

DHSIDV

Dağıtım üstbilgisi yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri DHSIDV 'dir.

DHVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

DHVER1

Dağıtım üstbilgisi yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

DHVERC

Dağıtım üstbilgisi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri DHVER1' dir.

Başlangıçtaki değerler

Çizelge 696. MQDH ' deki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
DHSID	DHSIDV	'DH↯↯'
DHVER	DHVER1	1
DHLEN	Yok	0
DHENC	Yok	0
DHCSI	CSUNDF	0
DHFMT	FMNONE	Boşluklar
DHFLG	DHFNON	0
DHPRF	PFNONE	0
DHCNT	Yok	0
DHORO	Yok	0
DHPRO	Yok	0

Notlar:

1. ↯ simgesi, tek bir boş karakteri temsil eder.

RPG bildirim

```
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 MQPMR records
D DHLEN          9      12I 0 INZ(0)
D* Numeric encoding of data that follows the MQOR and MQPMR records
D DHENC          13     16I 0 INZ(0)
D* Character set identifier of data that follows the MQOR and MQPMR
D* records
D DHCSI          17     20I 0 INZ(0)
D* Format name of data that follows the MQOR and MQPMR records
D DHFMT          21     28   INZ(' ')
D* General flags
D DHFLG          29     32I 0 INZ(0)
D* Flags indicating which MQPMR 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 MQPMR record from start of MQDH
D DHPRO          45     48I 0 INZ(0)

```

IBM i MQDLH (Ölü harf üstbilgisi)- IBM i

Genel Bakış

Amaç

MQDLH yapısı, teslim edilmeyen ileti kuyruğundaki iletilerin uygulama iletileri verilerinin önekini oluşturan bilgileri açıklar. Kuyruk yöneticisi ya da ileti kanalı aracısı iletiyi kuyruğa yeniden yönlendirdiği için, ileti gönderilmeyen iletiler kuyruğuna gelebilir. Bir uygulama iletiyi doğrudan kuyruğa koyabilir.

Biçim Adı

FMDLH

Karakter kümesi ve kodlama

MQDLH , uygulama iletilerinin başında olabilir. Bu durumda, MQDLH yapısındaki alanlar MDCSI ve MDENC alanları tarafından verilen karakter kümesi ve kodlamada bulunur. Yoksa, karakter kümesi ve kodlama MQDLH' den önce gelen üstbilgi yapısındaki MDCSI ve MDENC alanları tarafından ayarlanır.

Karakter takımı, kuyruk adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır.

Kullanım

İletileri doğrudan gönderilmeyen iletiler kuyruğuna koyan uygulamalar, ileti verilerine bir MQDLH yapısıyla önek eklemeli ve alanları uygun değerlerle kullanıma hazırlamalıdır. Ancak, kuyruk yöneticisi MQDLH yapısının var olmasını ya da alanlar için geçerli değerlerin belirtilmesini gerektirmez.

Bir ileti, gönderilmeyen iletiler kuyruğuna konamayacak kadar uzunsa, uygulamanın aşağıdakilerden birini yapmayı düşünmesi gerekir:

- İleti verilerini, gönderilmeyen ileti kuyruğuna sığacak şekilde kısaltın.
- İletiyi yardımcı belleğe kaydedin ve gitmeyen ileti kuyruğuna iletinin çok uzun olduğunu belirten bir kural dışı durum raporu iletileri yerleştirin.
- İletiyi atın ve oluşturana bir hata döndürün. İleti kritik bir iletiyse. İletiyi, yalnızca oluşturanın iletinin bir kopyasına sahip olduğu biliniyorsa atın. Örneğin, bir iletişim kanalından ileti kanalı aracısı tarafından alınan bir ileti.

Hangi seçeneklerin uygun olduğu, uygulamanın tasarımına bağlıdır.

Kuyruk yöneticisi, bölüm olan bir ileti ön tarafa MQDLH yapısıyla yerleştirildiğinde özel işlem gerçekleştirir. Daha fazla ayrıntı için MQMDE yapısının açıklamasına bakın.

- [“İletileri gönderilmeyen iletiler kuyruğuna koyma” sayfa 1050](#)
- [“Gitmeyen iletiler kuyruğundan ileti alma” sayfa 1050](#)
- [“Alanlar” sayfa 1050](#)
- [“Başlangıçtaki değerler” sayfa 1054](#)

- “RPG bildirimi” sayfa 1054

İletileri gönderilmeyen iletiler kuyruğuna koyma

İleti teslim edilmeyen iletiler kuyruğuna konursa, MQPUT ya da MQPUT1 çağrısı için kullanılan MQMD yapısı, iletiyle ilişkili MQMD ile aynı olmalıdır. MQMD , genellikle aşağıdaki durumlar dışında MQGET çağrısı tarafından döndürülen bir çağrıdır:

- MDCSI ve MDENC alanları, MQDLH yapısındaki alanlar için kullanılan karakter kümesi ve kodlamaya ayarlanmalıdır.
- Verilerin MQDLH yapısıyla başladığını belirtmek için MDFMT alanı FMDLH olarak ayarlanmalıdır.
- MDACC, MDAID, MDAOD, MDPAN, MDPAT, MDPD, MDPTve MDUID bağlam alanları, koşullara uygun bir bağlam seçeneği kullanılarak ayarlanmalıdır:
 - Gitmeyen ileti kuyruğuna önceki iletilerle ilgili olmayan bir iletiyi koyan bir uygulamanın PMDEFEC seçeneğini kullanması gerekir. PMDEFEC seçeneği, kuyruk yöneticisinin ileti tanımlayıcısındaki tüm bağlam alanlarını varsayılan değerlerine ayarlamasına neden olur.
 - Alınan bir iletiyi göndermeyen ileti kuyruğuna koyacak bir sunucu uygulaması, özgün bağlam bilgilerini korumak için PMPASA seçeneğini kullanmalıdır.
 - Alınan iletiye yanıt gönderen bir sunucu uygulaması PMPASI seçeneğini kullanmalıdır. PMPASI seçeneği kimlik bilgilerini korur, ancak kaynak bilgilerini sunucu uygulamasının kaynağı olarak ayarlar.
 - İleti kanalı aracısı, iletişim kanalından aldığı bir iletiyi göndererek PMSETA seçeneğini kullanmalıdır. PMSETA seçeneği, özgün bağlam bilgilerini korur.

MQDLH yapısının kendisinde, alanlar aşağıdaki gibi ayarlanmalıdır:

- DLCSI, DLENCve *DLFMT* alanları, MQDLH yapısını izleyen verileri tanımlayan değerlere ayarlanmalıdır. Bu değerler genellikle özgün ileti tanımlayıcısındaki değerlerdir.
- DLPAT, DLPAN, DLPDve DLPT bağlam alanları, iletiyi teslim edilmeyen ileti kuyruğuna koyan uygulamaya uygun değerlere ayarlanmalıdır. Bu değerler özgün iletiyle ilgili değildir.
- Diğer alanlar uygun şekilde ayarlanmalıdır.

Uygulama, tüm alanların geçerli değerlere sahip olduğundan ve karakter alanlarının, tanımlanan alan uzunluğuna kadar boşluklarla doldurulması gerektiğinden emin olmalıdır. Karakter verileri, boş bir karakter kullanılarak zamanından önce sonlandırılmamalıdır. Kuyruk yöneticisi, MQDLH yapısında boş ve sonraki karakterleri boşluklara dönüştürmez.

Gitmeyen iletiler kuyruğundan ileti alma

Gitmeyen iletiler kuyruğundan ileti alan uygulamalar, iletilerin MQDLH yapısıyla başladığını doğrulamalıdır. Uygulama, ileti tanımlayıcısındaki MDFMT alanını inceleyerek MQDLH yapısının var olup olmadığını saptayabilir MQMD. Alan FMDLHdeğerine sahipse, ileti verileri bir MQDLH yapısıyla başlar. Gönderilmeyen iletiler kuyruğundaki iletiler, başlangıçta amaçlandıkları kuyruk için çok uzunsa kesilebilir.

Alanlar

MQDLH yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

DLCISI (10 basamaklı imzalı tamsayı)

MQDLH' i izleyen verilerin karakter kümesi tanıtıcısı.

DLCISI , MQDLH yapısını izleyen verilerin karakter takımı tanıtıcısını belirtir. Veriler genellikle özgün iletiden elde edilir. MQDLH yapısındaki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulama bu alanı veriler için uygun değere ayarlamalıdır. Aşağıdaki özel değer kullanılabilir:

CSINHT

Bu yapının karakter kümesi tanıtıcısını edin.

Bu yapıyı izleyen verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı CSINHT değerini döndürmez.

CSINHT , MQMD içindeki MDPAT alanının değeri ATBRKRise kullanılamaz.

Bu alanın ilk değeri şudur: CSUNDF.

DLDM (48 baytlık karakter dizgisi)

Özgün hedef kuyruk yöneticisinin adı.

Bu, iletinin özgün hedefi olan kuyruk yöneticisinin adıdır.

Bu alanın uzunluğu LNQMntarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

DLDQ (48 baytlık karakter dizilimi)

Özgün hedef kuyruğun adı.

Bu, iletinin özgün hedefi olan ileti kuyruğunun adıdır.

Bu alanın uzunluğu LNQNtarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

DLENC (10 basamaklı işaretli tamsayı)

MQDLH' i izleyen verilerin sayısal kodlaması.

DLENC , MQDLH yapısını izleyen verilerin sayısal kodlamasını belirtir. Veriler genellikle özgün iletiden elde edilir. MQDLH yapısının kendisindeki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulama bu alanı veriler için uygun değere ayarlamalıdır.

Bu alanın ilk değeri 0 'dır.

DLFMT (8 baytlık karakter dizgisi)

MQDLH' i izleyen verilerin biçim adı.

Bu, MQDLH yapısını izleyen verilerin biçim adını belirtir (genellikle özgün iletideki veriler).

MQPUT ya da MQPUT1 çağrısında, uygulama bu alanı veriler için uygun değere ayarlamalıdır. Bu alanı kodlamaya ilişkin kurallar, MQMDiçindeki MDFMT alanına ilişkin kurullarla aynıdır.

Bu alanın uzunluğu LNFMTtarafından verilir. Bu alanın ilk değeri şudur: FMNONE.

DLPAN (28 baytlık karakter dizgisi)

İletiyi teslim edilmeyen ileti kuyruğuna koyan uygulamanın adı.

Adın biçimi DLPAT alanına bağlıdır. [“IBM i üzerinde MQMD \(İleti tanımlayıcı\)” sayfa 1093](#)içindeki MDPAN alanının açıklamasına bakın.

İletiyi göndermeyen iletiler kuyruğuna yeniden yönlendiren kuyruk yöneticisiyse, DLPAN kuyruk yöneticisi adının ilk 28 karakterini içerir. Gerekirse ad boşluklarla doldurulur.

Bu alanın uzunluğu LNPANtarafından verilir. Bu alanın ilk değeri 28 boş karakterdir.

DLPAT (10 basamaklı işaretli tamsayı)

İletiyi teslim edilmeyen ileti kuyruğuna koyan uygulamanın tipi.

Bu alan, MQMD ileti tanımlayıcısındaki MDPAT alanıyla aynı anlama sahiptir (ayrıntılar için bkz. [“IBM i üzerinde MQMD \(İleti tanımlayıcı\)” sayfa 1093](#)).

İletiyi göndermeyen iletiler kuyruğuna yeniden yönlendiren kuyruk yöneticisiyse, DLPAT ' in değeri ATQMolur.

Bu alanın ilk değeri 0 'dır.

DLPD (8 baytlık karakter dizgisi)

İletinin teslim edilmeyen ileti kuyruğuna yerleştirildiği tarih.

Bu alanın kuyruk yöneticisi tarafından oluşturulduğu tarih için kullanılan biçim:

• YYYYMMDD

karakterlerin temsil ettiği yer:

YYYY

yıl (dört sayısal basamak)

MM

yılın ayı (01-12)

DD

ayın günü (01-31)

Greenwich Ortalama Saati (GMT), DLPD ve DLPT alanları için kullanılır. Bu alanlar, sistem saatinin GMT 'ye doğru olarak ayarlanmasına tabidir.

Bu alanın uzunluğu LNPDAT tarafından verilir. Bu alanın ilk değeri sekiz boş karakterdir.

DLPT (8 baytlık karakter dizgisi)

İletinin teslim edilmeyen ileti kuyruğuna konma süresi.

Bu alanın kuyruk yöneticisi tarafından oluşturulduğu saat için kullanılan biçim:

• HHMMSSSTH

burada karakterler (sırayla) temsil edilir:

HH

saat (00-23)

MM

dakika (00-59)

SS

saniye (00-59; bu konuda daha sonra nota bakın)

T

saniyenin onda biri (0-9)

H

saniyenin yüzde biri (0-9)

Not: Sistem saati doğru bir zaman standardına eşitlenirse, DLPT içindeki saniyeler için 60 ya da 61 döndürülebilir. Fazladan saniye, artık saniyeler genel zaman standardına eklendiğinde ortaya çıkar.

Greenwich Ortalama Saati (GMT), DLPD ve DLPT alanları için kullanılır. Bu alanlar, sistem saatinin GMT 'ye doğru olarak ayarlanmasına tabidir.

Bu alanın uzunluğu LNPTIM tarafından verilir. Bu alanın ilk değeri sekiz boş karakterdir.

DLREA (10 basamaklı işaretli tamsayı)

Neden iletisinin teslim edilmeyen (teslim edilmeyen) iletiler kuyruğuna gelmesinin nedeni.

Bu, iletinin özgün hedef kuyruk yerine, gönderilmeyen iletiler kuyruğuna yerleştirilme nedenini tanımlar. FB* ya da RC* değerlerinden biri olmalıdır (örneğin, RC2053). Ortaya çıkabilecek ortak FB* değerlerinin ayrıntıları için ["IBM i üzerinde MQMD \(İleti tanımlayıcı\)" sayfa 1093](#) içindeki *MDFB* alanının açıklamasına bakın.

Değer FBIFST - FBILST aralığındaysa, gerçek IMS hata kodu, *DLREA* alanının değerinden FBIERR çıkarılarak belirlenebilir.

Bazı FB* değerleri yalnızca bu alanda oluşur. Bunlar, teslim edilmeyen iletiler kuyruğuna aktarılan havuz iletileri, tetikleyici iletiler ya da iletim kuyruğu iletileriyle ilişkilendirir. Bu değerler şunlardır:

FBABEG

Uygulama başlatılmıyor.

Tetikleyici iletisini işleyen bir uygulama, tetikleyici iletisinin TMAI alanında adı belirtilen uygulamayı başlatamadı; bkz. “MQTM-Tetikleyici iletisi” sayfa 1215.

FBATYP

Uygulama tipi hatası.

Tetikleyici iletisinin TMAI alanı geçersiz olduğundan, tetikleyici iletisini işleyen bir uygulama uygulamayı başlatamadı; bkz. “MQTM-Tetikleyici iletisi” sayfa 1215.

FBBODC

Küme-alıcı kanalı silindi.

İleti, FBIERR seçeneğiyle açılan bir küme kuyruğuna yönelik bir küme iletim kuyruğuydu . İletiyi hedef kuyruğa iletmek için kullanılacak uzak küme-alıcı kanalı, ileti gönderilmeden önce silindi. FBIERR belirtildiğinden, iletiyi iletmek için yalnızca kuyruk açıldığında seçilen kanal kullanılabilir. Bu kanal artık kullanılmadığı için, ileti gitmeyen iletiler kuyruğuna yerleştirildi.

FBNARM

İleti bir havuz iletisi değil.

FBSBCX

İleti, kanal otomatik tanımlama çıkışı tarafından durduruldu.

FBSBMX

İleti, kanal ileti çıkışı tarafından durduruldu.

FBTM

MQTM yapısı geçerli değil ya da eksik.

MQMD içindeki MDFMT alanı FMTMdeğerini belirtir, ancak ileti geçerli bir MQTM yapısıyla başlamaz. Örneğin, *TMSID* anımsatıcı geçerli olmayabilir. *TMVER* tanınmayabilir. Tetikleyici iletisinin uzunluğu, MQTM yapısını içermek için yetersiz olabilir.

FBXQME

İletim kuyruğundaki ileti biçimi yanlış.

Bir ileti kanalı aracısı, iletim kuyruğundaki bir iletinin doğru biçimde olmadığını buldu. İleti kanalı aracısı, bu geribildirim kodunu kullanarak iletiyi gönderilmeyen ileti kuyruğuna koyar.

Bu alanın ilk değeri şudur: RCNONE.

DLSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

DLSIDV

Gönderilmeyen harf üstbilgi yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri şudur: DLSIDV.

DLVER (10 basamaklı imzalı tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

DLVER1

Gönderilmeyen harf üstbilgi yapısına ilişkin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

DLVERC

Ölü harf üstbilgi yapısının geçerli sürümü.

Bu alanın ilk değeri şudur: DLVER1.

Başlangıçtaki değerler

Çizelge 697. MQDLH içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
DLSID	DLSIDV	'DLH↵'
DLVER	DLVER1	1
DLRE	RCNONE	0
DLDQ	Yok	Boşluklar
DLDM	Yok	Boşluklar
DLENC	Yok	0
DLCSI	CSUNDF	0
DLFMT	FMNONE	Boşluklar
DLPAT	Yok	0
DLPAN	Yok	Boşluklar
DLPD	Yok	Boşluklar
DLPT	Yok	Boşluklar

Notlar:

1. ↵ simgesi, tek bir boş karakteri temsil eder.

RPG bildirim

```
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
```

IBM i

IBM i üzerinde MQDMHO (İleti işleme seçeneklerini sil)

MQDMHO yapısı, uygulamaların ileti tanıtıcılarının nasıl silineceğini denetleyen seçenekleri belirtmesine olanak sağlar.

Genel Bakış

Amaç: Yapı, **MQDLTMH** çağrısında bir giriş parametresidir.

Karakter kümesi ve kodlama: **MQDMHO** içindeki veriler, uygulamanın karakter kümesinde ve kodlamasında (ENNAT) olmalıdır.

- “Alanlar” sayfa 1055
- “Başlangıçtaki değerler” sayfa 1055
- “RPG bildirimi” sayfa 1055

Alanlar

MQDMHO yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

DMOPT (10 basamaklı işaretli tamsayı)

Değer şöyle olmalıdır:

DMNONE

Seçenek belirtilmedi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **DMNONE**.

DMSID (10 basamaklı işaretli tamsayı)

Bu yapı tanıtıcısıdır; değer şu olmalıdır:

DMSIDV

İleti silme tanıtıcı seçenekleri yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **DMSIDV**.

DMVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

DMVER1

Version-1 ileti tanıtıcısı seçeneklerini silme yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

DMVERC

İleti silme tanıtıcısı seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **DMVER1**.

Başlangıçtaki değerler

Çizelge 698. MQDMHO içindeki alanlar		
Alan adı	Değişmez adı	Değişmez değeri
DMSID	DMSIDV	' DMHO '
DMVER	DMVER1	1
DMOPT	DMNONE	0

RPG bildirimi

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 IBM i üzerinde MQDMPO (İleti özelliğini silme seçenekleri)

İleti silme özelliği seçeneklerini tanımlayan yapı.

Genel Bakış

Amaç: MQDMPO yapısı, uygulamaların iletilerin özelliklerinin nasıl silineceğini denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQDLTMP çağrısına ilişkin bir giriş değişikliğidir.

Karakter kümesi ve kodlama: MQDMPO 'daki veriler, uygulamanın (ENNAT) karakter kümesinde ve kodlamasında olmalıdır.

- [“Alanlar” sayfa 1056](#)
- [“Başlangıçtaki değerler” sayfa 1057](#)
- [“RPG bildirim” sayfa 1057](#)

Alanlar

MQDMPO yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

DPOPT (10 basamaklı işaretli tamsayı)

İleti özellik seçenekleri yapısını sil-DPOPT alanı.

Konum seçenekleri: Aşağıdaki seçenekler, özellik imleciyle karşılaştırıldığında özelliğin görelî konumuyla ilgilidir.

DPDEF

Belirtilen adyla eşleşen ilk özelliği siler.

DPDEL

Özellik imleci tarafından gösterilen özelliği siler; bu, IPINQF ya da IPINQN seçeneği kullanılarak en son sorulmuş olan özelliktir.

İleti tanıtıcısı yeniden kullanıldığında özellik imleci ilk durumuna getirilir. İleti tanıtıcısı bir MQGET çağrısındaki MQGMO ' nun HMSG alanında ya da MQPUT çağrısındaki MQPMO yapısında belirtildiğinde de ilk durumuna getirilir.

İleti tanıtıcısı yeniden kullanıldığında ya da ileti tanıtıcısı MQGET çağrısındaki MQGET yapısında ya da MQPUT çağrısındaki MQPMO yapısında MQGMO yapısının HMSG alanında belirtildiğinde özellik geçici çizelgesi ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında bu seçenek kullanılırsa, çağrı tamamlanma kodu CCFAIL ve neden RC2471 ile başarısız olur. Özellik imlecinin gösterdiği özellik önceden silinmişse, bu kodlarla da başarısız olur.

Bu seçeneklerden hiçbirini gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

DPNONE

Seçenek belirtilmedi.

Bu giriş alanının ilk değeri DPDEF 'dir.

DPSID (10 basamaklı işaretli tamsayı)

İleti özellik seçenekleri yapısını sil-DPSID alanı.

Bu, yapı tanıtıcısıdır. Değer şöyle olmalıdır:

DPSIDV

İleti silme özelliği seçenekleri yapısına ilişkin tanıtıcı.

Bu alan her zaman bir giriş alanıdır. Bu alanın ilk değeri DPSIDV 'dir.

DPVER (10 basamaklı işaretli tamsayı)

İleti özellik seçenekleri yapısını sil-DPVER alanı.

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

DPVER1

İleti silme özelliği seçenekleri yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

DPVERC

İleti silme özelliği seçenekleri yapısının yürürlükteki sürümü.

Bu alan her zaman bir giriş alanıdır. Bu alanın ilk değeri DPVER1' dir.

Başlangıçtaki değerler

Çizelge 699. MQDPMO içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
DPSID	DPSIDV	'DMPO'
DPVER	DPVER1	1
DPOPT	MQDLTMP işlemini denetleyen seçenekler	DPNONE

RPG bildirim

```
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)
```

IBM i üzerinde MQEPH (Embedded PCF header)

Genel Bakış

Amaç

MQEPH yapısı, ileti programlanabilir bir komut biçimi (PCF) iletisi olduğunda iletide bulunan ek verileri tanımlar. *EPPFH* alanı, bu yapıyı izleyen PCF değiştiricelerini tanımlar ve bu, PCF ileti verilerini diğer üstbilgilerle izlemenizi sağlar.

Biçim Adı

EPFMT

Karakter kümesi ve kodlama

MQEPH 'deki veriler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır; bu, **CCSID** kuyruk yöneticisi özniteliği tarafından verilir.

MQEPH 'nin karakter kümesini ve kodlamasını aşağıdaki alanlardaki *MDCSI* ve *MDENC* alanlarına ayarlayın:

- MQMD (MQEPH yapısı ileti verilerinin başındadır) ya da
- MQEPH yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Kullanım

Komut sunucusuna ya da başka bir kuyruk yöneticisi PCF kabul eden sunucuya komut göndermek için MQEPH yapılarını kullanamazsınız.

Benzer şekilde, komut sunucusu ya da başka bir kuyruk yöneticisi PCF kabul eden sunucu, MQEPH yapıları içeren yanıtlar ya da olaylar oluşturmaz.

- [“Alanlar” sayfa 1058](#)
- [“Başlangıçtaki değerler” sayfa 1060](#)
- [“RPG bildirim” sayfa 1060](#)

Alanlar

MQEPH yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

EPCSI (10 basamaklı işaretli tamsayı)

Bu, MQEPH yapısını ve ilişkili PCF değiştirgelerini izleyen verilerin karakter kümesi tanıtıcısıdır; MQEPH yapısının kendisindeki karakter verileri için geçerli değildir.

Bu alanın ilk değeri EPCUND 'dir.

EPENC (10 basamaklı işaretli tamsayı)

Bu, MQEPH yapısını ve ilişkili PCF değiştirgelerini izleyen verilerin sayısal kodlamasıdır; MQEPH yapısındaki karakter verileri için geçerli değildir.

Bu alanın ilk değeri 0 'dır.

EPFLG (10 basamaklı işaretli tamsayı)

Aşağıdaki değerler kullanılabilir:

EPNONE

Hiçbir işaret belirtilmedi. *MDCSI* EPNONE, program belgelerine yardımcı olmak için tanımlanır. Bu değişmezin başka bir değişmeze birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

EPCSEM

Karakter verileri içeren parametrelerin karakter kümesi, her bir yapıdaki *CCSID* alanında tek tek belirtilir. *EPSID* ve *EPFMT* alanlarının karakter kümesi, MQEPH yapısından önce gelen üstbilgi yapısında *CCSID* ile ya da MQEPH iletinin başlangıcında ise MQMD 'deki *MDCSI* alanıyla tanımlanır.

Bu alanın ilk değeri EPNONE.

EPFMT (8 baytlık karakter dizgisi)

MQEPH yapısını ve ilişkili PCF değiştirgelerini izleyen verilerin biçim adıdır.

Bu alanın ilk değeri EPFMNO.

EPLEN (10 basamaklı işaretli tamsayı)

Bu, sonraki üstbilgi yapısından önceki veri miktarıdır. Aşağıdakileri içerir:

- MQEPH üstbilgisinin uzunluğu

- Üstbilgiyi izleyen tüm PCF değıştirmelerinin uzunluęu
- Bu parametreleri izleyen herhangi bir boş dolgu

EPLEN, 4 'ün katı olmalıdır.

Yapının sabit uzunluklu kısmı EPSTLF tarafından tanımlanır.

Bu alanın ilk değeri 68 'dir.

EPPCFH (MÇCFH)

Bu, MÇEPH yapısını izleyen PCF değıştirmelerini tanımlayan programlanabilir komut biçimi (PCF) üstbilgisidir. Bu, PCF ileti verilerini dięer üstbilgilerle izlemenizi sağlar.

PCF üstbilgisi başlangıçta aşağıdaki değerlerle tanımlanır:

<i>Çizelge 700. EPPCFH ' deki Alanlar</i>		
Alan adı	Deęişmezin adı	Deęişmezin değeri
EP3TYP	CFTNON	0
EP3LEN	FHLENV	36
EP3VER	FHVER3	3
EP3CMD	CMNONE	0
EP3SEQ	Yok	1
EP3CTL	CFCLST	1
EEP3CC	CCOK	0
EP3REA	RCNONE	0
EP3CNT	Yok	0

Uygulamanın EP3TYP değeri CFTNON olarak değıştirmesi, yerleşik PCF üstbilgisini kullanması için geçerli bir yapı tipine çevirmesi gerekir.

EPSID (4 baytlık karakter dizilimi)

Deęer şöyle olmalıdır:

EPSTID

Gömülü PCF üstbilgi yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri EPSTID 'dir.

EPVER (10 basamaklı işaretli tamsayı)

Deęer şöyle olabilir:

EPVER1

Yerleşik PCF üstbilgi yapısı için sürüm numarası.

Aşağıdaki değışmez, yürürlükteki sürümün sürüm numarasını belirtir:

EPVER3

Gömülü PCF üstbilgi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri EPVER3' tür.

Başlangıçtaki değerler

Çizelge 701. MÇEPH içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
EPSID	EPSTID	'EP--'
EPVER	EPVER1	1
EPLEN	EPSTLF	68
EPENC	Yok	0
EPCSI	EPCUND	0
EPFMT	EPFMNO	Boşluklar
EPFLG	EPNONE	0
EPPCFH	Çizelge 700 sayfa 1059 içinde tanımlandığı şekilde adlar ve değerler	0

Not:

1. - simgesi, tek bir boş karakteri temsil eder.

RPG bildirim

```
D*..1.....2.....3.....4.....5.....6.....7..
D* MÇEPH Structure
D*
D* Structure identifier
D EPSID 1 4
D* Structure version number
D EPVER 5 8I 0
D* Total length of MÇEPH including MÇCFHand 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 üzerinde MÇGMO (Get-message seçenekleri)

MÇGMO yapısı, uygulamanın iletilerin kuyruklardan nasıl kaldırılacağını denetleyen seçenekleri belirtmesini sağlar.

Genel Bakış

Amaç

Yapı, MÇGET çağrısına ilişkin bir giriş/çıkış değiştirgesidir.

Sürüm

MÖGMO ' nun yürürlükteki sürümü: GMVER4. Yapının yalnızca daha yeni sürümlerinde bulunan alanlar, izleyen açıklamalarda bu şekilde tanımlanır.

Sağlanan COPY dosyası, ortam tarafından desteklenen en son MÖGMO sürümünü içeriyor, ancak *GMVER* alanının ilk değeri GMVER1olarak ayarlanmış. version-1 yapısında var olmayan alanları kullanmak için uygulamanın *GMVER* alanını gerekli sürümün sürüm numarasına ayarlaması gerekir.

Karakter kümesi ve kodlama

MÖGMO ' daki veriler, ENNAT tarafından verilen yerel kuyruk yöneticisinin **CodedCharSetId** kuyruk yöneticisi özniteliği ve kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

- “Alanlar” sayfa 1061
- “Başlangıçtaki değerler” sayfa 1080
- “RPG bildirim” sayfa 1080

Alanlar

MÖGMO yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

GMGST (1 baytlık karakter dizgisi)

Alınan iletinin bir grupta olup olmadığını gösteren işaret.

Aşağıdaki değerlerden birini içerir:

GSNIG

İleti bir grupta değil.

GSMIG

İleti bir grupta, ancak gruptaki son ileti değil.

GSLMIG

İleti, gruptaki son iletidir.

Bu değer, grup tek bir iletiden oluşuyorsa da döndürülür.

Bu alan bir çıkış alanıdır. Bu alanın ilk değeri GSNIG 'dir. *GMVER* , GMVER2değerinden küçükse bu alan yoksayılır.

GMMH (10 basamaklı işaretli tamsayı)

İleti tanıtıcısı

GMPRAQ seçeneği belirtilirse ve PRPCTL kuyruk özniteliği PRPRFH olarak ayarlanmazsa, bu, kuyruktan alınmakta olan iletinin özellikleriyle doldurulan bir iletinin tanıtıcısıdır. Tanıtıcı bir MQCRTMH çağrısıyla yaratılır. Bir ileti alınmadan önce, tanıtıcı ile önceden ilişkilendirilmiş özellikler temizlenir.

Aşağıdaki değer de belirtilebilir:

MQHM_NONE

İleti tanıtıcısı belirtilmedi.

Geçerli bir ileti tanıtıcısı belirtilirse ve çıkışta ileti özelliklerini içerecek şekilde kullanılırsa, MQGET çağrısında ileti tanımlayıcısı gerekmez; giriş alanları için, ileti tanıtıcısıyla ilişkili ileti tanımlayıcısı kullanılır.

MQGET çağrısında bir ileti tanımlayıcısı belirtilirse, bu her zaman bir ileti tanıtıcısıyla ilişkilendirilmiş ileti tanımlayıcısından önceliklidir.

GMPRRF belirtilirse ya da GMPRAQ belirtilirse ve PRPCTL kuyruk özniteliği PRPRFH ise, ileti tanımlayıcı parametresi belirtilmediğinde çağrı RC2026 neden koduyla başarısız olur.

MQGET çağrısından döndüğünüzde, bu ileti tanıtıcısıyla ilişkili özellikler ve ileti tanımlayıcısı, alınan iletinin durumunu (MQGET çağrısında belirtildiyse, ileti tanımlayıcısının yanı sıra) yansıtacak şekilde güncellenir. Daha sonra, MQINQMP çağrısı kullanılarak iletinin özellikleri sorulabiliyor.

İleti tanımlayıcı uzantıları dışında, varsa, ileti verilerinde MQINQMP çağrısıyla sorulabilecek bir özellik bulunmaz; kuyruktaki ileti, ileti verilerinde özellikler içeriyorsa, veriler uygulamaya döndürülmeden önce bunlar ileti verilerinden kaldırılır.

Hiçbir ileti tanıtıcısı sağlanmazsa ya da sürüm GMVER4 değerinden küçükse, MQGET çağrısında geçerli bir ileti tanımlayıcısı sağlamanız gerekir. İleti verilerinde, MQGMO yapısında ve PRPCTL kuyruk özneliğinde özellik seçeneklerinin değerine konu olan ileti özellikleri (ileti tanımlayıcısında bulunan özellikler dışında) döndürülür.

Bu alan her zaman bir giriş alanıdır. Bu alanın ilk değeri HMNONE değeridir. *GMVER* , *GMVER4* değerinden küçükse bu alan yoksayılr.

GMMO (10 basamaklı işaretli tamsayı)

MQGET için kullanılan seçim ölçütlerini denetleyen seçenekler.

Bu seçenekler, uygulamanın MQGET çağrısıyla döndürülen iletiyi seçmek için **MSGDSC** değiştirgesindeki hangi alanların kullanılacağını seçmesini sağlar. Uygulama, bu alanda gerekli seçenekleri ayarlar ve **MSGDSC** parametresindeki karşılık gelen alanları bu alanlar için gerekli olan değerlere ayarlar. Yalnızca, MQMD ' de iletiye ilişkin bu değerleri içeren iletiler, MQGET çağrısında o **MSGDSC** değiştirgesini kullanarak alma adaylarıdır. Karşılık gelen eşleşme seçeneğinin belirlenmediği alanlar, döndürülecek ileti seçilirken yoksayılr. MQGET çağrısında seçim ölçütü kullanılmazsa (yani, herhangi bir ileti kabul edilebilir), *GMMO* MONONE olarak ayarlanmalıdır.

GMLOGO belirtilirse, sonraki MQGET çağrısıyla yalnızca belirli iletiler döndürülebilir:

- Geçerli bir grup ya da mantıksal ileti yoksa, yalnızca *MDSEQ* buna eşit 1 ve *MDOFF* buna eşit 0 iletilere dönüş için uygun olur. Bu durumda, hangi uygun iletilerin döndürüleceğini seçmek için aşağıdaki seçeneklerden biri ya da birkaçı kullanılabilir:
 - MOMSGI
 - MOCORI.
 - MOGRPI.
- Yürürlükteki bir grup ya da mantıksal ileti varsa, yalnızca gruptaki ya da mantıksal iletideki sonraki ileti döndürülmeye uygun olur ve MO* seçenekleri belirtilerek bu değiştirilemez.

Her iki durumda da, geçerli olmayan eşleştirme seçenekleri belirtilebilir, ancak **MSGDSC** parametresindeki ilgili alanın değeri, döndürülecek iletideki ilgili alanın değeriyle eşleşmelidir; çağrı başarısız olur; neden kodu RC2247 ile bu koşul karşılanmaz.

GMMUC ya da GMBRWC belirtilirse *GMMO* yoksayılr.

Aşağıdaki seçeneklerden biri ya da birkaçı belirtilebilir:

MOMSGI

Belirtilen ileti tanıtıcısına sahip iletiyi alın.

Bu seçenek, alınacak iletinin MQGET çağrısının **MSGDSC** değiştirgesindeki *MDMID* alanının değeriyle eşleşen bir ileti tanıtıcısı olması gerektiğini belirtir. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, ilinti tanıtıcısı) ek olarak kullanılır.

Bu seçenek belirtilmezse, **MSGDSC** parametresindeki *MDMID* alanı yoksayılr ve ileti tanıtıcısı eşleşir.

Not: MINONE ileti tanıtıcısı, iletiye ilişkin MQMD ' deki herhangi bir ileti tanıtıcısıyla eşleşen özel bir değerdir. Bu nedenle, MOMSGI değerinin MINONE ile belirlenmesi, MOMSGI değerinin belirlenmemesiyle aynıdır.

MOCORI.

Belirtilen ilinti tanıtıcısına sahip iletiyi alın.

Bu seçenek, alınacak iletinin MQGET çağrısının **MSGDSC** değiştirgesindeki *MDCID* alanının değeriyle eşleşen bir ilinti tanıtıcısı olması gerektiğini belirtir. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, ileti tanıtıcısı) ek olarak bulunur.

Bu seçenek belirtilmezse, **MSGDSC** parametresindeki *MDCID* alanı yoksayılr ve ilinti tanıtıcısı eşleşir.

Not: CINONE ilinti tanıtıcısı, iletiye ilişkin MQMD ' deki herhangi bir ilinti tanıtıcısıyla eşleşen özel bir değerdir. Bu nedenle, MOCORI 'un CINONE ile belirtilmesi MOCORI' un belirtilmemesiyle aynıdır.

MOGRPI.

Belirtilen grup tanıtıcısına sahip iletiyi alın.

Bu seçenek, alınacak iletinin, MQGET çağrısının **MSGDSC** değiştirgesindeki *MDGID* alanının değeriyle eşleşen bir grup tanıtıcısına sahip olması gerektiğini belirtir. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, ilinti tanıtıcısı) ek olarak kullanılır.

Bu seçenek belirtilmezse, **MSGDSC** parametresindeki *MDGID* alanı yoksayılr ve grup tanıtıcısı eşleşir.

Not: GINONE grup tanıtıcısı, iletiye ilişkin MQMD ' deki herhangi bir grup tanıtıcısıyla eşleşen özel bir değerdir. Bu nedenle, MOGRPI 'nin GINONE ile belirtilmesi MOGRPI' nin belirtilmemesiyle aynıdır.

MOSEQN

Belirtilen ileti sıra numarasına sahip iletiyi alın.

Bu seçenek, alınacak iletinin MQGET çağrısının **MSGDSC** değiştirgesindeki *MDSEQ* alanının değeriyle eşleşen bir ileti sıra numarasının olması gerektiğini belirtir. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, grup tanıtıcısı) ek olarak bulunur.

Bu seçenek belirtilmezse, **MSGDSC** parametresindeki *MDSEQ* alanı yoksayılr ve ileti sıra numarası eşleşir.

MOOFFS

Belirtilen görelî konuma sahip iletiyi alın.

Bu seçenek, alınacak iletinin MQGET çağrısının **MSGDSC** değiştirgesindeki *MDOFF* alanının değeriyle eşleşen bir görelî konuma sahip olması gerektiğini belirtir. Bu eşleşme, geçerli olabilecek diğer eşleşmelere (örneğin, ileti sıra numarası) ek olarak bulunur.

Bu seçenek belirtilmezse, **MSGDSC** parametresindeki *MDOFF* alanı yoksayılr ve tüm görelî konumlar eşleşir.

Açıklanan seçeneklerden hiçbiri belirtilmezse, aşağıdaki seçenek kullanılabilir:

MONONE

Eşleşme yok.

Bu seçenek, döndürülecek iletinin seçilmesinde eşleşme kullanılmayacağını belirtir; bu nedenle, kuyruktaki tüm iletiler alınmaya uygundur (ancak, GMAMSA, GMASGA ve GMCMPM seçenekleri tarafından denetlenir).

MONONE, program belgelerine yardımcı olmak için tanımlanır. Bu seçeneğin başka bir MO* seçeneğiyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğundan bu tür bir kullanım algılanamaz.

Bu alan bir giriş alanıdır. Bu alanın ilk değeri MOCORI ile MOMSGI 'dir. *GMVER* , *GMVER2*değerinden küçükse bu alan yoksayılr.

Not: *GMMO* alanının ilk değeri, önceki sürüm kuyruk yöneticileriyle uyumluluk için tanımlanır. Ancak, seçim ölçütleri kullanılmadan bir kuyruktan bir dizi ileti okunurken, bu ilk değer uygulamanın her MQGET çağrısından önce *MDMID* ve *MDCID* alanlarını MINONE ve CINONE olarak sıfırlamasını gerektirir. *MDMID* ve *MDCID* ilk durumuna getirme gereksinimi, *GMVER* *GMVER2*ve *GMMO* ayarı MONONE olarak ayarlanarak önlenir.

GMOPT (10 basamaklı işaretli tamsayı)

MQGET işlemini denetleyen seçenekler.

Aşağıdaki açıklanan seçeneklerden sıfır ya da daha fazlası belirtilebilir. Birden çok değer gerekiyorsa, değerler eklenebilir (aynı değişmezi bir kereden fazla eklemeyin). Geçerli olmayan seçenek birleşimleri belirtilir; diğer tüm birleşimler geçerlidir.

Bekleme seçenekleri: Aşağıdaki seçenekler, iletilerin kuyruğa gelmesini bekleme ile ilgilidir:

GMWT

Mesajın gelmesini bekleyin.

Uygulama, uygun bir ileti gelinceye kadar beklemelidir. Uygulamanın bekleme süresi üst sınırı *GMWT* içinde belirtilir.

MQGET istekleri engellenirse ya da MQGET istekleri bekleme sırasında engellenirse, bekleme iptal edilir ve çağrı CCFAIL ve RC2016neden koduyla tamamlanır; kuyrukta uygun ileti olup olmadığına bakılmaksızın.

Bu seçenek, GMBRWF ya da GMBRWN seçenekleriyle kullanılabilir.

Aynı paylaşılan kuyrukta bekleyen birden çok uygulama varsa, uygulama ya da uygulamalar, bu bölümde daha sonra uygun bir ileti geldiğinde etkinleştirilir.

Not: Aşağıdaki tanımda, göz atma MQGET çağrısı GMLK değil, göz atma seçeneklerinden birini belirtir; GMLK seçeneğini belirten bir MQGET çağrısı göz atmayan bir çağrı olarak işlenir.

- Bir ya da daha fazla göz atmayan MQGET çağrısı bekliyorsa, ancak göz atma MQGET çağrısı beklemezse, bir çağrı etkinleştirilir.
- Bir ya da daha çok göz atma MQGET çağrısı bekliyorsa, ancak göz atmayan MQGET çağrıları beklemezse, tümü etkinleştirilir.
- Bir ya da daha çok MQGET çağrısı ve bir ya da daha çok göz atma MQGET çağrısı bekliyorsa, bir göz atmayan MQGET çağrısı etkinleştirilir ve hiçbiri, bazıları ya da tüm göz atma MQGET çağrıları etkinleştirilir. (Etkinleştirilen göz atma MQGET çağrılarının sayısı, işletim sisteminin ve diğer etkenlerin zamanlamasına bağlı olduğundan tahmin edilemiyor.)

Aynı kuyrukta birden çok göz atmayan MQGET çağrısı bekliyorsa, yalnızca biri etkinleştirilir; bu durumda kuyruk yöneticisi, göz atmayan çağrıları bekleme önceliğini aşağıdaki sırada vermeyi dener:

1. Yalnızca belirli iletiler (örneğin, belirli bir *MDMID* ya da *MDCID* (ya da her ikisi) ile karşılanabilen belirli get-wait istekleri.
2. Herhangi bir ileti tarafından karşılanabilen genel bekleme alma istekleri.

Aşağıdaki noktalar dikkate alınmalıdır:

- İlk kategori içinde, daha belirli alma bekleme isteklerine (örneğin, hem *MDMID* hem de *MDCID* belirtilirler) ek öncelik verilmez.
- Her iki kategori içinde hangi uygulamanın seçildiği tahmin edilemez. Özellikle, en uzun süre bekleyen uygulama mutlaka seçilen uygulama değildir.
- İşletim sisteminin yol uzunluğu ve öncelik zamanlaması ile ilgili önemli noktalar, beklenenden daha düşük işletim sistemi önceliğine sahip bir bekleme uygulamasının iletiyi aldığı anlamına gelebilir.
- Beklemeyen bir uygulama, iletiyi olduğu gibi almayı tercih eden bir uygulama da olabilir.

GMWT, GMBRWC ya da GMMUC ile belirtilirse yoksayılar; hata oluşmaz.

GMNWT

Uygun bir ileti yoksa hemen geri dönün.

Uygun bir ileti yoksa uygulama beklemez. Bu, GMWT seçeneğinin karşıtıdır ve program belgelerine yardımcı olmak için tanımlanır. İkisi de belirtilmezse, varsayılan değer budur.

GMFIQ

Kuyruk yöneticisi susturulursa başarısız olur.

Bu seçenek, kuyruk yöneticisi susturma durumundaysa MQGET çağrısı başarısız etmeye zorlar.

Bu seçenek GMWT ile birlikte belirtilirse ve kuyruk yöneticisi susturma durumuna girdiğinde bekleme işlemi tamamlanmazsa:

- Bekleme iptal edilir ve çağrı, RC2161 neden koduyla CCFAIL tamamlanma kodunu döndürür.

GMFIQ belirtilmezse ve kuyruk yöneticisi susturma durumuna girerse, bekleme iptal edilmez.

Eşitleme noktası seçenekleri: Aşağıdaki seçenekler, MQGET çağrısının bir iş birimi içindeki katılımıyla ilgilidir:

GMSYP

Eşitleme noktası denetimine sahip iletiyi alın.

İstek, normal iş birimi protokolleri içinde çalışmaktadır. İleti, diğer uygulamalar tarafından kullanılamıyor olarak işaretlenir, ancak yalnızca iş birimi kesinleştirildiğinde kuyruktan silinir. İş birimi getirildiyse, ileti yeniden kullanılabilir kılınır.

Bu seçenek ya da GMNSYP belirtilmezse, alma isteği bir iş birimi içinde değildir.

Bu seçenek, aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- GMBRWF
- GMBRWC
- GMBRWN
- GMLK
- GMNSYP
- GMPSYP
- GMUNLK

GMPSYP

İleti kalıcıysa, eşitleme noktası denetimine sahip iletiyi alın.

İstek, normal iş birimi iletişim kuralları içinde çalışmaktadır, ancak yalnızca alınan ileti kalıcı ise. Kalıcı bir ileti, MQMD 'deki *MDPER* alanında *PEPER* değerini içeriyor.

- İleti kalıcıysa, kuyruk yöneticisi çağrısı uygulama GMSYP 'yi belirtmiş gibi işler.
- İleti kalıcı değilse, kuyruk yöneticisi çağrısı uygulamanın GMNSYP belirttiği gibi işler (ayrıntılar için aşağıdaki bölüme bakın).

Bu seçenek, aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- GMBRWF
- GMBRWC
- GMBRWN
- GMCMPM
- GMNSYP
- GMSYP
- GMUNLK

GMNSYP

Eşitleme noktası denetimi olmadan ileti alın.

İstek, normal iş birimi protokollerinin dışında çalışmaktadır. İleti hemen kuyruktan silinir (bu bir göz atma isteği değilse). İş birimi yedeklenerek ileti yeniden kullanılabilir kılınmaz.

Bu seçenek, GMBRWF ya da GMBRWN belirtilirse kabul edilir.

Bu seçenek ve GMSYP belirtilmezse, alma isteği bir iş birimi içinde değildir.

Bu seçenek, aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- GMSYP
- GMPSYP

Göz atma seçenekleri: Aşağıdaki seçenekler, kuyruktaki iletilere göz atma ile ilgilidir:

GMBRWF

Kuyruğun başlangıcından itibaren göz atın.

Bir kuyruk OOBROW seçeneğiyle açıldığında, mantıksal olarak kuyruktaki ilk iletiden önce konumlandırılmış bir göz atma imleci oluşturulur. GMBRWF, GMBRWN ya da GMBRWC seçeneğini belirten sonraki MQGET çağrıları, kuyruktan iletileri yıkıcı olmadan almak için kullanılabilir. Göz atma imleci, kuyruktaki iletiler içinde, GMBRWN ile birlikte bir sonraki MQGET çağrısından uygun bir ileti arandığı konumu işaretler.

GMBRWF içeren bir MQGET çağrısı, göz atma imlecinin önceki konumunun yoksayılmasına neden olur. Kuyruktaki, ileti tanımlayıcısında belirtilen koşullara uyan ilk ileti alınır. İleti kuyrukta kalır ve göz atma imleci bu iletide konumlandırılır.

Bu çağrıdan sonra, göz atma imleci döndürülen iletide konumlandırılır. İleti, GMBRWN ile bir sonraki MQGET çağrısı yayınlanmadan önce kuyruktan kaldırılırsa, göz atma imleci, bu konum boş olsa da, iletinin bulunduğu kuyrukta kalır.

GMMUC seçeneği, gerekiyorsa, iletiyi kuyruktan kaldırmak için göz atmayan bir MQGET çağrısıyla birlikte kullanılabilir.

Göz atma imleci, aynı HOBJ tanıtıcısı kullanılarak, göz atmayan bir MQGET çağrısıyla taşınmaz. CCFAIL ' in tamamlanma kodunu ya da RC2080 neden kodunu döndüren bir göz atma MQGET çağrısıyla da taşınmaz.

GMLK seçeneği, göz atılan iletinin kilitletmesini sağlamak için bu seçenekle birlikte belirtilebilir.

GMBRWF, gruplar ve mantıksal ileti kesimleri içindeki iletilerin işlenmesini denetleyen GM* ve MO* seçeneklerinin geçerli bir birleşimiyle belirtilebilir.

GMLOGO belirtilirse, iletilere mantıksal sırayla göz atılır. Bu seçenek atlanırsa, iletilere fiziksel sırayla göz atılır. GMBRWF belirtildiğinde, mantıksal sıra ile fiziksel sıra arasında geçiş yapılabilir, ancak GMBRWN kullanılarak yapılan sonraki MQGET çağrıları, kuyruk tanıtıcısı için GMBRWF ' yi belirten en son çağrıyla aynı sırada kuyruğa göz atmalıdır.

Kuyruk yöneticisinin kuyruktaki iletilere göz atayan MQGET çağrıları için sakladığı grup ve bölüm bilgileri, kuyruk yöneticisinin kuyruktan ileti kaldran MQGET çağrıları için sakladığı grup ve bölüm bilgilerinden farklıdır. GMBRWF belirtildiğinde, kuyruk yöneticisi göz atmak için grup ve bölüm bilgilerini yoksayar ve kuyruğu, yürürlükteki grup ve yürürlükteki mantıksal ileti yokmuş gibi tarar. MQGET çağrısı başarılı olursa (tamamlama kodu CCOK ya da CCWARN), göz atmaya ilişkin grup ve bölüm bilgileri döndürülen iletiye ayarlanır; çağrı başarısız olursa, grup ve bölüm bilgileri çağrıdan önceki gibi kalır.

Bu seçenek, aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- GMBRWC
- GMBRWN
- GMMUC
- GMSYP
- GMPSYP
- GMUNLK

Kuyruk göz atmak için açılmamışsa da bu bir hatadır.

GMBRWN

Kuyruktaki geçerli konumdan göz atın.

Göz atma imleci, kuyrukta MQGET çağrısında belirtilen seçim ölçütlerini karşılayan sonraki iletiye iletilemiştir. İleti uygulamaya döndürülür, ancak kuyrukta kalır.

Göz atmak üzere bir kuyruk açıldıktan sonra, tanıtıcı kullanılarak yapılan ilk göz atma çağrısı, GMBRWF ya da GMBRWN seçeneğini belirlese de aynı etkiye sahiptir.

İleti, GMBRWN ile bir sonraki MQGET çağrısı yayınlanmadan önce kuyruktan kaldırılırsa, göz atma imleci artık boş olsa da, mantıksal olarak iletinin bulunduğu kuyrukta kalır.

İletiler şu iki yoldan biriyle kuyrukta saklanır:

- Öncelik içinde FIFO (MSPRIO) ya da
- Önceliğe bakılmaksızın FIFO (MSFIFO)

MsgDeliverySequence kuyruk özneliği hangi yöntemin geçerli olduğunu gösterir (ayrıntılar için bkz. “Kuyruklara ilişkin öznelikler” sayfa 1345).

Kuyrukta bir *MsgDeliverySequence* /MSPRIO varsa ve kuyruğa, göz atma imleci tarafından gösterilenden daha yüksek önceliğe sahip bir ileti gelirse, bu ileti, GMBRWN kullanılarak kuyruğun yürürlükteki taraması sırasında bulunmaz. Yalnızca, göz atma imleci GMBRWF ile ilk durumuna getirildikten sonra (ya da kuyruk yeniden açılarak) bulunabilir.

GMMUC seçeneği, gerekiyorsa, iletiyi kuyruktan kaldırmak için göz atmayan bir MQGET çağrısıyla birlikte kullanılabilir.

Göz atma imleci, aynı *HOB*J tanıtıcısı kullanılarak MQGET çağrılarıyla taşınmaz.

GMLK seçeneği, göz atılan iletinin kilitletmesini sağlamak için bu seçenekle birlikte belirtilebilir.

GMBRWN, iletilerin gruplar ve mantıksal ileti kesimleri içinde işlenmesini denetleyen GM* ve MO* seçeneklerinin geçerli bir birleşimiyle belirtilebilir.

GMLOGO belirtilirse, iletilere mantıksal sırayla göz atılır. Bu seçenek atlanırsa, iletilere fiziksel sırayla göz atılır. GMBRWF belirtildiğinde, mantıksal sıra ile fiziksel sıra arasında geçiş yapılabilir, ancak GMBRWN kullanılarak yapılan sonraki MQGET çağrıları, kuyruk tanıtıcısı için GMBRWF ' yi belirten en son çağrıyla aynı sırada kuyruğa göz atmalıdır. Bu koşul karşılanmazsa, çağrı RC2259 neden koduyla başarısız olur.

Not: GMLOGO belirtilmediğinde bir ileti grubunun sonuna (ya da bir grupta olmayan mantıksal iletiye) göz atmak için bir MQGET çağrısı kullanılırsa özel önem gerekir. Örneğin, gruptaki son ileti kuyruktaki gruptaki ilk iletiden önce geliyorsa, grubun sonunun ötesine göz atmak için GMBRWN kullanılarak, MOSEQ için *MDSEQ* set 1 değeri belirlenirse (sonraki grubun ilk iletisini bulmak için), gruptaki ilk iletiye yeniden göz atılır. Bu durum hemen oluşabilir ya da daha sonra (araya giren gruplar varsa) MQGET çağrıları olabilir.

Göz atmak için kuyruk iki kez açılarak sonsuz döngü olasılığı önlenir:

- Her gruptaki yalnızca ilk iletiye göz atmak için ilk tanıtıcıyı kullanın.
- Yalnızca belirli bir gruptaki iletilere göz atmak için ikinci tanıtıcıyı kullanın.
- Gruptaki iletilere göz atmadan önce, ikinci göz atma imlecini ilk göz atma imlecinin konumuna taşımak için MO* seçeneklerini kullanın.
- Bir grubun sonundan sonrasına göz atmak için GMBRWN seçeneğini kullanmayın.

Kuyruk yöneticisinin kuyruktaki iletilere göz atayan MQGET çağrıları için sakladığı grup ve bölüm bilgileri, kuyruktan ileti kaldran MQGET çağrıları için sakladığı grup ve bölüm bilgilerinden farklıdır.

Bu seçenek, aşağıdaki seçeneklerin hiçbirleriyle geçerli değildir:

- GMBRWF
- GMBRWC
- GMMUC
- GMSYP
- GMPSYP

- GMUNLK

Kuyruk göz atmak için açılmamışsa da bu bir hatadır.

GMBRWC

Göz atma imlecinin altındaki iletiye göz atın.

Bu seçenek, MQGMO 'daki *GMMO* alanında belirtilen MO* seçeneklerinden bağımsız olarak, göz atma imlecinin gösterdiği iletinin yıkıcı olmayan bir şekilde alınmasına neden olur.

Göz atma imlecinin gösterdiği ileti, en son GMBRWF ya da GMBRWN seçeneği kullanılarak alınan iletidir. Açıldığından beri bu kuyruk için bu çağrılarının hiçbiri yayınlanmamışsa ya da göz atma imlecinin altındaki ileti yıkıcı bir şekilde alındıysa, çağrı başarısız olur.

Göz atma imlecinin konumu bu çağrı tarafından değiştirilmez.

GMMUC seçeneği, gerekiyorsa, iletiyi kuyruktan kaldırmak için göz atmayan bir MQGET çağrısıyla birlikte kullanılabilir.

Göz atma imleci, aynı *HOBJ* tanıtıcısı kullanılarak, göz atmayan bir MQGET çağrısıyla taşınmaz. Ayrıca, CCFAIL 'in tamamlanma kodunu ya da RC2080neden kodunu döndüren bir göz atma MQGET çağrısıyla da taşınmaz.

GMLK ile GMBRWC belirtilirse:

- Kilitli bir ileti varsa, bu ileti imlecin altındaki ileti olmalıdır; bu ileti kilidi açılmadan ve yeniden kilitlenmeden döndürülür; ileti kilitli kalır.
- Kilitli ileti yoksa, göz atma imlecinin altındaki ileti (varsa) kilitlenir ve uygulamaya döndürülür; göz atma imlecinin altında ileti yoksa çağrı başarısız olur.

GMBRWC GMLK olmadan belirtilirse:

- Kilitli bir ileti varsa, imlecin altındaki ileti olmalıdır. Bu ileti uygulamaya döndürülür ve kilidi açılır. İletin kilidi açıldığından, iletiye yeniden göz atılabileceğinin ya da yıkıcı bir şekilde alınabileceğinin bir garantisi yoktur (kuyruktan ileti alan başka bir uygulama tarafından yıkıcı bir şekilde alınabilir).
- Kilitli ileti yoksa, göz atma imlecinin altındaki ileti (varsa) uygulamaya döndürülür; göz atma imlecinin altında ileti yoksa, çağrı başarısız olur.

GMBRWC ile GMCMPM belirtilirse, göz atma imleci MQMD 'de sıfır olan bir *MDOFF* alanıyla bir iletiyi tanımlamalıdır. Bu koşul karşılanmazsa, arama RC2246 neden koduyla başarısız olur.

Kuyruk yöneticisinin kuyruktaki iletilere göz atayan MQGET çağrıları için sakladığı grup ve bölüm bilgileri, kuyruktan ileti kaldran MQGET çağrıları için sakladığı grup ve bölüm bilgilerinden farklıdır.

Bu seçenek, aşağıdaki seçeneklerin hiçbiriyle geçerli değildir:

- GMBRWF
- GMBRWN
- GMMUC
- GMSYP
- GMPSYP
- GMUNLK

Kuyruk göz atmak için açılmamışsa da bu bir hatadır.

GMMUC

İletiyi göz atma imlecinin altına alın.

Bu seçenek, MQGMO 'nun *GMMO* alanında belirtilen MO* seçeneklerinden bağımsız olarak, göz atma imlecinin gösterdiği iletinin alınmasına neden olur. İleti kuyruktan kaldırılır.

Göz atma imlecinin gösterdiği ileti, en son GMBRWF ya da GMBRWN seçeneği kullanılarak alınan iletidir.

GMCMPPM GMMUC ile belirtilirse, göz atma imleci MQMD ' de sıfır olan bir MDOFF alanıyla bir iletiyi tanımlamalıdır. Bu koşul karşılanmazsa, arama RC2246 neden koduyla başarısız olur.

Bu seçenek, aşağıdaki seçeneklerin hiçbirleriyle geçerli değildir:

- GMBRWF
- GMBRWC
- GMBRWN
- GMUNLK

Kuyruk hem göz atma hem de giriş için açılmamışsa da bu bir hatadır. Göz atma imleci şu anda alınabilir bir iletiyi göstermezse, MQGET çağrısıyla bir hata döndürülür.

Kilit seçenekleri: Aşağıdaki seçenekler, kuyruktaki iletilerin kilitlenmesiyle ilgilidir:

GMLK

İletiyi kilitler.

Bu seçenek, göz atılan iletiyi kilitler; böylece ileti, kuyruk için açık olan başka bir tanıtıcı tarafından görünmez olur. Bu seçenek yalnızca aşağıdaki seçeneklerden biri de belirtilirse belirtilebilir:

- GMBRWF
- GMBRWN
- GMBRWC

Kuyruk tanıtıcısı başına yalnızca bir ileti kilitlenebilir, ancak bu bir mantıksal ileti ya da fiziksel ileti olabilir:

- GMCMPPM belirtilirse, mantıksal iletiyi oluşturan tüm ileti kesimleri kuyruk tanıtıcısı için kilitlenir (kuyrukta varsa ve alınabiliyorsa).
- GMCMPPM belirtilmezse, kuyruk tanıtıcısına yalnızca tek bir fiziksel ileti kilitlenir. Bu ileti mantıksal iletinin bir parçası olursa, kilitli bölüm, GMCMPPM kullanan diğer uygulamaların mantıksal iletiyi almasını ya da mantıksal iletiye göz atmasını engeller.

Kilitli ileti her zaman göz atma imlecinin altındaki ilettir ve ileti, GMMUC seçeneğini belirten daha sonraki bir MQGET çağrısıyla kuyruktan kaldırılabilir. Kuyruk tanıtıcısını kullanan diğer MQGET çağrıları da iletiyi kaldırabilir (örneğin, kilitli iletinin ileti tanıtıcısını belirten bir çağrı).

Çağrı, RC2080neden koduyla CCFAIL ya da CCWARN tamamlanma kodunu döndürürse, ileti kilitlenmez.

Uygulama iletiyi kuyruktan kaldırmamaya karar verirse, kilit aşağıdaki şekilde serbest bırakılır:

- Bu tanıtıcı için GMBRWF ya da GMBRWN belirtilerek (GMLK ile ya da olmadan) başka bir MQGET çağrısı yayınlanıyor; çağrı CCOK ya da CCWARN ile tamamlanırsa, ancak çağrı CCFAIL ile tamamlanırsa, iletinin kilidi açılır. Ancak, aşağıdaki özel durumlar geçerlidir:
 - RC2080ile CCWARN döndürülürse iletinin kilidi açılmaz.
 - RC2033ile CCFAIL döndürülürse iletinin kilidi açılır.

GMLK de belirtilirse, döndürülen ileti kilitlenir. GMLK belirtilmezse, aramadan sonra kilitli ileti yoktur.

GMWT belirtilirse ve hemen ileti yoksa, özgün iletideki kilit bekleme başlamadan önce açılır (çağrı hatasız değilse).

- Bu tanıtıcı için GMBRWC (GMLK olmadan) başka bir MQGET çağrısı yayınlanıyor; çağrı CCOK ya da CCWARN ile tamamlanırsa, ancak çağrı CCFAIL ile tamamlanırsa, ileti kilitli kalıyor. Ancak, aşağıdaki kural dışı durum geçerlidir:
 - RC2080ile CCWARN döndürülürse iletinin kilidi açılmaz.
- Bu tanıtıcı için GMUNLK ile başka bir MQGET çağrısı yayınlanıyor.
- Bu tanıtıcı için bir MQCLOSE çağrısı yayınlanıyor (belirtik ya da örtük olarak uygulama sona eriyor).

Bu seçeneği belirlemek için OOBRW dışında özel bir açma seçeneği gerekmez; bu seçenek, eşlik eden göz atma seçeneğini belirlemek için gereklidir.

Bu seçenek, aşağıdaki seçeneklerin hiçbirisiyle geçerli değildir:

- GMSYP
- GMPSYP
- GMUNLK

GMUNLK

İletinin kilidini açın.

Kilidinin açılacak ileti, GMLK seçeneğiyle bir MQGET çağrısı tarafından önceden kilitlemiş olmalıdır. Bu tanıtıcı için kilitli bir ileti yoksa, çağrı CCWARN ve RC2209 ile tamamlanır.

GMUNLK belirtilirse, **MSGDSC**, **BUFLN**, **BUFFER** ve **DATLEN** değiştiricileri denetlenmez ya da değiştirilmez. *BUFFER* içinde herhangi bir ileti döndürülmez.

Bu seçeneği belirlemek için özel bir açma seçeneği gerekmez (ancak kilit isteğini ilk olarak yayınlamak için OOBRW gereklidir).

Bu seçenek, aşağıdakiler dışında hiçbir seçenekle geçerli değildir:

- GMNWT
- GMNSYP

Bu seçeneklerin her ikisi de belirtilmiş ya da belirtilmemiş olarak kabul edilir.

İleti-veri seçenekleri: Aşağıdaki seçenekler, ileti kuyruktan okunduğunda ileti verilerinin işlenmesiyle ilgilidir:

GMATM

İleti verilerinin kesilmesine izin ver.

İleti arabelleği tüm iletiyi tutamayacak kadar küçükse, bu seçenek MQGET çağrısının arabelleği arabellek kadar iletiyi doldurmasına, bir uyarı tamamlama kodu yayınlamasına ve işlenmesini tamamlamasına izin verir. Bunun anlamı:

- İletilere göz atarken, göz atma imleci döndürülen iletiye ilerler.
- İletiler kaldırılırken, döndürülen ileti kuyruktan kaldırılır.
- Başka bir hata oluşmazsa, neden kodu RC2079 döndürülür.

Bu seçenek olmadan, arabellek, tutabildiği kadar ileti ile doldurulur, bir uyarı tamamlama kodu verilir, ancak işlem tamamlanmaz. Bunun anlamı:

- İletilere göz atarken, göz atma imleci gelişmiş değildir.
- İletiler kaldırılırken, ileti kuyruktan kaldırılmaz.
- Başka bir hata oluşmazsa, neden kodu RC2080 döndürülür.

GMCONV

İleti verilerini dönüştürün.

Bu seçenek, veriler **BUFFER** değiştiricisine kopyalanmadan önce, iletideki uygulama verilerinin MQGET çağrısındaki **MSGDSC** değiştiricisinde belirtilen *MDCSI* ve *MDENC* değerlerine uygun olarak dönüştürülmesini ister.

İletinin konması sırasında belirtilen *MDFMT* alanı, iletideki verilerin türünü tanımlamak için dönüştürme işlemi tarafından kabul edilir. İleti verilerinin dönüştürülmesi, yerleşik biçimler için kuyruk yöneticisi tarafından ve diğer biçimler için kullanıcı tarafından yazılan bir çıkış tarafından gerçekleştirilmiştir.

- Dönüştürme başarılı bir şekilde gerçekleştirilirse, **MSGDSC** değiştiricisinde belirtilen *MDCSI* ve *MDENC* alanları MQGET çağrısından dönüşte değişmez.

- Dönüştürme başarılı bir şekilde gerçekleştirilemezse (ancak MQGET çağrısı hatasız tamamlanırsa), ileti verileri dönüştürülmeden döndürülür ve MSGDSC içindeki MDCSI ve MDENC alanları, dönüştürülmemiş iletinin değerlerine ayarlanır. Bu durumda tamamlanma kodu CCWARN olur.

Her iki durumda da, bu alanlar **BUFFER** deęiştirgesinde döndürülen ileti verilerinin karakter kümesi tanıtıcısını ve kodlamasını açıklar.

Kuyruk yöneticisinin dönüştürmeyi gerçekleştirdięi biçim adlarının bir listesi için "IBM i üzerinde MQMD (İleti tanımlayıcı)" sayfa 1093 içinde açıklanan *MDFMT* alanına bakın.

Grup ve bölüm seçenekleri: Aşağıdaki seçenekler, iletilerin gruplar ve mantıksal ileti kesimleri halinde işlenmesiyle ilgilidir. Bu tanımlar, seçeneklerin anlaşılmasına yardımcı olabilir:

Fiziksel ileti

Bu, bir kuyruęa yerleştirebilecek ya da kuyruktan kaldırılabilir en küçük bilgi birimidir; genellikle tek bir MQPUT, MQPUT1 ya da MQGET çağrısında belirtilen ya da alınan bilgilere karşılık gelir. Her fiziksel iletinin kendi ileti tanımlayıcısı (MQMD) vardır. Genellikle, fiziksel iletiler ileti tanıtıcısı (MQMD 'deki *MDMID* alanı) için farklı değerlerle ayırt edilir, ancak bu kuyruk yöneticisi tarafından zorlanmaz.

Mantıksal ileti

Bu, tek bir uygulama bilgisi birimidir. Sistem kısıtlamaları olmadığında, mantıksal bir ileti fiziksel bir iletiyle aynı olur. Ancak, mantıksal iletilerin büyük olduęu durumlarda, sistem kısıtlamaları, mantıksal bir iletiyi bölümler adı verilen iki ya da daha fazla fiziksel iletiye bölmeyi uygun ya da gerekli kılabilir.

Bölümlenmiş bir mantıksal ileti, boş olmayan grup tanıtıcısı (MQMD 'de *MDGID* alanı) aynı olan iki ya da daha çok fiziksel iletiden ve aynı ileti sıra numarasından (MQMD' de *MDSEQ* alanı) oluşur. Kesimler, mantıksal iletideki verilerin başlangıcından fiziksel iletideki verilerin görelî konumunu veren kesim görelî konumu (MQMD ' de *MDOFF* alanı) için farklı değerlerle ayırt edilir. Her bir bölüm fiziksel bir ileti olduęu için, mantıksal bir iletideki bölümlerin genellikle farklı ileti tanıtıcıları vardır.

Bölümlenmemiş, ancak bölümlenmeye gönderen uygulama tarafından izin verilen mantıksal bir ileti, boş olmayan bir grup tanıtıcısına da sahiptir; ancak bu durumda, mantıksal ileti bir ileti grubuna ait deęilse, bu grup tanıtıcısına sahip yalnızca bir fiziksel ileti vardır. Mantıksal ileti bir ileti grubuna ait deęilse, bölümlenme gönderen uygulama tarafından engellenen mantıksal iletilerin boş deęerli bir grup tanıtıcısı (GINONE) vardır.

İleti grubu

Bu, aynı boş olmayan grup tanıtıcısına sahip bir ya da daha fazla mantıksal iletiden oluşan bir kümedir. Gruptaki mantıksal iletiler, 1-n aralığında bir tamsayı olan ileti sıra numarasına ilişkin farklı deęerlerle ayırt edilir; burada n, gruptaki mantıksal ileti sayısıdır. Mantıksal iletilerden biri ya da daha fazlası bölümlenmişse, grupta n 'den fazla fiziksel ileti vardır.

GMLOGO

Mantıksal ileti grupları ve kesimleri içindeki iletiler mantıksal sırayla döndürülür.

Bu seçenek, kuyruk tanıtıcısı için ardışık MQGET çağruları tarafından iletilerin döndürüleceęi sırayı denetler. Bir etki yaratmak için bu çağrılarının her birinde seçenek belirtilmelidir.

Kuyruk tanıtıcısı için ardışık MQGET çağruları için GMLOGO belirtilirse, gruplar içindeki iletiler ileti sırası numaralarına göre döndürülür ve mantıksal ileti bölümleri, kesim görelî konumlarına göre verilen sırayla döndürülür. Bu sıra, bu ileti ve bölümlerin kuyrukta oluşma sırasından farklı olabilir.

Not: GMLOGO belirtilmesi, gruplara ait olmayan ve bölüm olmayan iletiler üzerinde olumsuz sonuçlar doğurmaz. Gerçekte, bu tür iletilerin her biri yalnızca bir iletiden oluşan bir ileti grubuna aitmiş gibi ele alınır. Bu nedenle, gruplarda, ileti bölümlerinde ve gruplarda olmayan bölümlenmemiş iletilerde ileti karışımı içerebilecek kuyruklardan ileti alınırken GMLOGO belirtilmesi son derece güvenlidir.

İletileri istenen sırayla döndürmek için kuyruk yöneticisi, ardışık MQGET çağruları arasındaki grup ve bölüm bilgilerini korur. Bu bilgiler, kuyruk tanıtıcısı için yürürlükteki ileti grubunu ve yürürlükteki mantıksal iletiyi, grup ve mantıksal ileti içindeki yürürlükteki konumu ve iletilerin bir iş birimi

içinde alınıp alınmadığını tanımlar. Kuyruk yöneticisi bu bilgileri sakladığı için, uygulamanın her MQGET çağrısından önce grup ve bölüm bilgilerini ayarlamasına gerek yoktur. Bu, uygulamanın MQMD ' de *MDGID*, *MDSEQ* ve *MDOFF* alanlarını ayarlamasına gerek olmadığı anlamına gelir. Ancak, uygulamanın her çağrıda *GMSYP* ya da *GMNSYP* seçeneğini doğru olarak ayarlaması gerekir.

Kuyruk açıldığında, yürürlükteki ileti grubu ve yürürlükteki mantıksal ileti yok. *MFMIG* işareti olan bir ileti MQGET çağrısı tarafından döndürüldüğünde, ileti grubu yürürlükteki ileti grubu olur. *GMLOGO* art arda çağrılarda belirtildiğinde, aşağıdaki özelliklere sahip bir ileti döndürülünceye kadar bu grup geçerli grup olarak kalır:

- *MFSEG* olmadan *MFLMIG* (yani, gruptaki son mantıksal ileti bölümlenmez) ya da
- *MFLSEG* ile *MFLMIG* (yani, döndürülen ileti, gruptaki son mantıksal iletinin son bölümüdür).

Böyle bir ileti döndürüldüğünde, ileti grubu sonlandırılır ve bu MQGET çağrısı başarıyla tamamlandığında artık yürürlükteki bir grup olmaz. Benzer bir şekilde, MQGET çağrısı *MFSEG* işaretine sahip bir ileti döndürdüğünde ve *MFLSEG* işaretine sahip ileti döndürüldüğünde mantıksal ileti geçerli mantıksal ileti olur.

Seçim ölçütü belirtilmezse, ardışık MQGET çağrıları, kuyruktaki ilk ileti grubuna ilişkin iletileri (doğru sırada) döndürür ve ikinci ileti grubuna ilişkin iletiler, kullanılabilir ileti kalmancaya kadar devam eder. *GMMO* alanında aşağıdaki seçeneklerden birini ya da birkaçını belirterek, döndürülen ileti grupları seçilebilir:

- *MOMSGI*
- *MOCORI*.
- *MOGRPI*.

Ancak, bu seçenekler yalnızca geçerli ileti grubu ya da mantıksal ileti olmadığında geçerlidir; bu konuda açıklanan *GMMO* alanına bakın.

Çizelge 702 sayfa 1072 içinde, kuyruk yöneticisinin MQGET çağrısında döndürülecek bir iletiyi bulma girişimi sırasında aradığı *MDMID*, *MDCID*, *MDGID*, *MDSEQ* ve *MDOFF* alanlarının değerleri gösterilir. Bu, kuyruktaki iletilerin kaldırılmasına ve kuyruktaki iletilere göz atılmasına uygulanır. Çizelgedeki kolonlar aşağıdaki anlamlara sahiptir:

Günlük ORDı

Çağrıda *GMLOGO* seçeneğinin belirtilip belirtilmediğini belirtir.

Cur grp

Aramadan önce geçerli bir ileti grubunun var olup olmadığını belirtir.

Cur günlük iletisi

Çağrıda önce geçerli bir mantıksal iletinin bulunup bulunmadığını belirtir.

Diğer sütunlar

Kuyruk yöneticisinin aradığı değerleri gösterin. "Önceki", kuyruk tanıtıcısı için önceki iletide alan için döndürülen değeri belirtir.

Çizelge 702. Mantıksal ileti grupları ve bölümlerindeki iletilerle ilgili MQGET seçenekleri							
Belirled iğiniz seçene- kler	Çağrıda önce grup ve günlük- iletisi durumu		Kuyruk yöneticisinin aradığı değerler				
	Cur grp	Cur günlük iletisi	<i>MDMID</i>	<i>MDCID</i>	<i>MDGID</i>	<i>MDSEQ</i>	<i>MDOFF</i>
Evet	Hayır	Hayır	<i>GMMO</i> tarafından denetlenir	<i>GMMO</i> tarafından denetlenir	<i>GMMO</i> tarafından denetlenir	1	0

Çizelge 702. Mantıksal ileti grupları ve bölümlerindeki iletilerle ilgili MQGET seçenekleri (devamı var)

Belirled iğiniz seçene kler	Çağrıdan önce grup ve günlük- ileti durumu		Kuyruk yöneticisinin aradığı değerler				
	Evet	Hayır	Evet	Herhangi bir ileti tanıtıcısı	Herhangi bir ilinti tanıtıcısı	Önceki grup tanıtıcısı	1
Evet	Evet	Hayır	Herhangi bir ileti tanıtıcısı	Herhangi bir ilinti tanıtıcısı	Önceki grup tanıtıcısı	Önceki sıra numarası + 1	0
Evet	Evet	Evet	Herhangi bir ileti tanıtıcısı	Herhangi bir ilinti tanıtıcısı	Önceki grup tanıtıcısı	Önceki sıra numarası	Önceki görel konum + önceki kesim uzunluğu
Hayır	Herhan gi biri	Herhan gi biri	GMMO tarafından denetlenir	GMMO tarafından denetlenir	GMMO tarafından denetlenir	GMMO tarafından denetlenir	GMMO tarafından denetlenir

Kuyrukta birden çok ileti grubu olduğunda ve iade edilmeye uygun olduğunda, gruplar, her bir gruptaki ilk mantıksal iletinin ilk kesiminin kuyruğundaki konumuna göre belirlenen sırayla döndürülür (yani, ileti sıra numaraları 1 olan fiziksel iletiler ve görel konumlar 0 ise, uygun grupların döndürüleceği sırayı belirler).

GMLOGO seçeneği, iş birimlerini aşağıdaki gibi etkiler:

- Bir gruptaki ilk mantıksal ileti ya da bölüm bir iş birimi içinde alındıysa, aynı kuyruk tanıtıcısı kullanılırsa, gruptaki diğer tüm mantıksal iletiler ve kesimler bir iş birimi içinde alınmalıdır. Ancak, aynı iş birimi içinde alınmaları gerekmez. Bu, birçok fiziksel iletiden oluşan bir ileti grubunun, kuyruk tanıtıcısı için iki ya da daha fazla ardışık iş birimine bölünmesini sağlar.
- Bir gruptaki ilk mantıksal ileti ya da bölüm bir iş birimi içinde alınmazsa, aynı kuyruk tanıtıcısı kullanılırsa, gruptaki diğer mantıksal iletilerin ve kesimlerin hiçbiri bir iş birimi içinde alınamaz.

Bu koşullar karşılanmazsa, MQGET çağrısı RC2245 neden koduyla başarısız olur.

GMLOGO belirtildiğinde, MQGET çağrısında belirtilen MQGMO, GMVER2 değerinden küçük olmamalı ve MQMD, MDVER2 değerinden küçük olmamalıdır. Bu koşul karşılanmazsa, arama RC2256 ya da RC2257 neden koduyla başarısız olur.

Kuyruk tanıtıcısı için ardışık MQGET çağrıları için GMLOGO belirtilmezse, iletiler ileti gruplarına ait olup olmadıklarına ya da mantıksal ileti bölümleri olup olmadıklarına bakılmaksızın döndürülür. Bu, belirli bir grup ya da mantıksal iletiden gelen iletilerin ya da bölümlerin sırasız olarak döndürülebileceği ya da diğer gruplardan ya da mantıksal iletilerden gelen iletilerle ya da gruplarda olmayan ve bölüm olmayan iletilerle karışabileceği anlamına gelir. In this situation, the particular messages that are returned by successive MQGET calls is controlled by the MO* options specified on those calls (see the GMMO field described in “IBM i üzerinde MQGMO (Get-message seçenekleri)” sayfa 1060 for details of these options).

Bu, bir sistem arızası oluştuğundan sonra ortada bir ileti grubunu ya da mantıksal iletiyi yeniden başlatmak için kullanılacak bir tekniktir. Sistem yeniden başlatıldığında, uygulama MDGID, MDSEQ, MDOFF ve GMMO alanlarını uygun değerlere ayarlayabilir ve daha sonra, GMLOGO belirtilmeden, GMSYP ya da GMNSYP ile MQGET çağrısı yürütebilir. Bu çağrı başarılı olursa, kuyruk yöneticisi grup ve bölüm bilgilerini saklar ve bu kuyruk tanıtıcısını kullanan sonraki MQGET çağrıları olağan olarak GMLOGO değerini belirtebilir.

Kuyruk yöneticisinin MQGET çağrısı için sakladığı grup ve bölüm bilgileri, MQPUT çağrısı için sakladığı grup ve bölüm bilgilerinden ayrıdır. Ayrıca, kuyruk yöneticisi aşağıdakilere ilişkin ayrı bilgileri de saklar:

- Kuyruktan ileti kaldran MQGET çağrıları.
- Kuyruktaki iletilere göz atayan MQGET çağrıları.

Belirli bir kuyruk tanıtıcısı için, uygulama GMLOGO ' u belirten MQGET çağrılarını MQGET çağrılarıyla karışık olarak kullanmakta serbesttir, ancak aşağıdaki noktalar dikkate alınmalıdır:

- GMLOGO belirtilmezse, her başarılı MQGET çağrısı, kuyruk yöneticisinin kaydedilen grup ve bölüm bilgilerini döndürülen iletiye karşılık gelen değerlere ayarlamasına neden olur; bu, kuyruk tanıtıcısı için kuyruk yöneticisi tarafından tutulan var olan grup ve bölüm bilgilerinin yerini alır. Yalnızca çağrı işlemine uygun bilgiler (göz at ya da kaldır) değiştirilir.
- GMLOGO belirtilmezse, yürürlükteki ileti grubu ya da mantıksal ileti varsa çağrı başarısız olmaz; ancak, çağrı CCWARN tamamlama koduyla başarılı olabilir. Çizelge 703 sayfa 1074 , ortaya çıkabilecek çeşitli durumları gösterir. Bu durumlarda, tamamlama kodu CCOK değilse, neden kodu aşağıdakilerden biridir:
 - RC2241
 - RC2242
 - RC2245

Not: Kuyruk yöneticisi, bir kuyruğa göz atarken ya da giriş için açılmış olan bir kuyruğu kapatırken grup ve bölüm bilgilerini denetlemez; bu durumlarda, tamamlama kodu her zaman CCOK olur (başka hata olmadığı varsayılarak).

Geçerli arama:	Önceki çağrı GMLOGO ile MQGET idi	Önceki çağrı GMLOGO olmadan MQGET idi
GMLOGO ile MQGET	CCFAIL	CCFAIL
GMLOGO olmadan MQGET	CCWARN	CCOK
Sonlandırılmamış grup ya da mantıksal ileti ile MQCLOSE	CCWARN	CCOK

İletileri ve bölümleri mantıksal sırayla almak isteyen uygulamalar, en basit seçenek olduğundan, GMLOGO ' ları belirtmek için önerilir. Kuyruk yöneticisi bu bilgileri yönettiği için, bu seçenek grup ve bölüm bilgilerini yönetme gereksinimini ortadan kaldırır. Ancak, özelleştirilmiş uygulamaların GMLOGO seçeneği tarafından sağlanandan daha fazla denetime gereksinimi olabilir ve bu, bu seçenek belirlenmediğinde gerçekleştirilebilir. Bu yapıldıysa, uygulama MQGET çağrılarında önce MQMD 'deki MDMID, MDCID, MDGID, MDSEQve MDOFF alanlarının ve MQGMO' daki GMMO MO* seçeneklerinin doğru olarak ayarlandığından emin olmalıdır.

Örneğin, aldığı fiziksel iletileri, bu iletilerin gruplar halinde mi, yoksa mantıksal ileti bölümlerinde mi olduğu dikkate alınmaksızın iletmek isteyen bir uygulamanın GMLOGO belirtmemesi gerekir. Bunun nedeni, gönderme ve alma kuyruğu yöneticileri arasında birden çok yolu olan karmaşık bir ağda fiziksel iletilerin sıradışı olarak gelebilmesidir. MQPUT çağrısında GMLOGO ve karşılık gelen PMLOGO belirtilmediğinde, iletmeye uygulaması her bir fiziksel iletiyi gelir gelmez, mantıksal sırayla bir sonrakini beklemek zorunda kalmadan alabilir ve iletir.

GMLOGO, diğer GM* seçeneklerinden herhangi biriyle ve uygun koşullarda çeşitli MO* seçenekleriyle belirtilebilir.

GMCMPPM

Yalnızca tam mantıksal iletiler alınabilir.

Bu seçenek, MQGET çağrısıyla yalnızca tam bir mantıksal ileti döndürüleceğini belirtir. Mantıksal ileti bölümlenmişse, kuyruk yöneticisi kesimleri yeniden birleştirir ve uygulamaya tüm mantıksal

iletiyi döndürür; mantıksal iletinin bölümlenmiş olması, bunu alan uygulama tarafından anlaşılır değildir.

Not: Bu, kuyruk yöneticisinin ileti bölümlerini yeniden birleştirmesine neden olan tek seçenektir. Belirtilmezse, bölümler kuyrukta varsa (ve MQGET çağrısında belirtilen diğer seçim ölçütlerine uyuyorsa) uygulamaya tek tek döndürülür. Tek tek kesimleri almak istemeyen uygulamalar her zaman GMCMPM ' yi belirtmelidir.

Bu seçeneği kullanmak için, uygulamanın iletinin tamamını kapsayacak büyüklükte bir arabellek sağlaması ya da GMATM seçeneğini belirtmesi gerekir.

Kuyruk, bazı bölümlerin eksik olduğu bölümlenmiş iletiler içeriyorsa (büyük olasılıkla ağda gecikmiş oldukları ve henüz gelmedikleri için), GMCMPM ' nin belirtilmesi, eksik mantıksal iletilere ait bölümlerin alınmasını engeller. Ancak, bu ileti bölümleri **CurrentQDepth** kuyruk özniteliğinin değerine katkıda bulunmaya devam eder; bu, *CurrentQDepth* sıfırdan büyük olsa da, alınabilir mantıksal ileti olmayabileceği anlamına gelir.

Kalıcı iletiler için, kuyruk yöneticisi bölümleri yalnızca bir iş birimi içinde yeniden birleştirebilir:

- MQGET çağrısı kullanıcı tanımlı bir iş birimi içinde çalışıyorsa, o iş birimi kullanılır. Çağrı, yeniden çevirme işlemi sırasında kısmen başarısız olursa, kuyruk yöneticisi, yeniden çevirme sırasında kaldırılan bölümleri kuyrukta yeniden gösterir. Ancak, başarısızlık iş biriminin başarıyla kesinleştirilmesini engellemez.
- Çağrı kullanıcı tanımlı bir iş birimi dışında çalışıyorsa ve kullanıcı tanımlı bir iş birimi yoksa, kuyruk yöneticisi çağrı süresi boyunca bir iş birimi yaratır. Çağrı başarılı olursa, kuyruk yöneticisi iş birimini otomatik olarak kesinleştirir (uygulamanın bunu yapması gerekmez). Çağrı başarısız olursa, kuyruk yöneticisi iş birimini geri alır.
- Çağrı kullanıcı tanımlı bir iş birimi dışında çalışıyorsa, ancak kullanıcı tanımlı bir iş birimi varsa, kuyruk yöneticisi yeniden çevirme gerçekleştiremez. İleti yeniden çevirme gerektirmiyorsa, arama yine de başarılı olabilir. Ancak, ileti yeniden çevirme gerektiriyorsa, arama RC2255 neden koduyla başarısız olur.

Kalıcı olmayan iletiler için kuyruk yöneticisi, yeniden çevirme gerçekleştirmek için bir iş biriminin kullanılabilir olmasını gerektirmez.

Bölüm olan her fiziksel iletinin kendi ileti tanımlayıcısı vardır. Tek bir mantıksal iletiyi oluşturan kesimler için, ileti tanımlayıcıdaki alanların çoğu mantıksal iletideki tüm kesimler için aynıdır- genellikle mantıksal iletideki kesimler arasında farklılık gösteren yalnızca *MDMID*, *MDOFF* ve *MDMFL* alanlarıdır. Ancak, bir kesim bir ara kuyruk yöneticisindeki bir teslim edilmeyen iletiler kuyruğuna yerleştirilirse, DLQ işleyicisi GMCONV seçeneğini belirten iletiyi alır ve bu, bölümün karakter kümesinin ya da kodlamasının değiştirilmesiyle sonuçlanabilir. DLQ işleyicisi segmenti başarılı bir şekilde gönderirse, kesim hedef kuyruk yöneticisine ulaştığında mantıksal iletideki diğer bölümlerden farklı bir karakter kümesi ya da kodlama olabilir.

MDCSI, *MDENC* ya da her iki alanın farklı olduğu bölümlerden oluşan bir mantıksal ileti, kuyruk yöneticisi tarafından tek bir mantıksal iletide yeniden birleştirilemez. Kuyruk yöneticisi, aynı karakter kümesi tanımlayıcılarına ve kodlamalarına sahip mantıksal iletinin başlangıcında ilk birkaç ardışık bölümü yeniden birleştirir ve döndürür; MQGET çağrısı, tamamlanma kodu CCWARN ve neden kodu RC2243 ya da RC2244 ile tamamlanır. Bu, GMCONV belirtilip belirtilmediğine bakılmaksızın gerçekleşir. Geri kalan bölümleri almak için, uygulamanın MQGET çağrısı GMCMPM seçeneği olmadan yeniden yayınlayarak bölümleri birer birer alması gerekir. GMLOGO, kalan bölümleri sırayla almak için kullanılabilir.

Ayrıca, bölümlerin ileti tanımlayıcısındaki diğer alanları bölümler arasında farklılık gösteren değerlere ayarlamasını sağlayan bir uygulama da olabilir. Ancak, alan uygulama mantıksal iletiyi almak için GMCMPM kullanıyorsa, bunu yapmanın bir yararı yoktur. Kuyruk yöneticisi bir mantıksal iletiyi yeniden birleştirdiğinde, ileti tanımlayıcısında ilk kesime ilişkin ileti tanımlayıcısının değerlerini döndürür; tek kural dışı durum, kuyruk yöneticisinin yeniden birleştirilen iletinin tek bölüm olduğunu belirtmek için ayarladığı *MDMFL* alanıdır.

Bir rapor iletisi için GMCMPM belirtilirse, kuyruk yöneticisi özel işleme gerçekleştirir. Kuyruk yöneticisi, mantıksal iletideki farklı bölümlerle ilgili rapor tipindeki tüm rapor iletilerinin kuyrukta

olup olmadığını kontrol eder. Varsa, GMCMPM belirtilerek tek bir ileti olarak alınabilir. Bunun mümkün olması için, rapor iletilerinin bölümlenmeyi destekleyen bir kuyruk yöneticisi ya da MCA tarafından oluşturulması ya da kaynak uygulamanın en az 100 baytlık ileti verisi (yani, uygun RO* D ya da RO* F seçenekleri belirtilmelidir) istemesi gerekir. Bir kesim için uygulama verilerinin tam miktarından az bir miktar varsa, eksik baytlar, döndürülen rapor iletilerinde boş değerlerle değiştirilir.

GMMUC ya da GMBRWC ile GMCMPM belirtilirse, göz atma imlecinin MQMD ' de 0 değerine sahip bir *MDOFF* alanı olan bir iletiye konumlandırılması gerekir. Bu koşul karşılanmazsa, arama RC2246 neden koduyla başarısız olur.

GMCMPM, belirtilmesi gerekmeyen GMASGA ' yi belirtir.

GMCMPM, GMPSYP dışındaki diğer GM* seçenekleriyle ve MOOFFS dışındaki MO* seçenekleriyle belirtilebilir.

GMAMSA

Gruptaki tüm iletiler kullanılabilir olmalıdır.

Bu seçenek, bir gruptaki iletilerin yalnızca gruptaki tüm iletiler kullanılabilir olduğunda alınabileceğini belirtir. Kuyrukta bazı iletilerin eksik olduğu ileti grupları varsa (belki de ağda gecikmiş oldukları ve henüz gelmedikleri için), GMAMSA belirtilmesi tamamlanmamış gruplara ait iletilerin alınmasını önler. Ancak, bu iletiler **CurrentQDepth** kuyruk özniteliğinin değerine katkıda bulunmaya devam eder; bu, **CurrentQDepth** sıfırdan büyük olsa da, alınabilir ileti grubu olmayabileceği anlamına gelir. Alınabilecek başka ileti yoksa, belirtilen bekleme aralığı (varsa) sona erdikten sonra RC2033 neden kodu döndürülür.

GMAMSA 'nın işlenmesi, GMLOGO' un da belirtilip belirtilmediğine bağlıdır:

- Her iki seçenek de belirtilirse, GMAMSA yalnızca geçerli bir grup ya da mantıksal ileti olmadığına bunu etkiler. Geçerli bir grup ya da mantıksal ileti varsa, GMAMSA yoksayılır. Bu, GMAMSA ' nın iletileri mantıksal sırayla işlerken açık kalabileceği anlamına gelir.
- GMAMSA GMLOGO olmadan belirtilirse, GMAMSA her zaman bir etkiye sahiptir. Bu, gruptaki kalan iletileri kaldırabilmek için, gruptaki ilk ileti kuyruktan kaldırıldıktan sonra seçeneğin kapatılması gerektiği anlamına gelir.

GMAMSA ile bir MQGET çağrısının başarıyla tamamlanması, MQGET çağrısının verildiği sırada gruptaki tüm iletilerin kuyrukta olduğu anlamına gelir. Ancak, diğer uygulamaların yine de gruptan ileti kaldırabildiklerini unutmayın (grup, gruptaki ilk iletiyi alan uygulamaya kilitlenmez).

Bu seçenek belirtilmezse, grup eksik olduğunda bile gruplara ait iletiler alınabilir.

GMAMSA, belirtilmesi gerekmeyen GMASGA ' yı belirtir.

GMAMSA, diğer GM* seçeneklerinden herhangi biriyle ve MO* seçeneklerinden herhangi biriyle belirtilebilir.

GMASGA

Mantıksal iletideki tüm bölümler kullanılabilir olmalıdır.

Bu seçenek, mantıksal iletideki bölümlerin yalnızca mantıksal iletideki tüm bölümler kullanılabilir olduğunda alınmak üzere kullanılabileceğini belirtir. Kuyruk, bazı bölümlerin eksik olduğu bölümlenmiş iletiler içeriyorsa (muhtemelen ağda geciktikleri ve henüz gelmedikleri için), GMASGA ' nın belirtilmesi, eksik mantıksal iletilere ait bölümlerin alınmasını önler. Ancak bu kesimler **CurrentQDepth** kuyruk özniteliğinin değerine katkıda bulunmaya devam eder; bu, **CurrentQDepth** sıfırdan büyük olsa da, alınabilir mantıksal ileti olmayabileceği anlamına gelir. Alınabilecek başka ileti yoksa, belirtilen bekleme aralığı (varsa) sona erdikten sonra RC2033 neden kodu döndürülür.

GMASGA 'nın işlenmesi, GMLOGO' un da belirtilip belirtilmediğine bağlıdır:

- Her iki seçenek de belirtilirse, GMASGA yalnızca geçerli bir mantıksal ileti olmadığına etkili olur. Geçerli bir mantıksal ileti varsa, GMASGA yoksayılır. Bu, GMASGA ' nın iletileri mantıksal sırayla işlerken açık kalabileceği anlamına gelir.

- GMLOGO olmadan GMASGA belirtilirse, GMASGA her zaman bir etkiye sahiptir. Bu, mantıksal iletideki kalan bölümleri kaldırabilmek için mantıksal iletideki ilk bölüm kaldırıldıktan sonra seçeneğin kapatılması gerektiği anlamına gelir.

Bu seçenek belirtilmezse, mantıksal ileti eksik olduğunda bile ileti bölümleri alınabilir.

Hem GMCMPM hem de GMASGA, herhangi birinin alınabilmesi için tüm bölümlerin kullanılabilir olmasını gerektiriyorken, önceki tüm iletiyi döndürür, ikincisi ise bölümlerin birer birer alınmasına izin verir.

Bir rapor iletisi için GMASGA belirtilirse, kuyruk yöneticisi özel işlem gerçekleştirir. Kuyruk yöneticisi, tam mantıksal iletiyi oluşturan her bölüm için en az bir rapor iletisi olup olmadığını görmek için kuyruğu denetler. Varsa, GMASGA koşulu karşılanır. Ancak, kuyruk yöneticisi mevcut rapor iletilerinin tipini denetlemez ve bu nedenle mantıksal iletinin bölümleriyle ilgili rapor iletilerinde rapor tiplerinin bir karışımı olabilir. Sonuç olarak, GMASGA 'nın başarısı GMCMPM' nin başarılı olduğu anlamına gelmez. Belirli bir mantıksal iletinin bölümleri için var olan rapor tiplerinin bir karışımı varsa, bu rapor iletileri birer birer alınmalıdır.

GMASGA, diğer GM* seçeneklerinden herhangi biriyle ve MO* seçeneklerinden herhangi biriyle belirtilebilir.

Varsayılan seçenek: Açıklanan seçeneklerin hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

GMNONE

Seçenek belirtilmedi.

Bu değer, başka seçenek belirtilmediğini belirtmek için kullanılabilir; tüm seçenekler varsayılan değerlerini varsayar. GMNONE, program belgelerine yardımcı olmak için tanımlanır; bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

GMOPT alanının ilk değeri GMNWT 'dir.

GMRE1 (1 baytlık karakter dizgisi)

Ayrılmış.

Bu ayrılmış bir alandır. Bu alanın ilk değeri boş bir karakterdir. *GMVER* , *GMVER2* değerinden küçükse bu alan yoksayılar.

GMRL (10 basamaklı işaretli tamsayı)

Döndürülen ileti verilerinin uzunluğu (bayt).

Bu, kuyruk yöneticisi tarafından **BUFFER** değiştirilmesinde MQGET çağrısının döndürdüğü ileti verilerinin bayt cinsinden uzunluğuna ayarlanan bir çıkış alanıdır. Kuyruk yöneticisi bu yeteneği desteklemiyorsa, *GMRL* *RLUNDF* değerine ayarlanır.

İletiler kodlamalar ya da karakter kümeleri arasında dönüştürüldüğünde, ileti verileri bazen boyutu değiştirebilir. MQGET çağrısından dönüş:

- *GMRL* *RLUNDF* değilse, döndürülen ileti verilerinin bayt sayısı *GMRL* tarafından verilir.
- *GMRL* *RLUNDF* değerine sahipse, döndürülen ileti verilerinin bayt sayısı genellikle *BUFLEN* ve *DATLEN* küçükleri tarafından verilir, ancak MQGET çağrısı RC2079 neden koduyla tamamlanırsa bundan daha az olabilir. Bu durumda, **BUFFER** değiştirilmesindeki önemsiz byte 'lar boş değere ayarlanır.

Aşağıdaki özel değer tanımlandı:

RLUNDF

Döndürülen verilerin uzunluğu tanımlanmadı.

Bu alanın ilk değeri *RLUNDF* 'dir. *GMVER* *GMVER3* değerinden küçükse bu alan yoksayılar.

GMRQN (48 baytlık karakter dizgisi)

Hedef kuyruğun çözülmüş adı.

Bu, kuyruk yöneticisi tarafından yerel kuyruk yöneticisine tanımlandığı şekilde, iletinin alındığı kuyruğun yerel adına ayarlanan bir çıkış alanıdır. Bu, aşağıdaki durumda kuyruğu açmak için kullanılan addan farklıdır:

- Bir diğer ad kuyruğu açıldı (bu durumda, çözümlenen diğer adın döndürüldüğü yerel kuyruğun adı) ya da
- Bir model kuyruğu açıldı (bu durumda, dinamik yerel kuyruğun adı döndürülür).

Bu alanın uzunluğu LNQN tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

GMRS2 (1 baytlık karakter dizgisi)

Ayrılmış.

Bu ayrılmış bir alandır. Bu alanın ilk değeri boş bir karakterdir. *GMVER* , *GMVER4*değerinden küçükse bu alan yoksayılr.

GMSEG (1 baytlık karakter dizgisi)

Alınan ileti için daha fazla bölümlenmeye izin verilip verilmediğini belirten işaret.

Aşağıdaki değerlerden birini içerir:

SEGIHB.

Bölümlenmeye izin verilmiyor.

DENİZCİK

Bölümlenmeye izin verilir.

Bu bir çıkış alanıdır. Bu alanın ilk değeri SEGIHB 'dir. *GMVER* , *GMVER2*değerinden küçükse bu alan yoksayılr.

GMSG1 (10 basamaklı işaretli tamsayı)

Sinyal.

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri 0 'dır.

GMSG2 (10 basamaklı işaretli tamsayı)

Sinyal tanıtıcısı.

Bu ayrılmış bir alandır; değeri önemli değildir.

GMSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

GMSIDV

İleti alma seçenekleri yapısına ilişkin tanıtıcı.

Bu alan her zaman bir giriş alanıdır. Bu alanın ilk değeri GMSIDV 'dir.

GMSST (1 baytlık karakter dizgisi)

Alınan iletinin bir mantıksal iletinin parçası olup olmadığını gösteren işaret.

Aşağıdaki değerlerden birini içerir:

SSNSEG

İleti bir bölüm değil.

SSSSEG

İleti bir bölüm, ancak mantıksal iletinin son bölümü değil.

SSLSEG

İleti, mantıksal iletinin son bölümüdür.

Bu değer, mantıksal ileti tek bir bölümden oluşuyorsa da döndürülür.

Bu alan bir çıkış alanıdır. Bu alanın ilk değeri SSNSEG 'dir. *GMVER* , *GMVER2*değerinden küçükse bu alan yoksayılr.

GMTOK (16 baytlık bit dizilim)

İleti simgesi.

Bu ayrılmış bir alandır; değeri önemli değildir. Aşağıdaki özel değer tanımlandı:

MTKN

İleti simgesi yok.

Değer, alanın uzunluğu için ikili sıfırdır.

Bu alanın uzunluğu LNMTOK tarafından verilir. Bu alanın ilk değeri MTKNON değeridir. *GMVER* *GMVER3*değerinden küçükse bu alan yoksayılr.

GMVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer aşağıdakilerden biri olmalıdır:

GMVER1

Version-1 get-message seçenekleri yapısı.

GMVER2

Version-2 get-message seçenekleri yapısı.

GMVER3

Version-3 get-message seçenekleri yapısı.

GMVER4

Version-4 ileti alma seçenekleri yapısı.

Yalnızca yapının daha yeni sürümlerinde bulunan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

GMVERC

get-message seçenekleri yapısının yürürlükteki sürümü.

Bu alan her zaman bir giriş alanıdır. Bu alanın ilk değeri *GMVER1*' dir.

GMVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer aşağıdakilerden biri olmalıdır:

GMVER1

Version-1 get-message seçenekleri yapısı.

GMVER2

Version-2 get-message seçenekleri yapısı.

GMVER3

Version-3 get-message seçenekleri yapısı.

GMVER4

Version-4 ileti alma seçenekleri yapısı.

Yalnızca yapının daha yeni sürümlerinde bulunan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

GMVERC

get-message seçenekleri yapısının yürürlükteki sürümü.

Bu alan her zaman bir giriş alanıdır. Bu alanın ilk değeri *GMVER1*' dir.

GMWI (10 basamaklı işaretli tamsayı)

Bekleme aralığı.

Bu, MQGET çağrısının uygun bir iletinin gelmesi için milisaniye cinsinden ifade edilen yaklaşık süredir (yani, MQGET çağrısının **MSGDSC** değiştirgesinde belirtilen seçim ölçütlerini karşılayan bir ileti; daha fazla ayrıntı için “IBM i üzerinde MQMD (İleti tanımlayıcı)” sayfa 1093 içinde açıklanan *MDMID* alanına bakın). Bu süre geçtikten sonra uygun bir ileti gelmezse, çağrı CCFAIL ve neden kodu RC2033 ile tamamlanır.

GMWI , *GMWT* seçeneğiyle birlikte kullanılır. Bu seçenek belirtilmezse yoksayılr. Belirtilirse, *GMWI* sıfırdan büyük ya da sıfıra eşit ya da aşağıdaki özel değer olmalıdır:

WIULIM

Sınırsız bekleme aralığı.

Bu alanın ilk değeri 0 'dır.

Başlangıçtaki değerler

Çizelge 704. MQGMO 'daki Alanlar		
Alan adı	Değişmez adı	Değişmez değeri
<i>GMSID</i>	GMSIDV	'GMO↔'
<i>GMVER</i>	GMVER1	1
<i>GMOPT</i>	GMNWT	0
<i>GMWI</i>	Yok	0
<i>GMSG1</i>	Yok	0
<i>GMSG2</i>	Yok	0
<i>GMRQN</i>	Yok	Boşluklar
<i>GMMO</i>	MOMSGI + MOCORI	3
<i>GMGST</i>	GSNIG	' '
<i>GMSST</i>	SSNSEG	' '
<i>GMSEG</i>	SEGIHB.	' '
<i>GMRE1</i>	Yok	' '
<i>GMTOK</i>	MTKN	Boş Değerler
<i>GMRL</i>	RLUNDF	-1
<i>GMRS2</i>	Yok	' '
<i>GMMH</i>	HMNONE.	0

Notlar:

1. ↔ simgesi, tek bir boş karakteri temsil eder.

RPG bildirim

```

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 criteria used for MQGET		
D	GMMO	73	76I 0 INZ(3)
D*	Flag indicating whether message retrieved is in a group		
D	GMGST	77	77 INZ(' ')
D*	Flag indicating whether message retrieved is a segment of a logical message		
D	GMSST	78	78 INZ(' ')
D*	Flag indicating whether further segmentation is allowed for the message 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

IBM i üzerinde MQIIH (IMS bilgi üstbilgisi)

MQIIH yapısı, IBM MQ for z/OS aracılığıyla IMS köprüsüne gönderilen bir iletinin başında bulunması gereken bilgileri açıklar.

Genel Bakış

Biçim adı: FMIMS.

Karakter kümesi ve kodlama: MQIIH yapısı ve uygulama iletisi verileri için kullanılan karakter kümesi ve kodlama için özel koşullar geçerlidir:

- IMS köprü kuyruğuna sahip kuyruk yöneticisine bağlanan uygulamalar, kuyruk yöneticisinin karakter kümesinde ve kodlamasında bulunan bir MQIIH yapısı sağlamalıdır. Bunun nedeni, bu durumda MQIIH yapısına ilişkin veri dönüştürmenin gerçekleştirilmemiş olmasıdır.
- Diğer kuyruk yöneticilerine bağlanan uygulamalar, desteklenen karakter kümelerinde ve kodlamasında bulunan bir MQIIH yapısı sağlayabilir; MQIIH ' nin dönüştürülmesi, IMS köprü kuyruğunun sahibi olan kuyruk yöneticisine bağlı olan alıcı ileti kanalı aracısı tarafından gerçekleştirilir.

Not: Bunun bir istisnası var. IMS köprü kuyruğuna sahip kuyruk yöneticisi dağıtımlı kuyruğa alma için CICS kullanıyorsa, MQIIH, IMS köprü kuyruğuna sahip kuyruk yöneticisinin karakter kümesinde ve kodlamasında olmalıdır.

- MQIIH yapısını izleyen uygulama iletisi verileri, MQIIH yapısıyla aynı karakter kümesinde ve kodlamada olmalıdır. MQIIH yapısındaki *IICSI* ve *IIENC* alanları, uygulama ileti verilerinin karakter kümesini ve kodlamasını belirtmek için kullanılamaz.

Veriler, kuyruk yöneticisi tarafından desteklenen yerleşik biçimlerden biri değilse, uygulama iletisi verilerini dönüştürmek için kullanıcı tarafından bir veri dönüştürme çıkışı sağlanmalıdır.

- [“IMS köprü uygulamaları için passticsepetlerinin doğrulanması” sayfa 1082](#)
- [“Alanlar” sayfa 1082](#)
- [“Başlangıçtaki değerler” sayfa 1085](#)
- [“RPG bildirim” sayfa 1085](#)

IMS köprü uygulamaları için passticsepetlerinin doğrulanması

Artık IBM MQ yöneticilerinin IMS köprü uygulamaları için passticsepetlerin kimlik doğrulaması için kullanılacak uygulama adını belirtmeleri mümkündür. Bunu yapmak için, uygulama adı STGCLASS nesne tanımlaması için yeni bir PTKTAPPL özniteliği olarak 1-8 karakterlik alfasayısal dizgi olarak belirlenir.

Boş değer, kimlik doğrulamanın önceki IBM MQ yayınlarda olduğu gibi gerçekleştiği anlamına gelir; yani, kimlik doğrulama isteğinde uygulama adı akışı yoktur ve bunun yerine MVSxxxx değeri kullanılır.

1-8 alfasayısal karakterden oluşan bir değer, RACF yayınlarında açıklandığı gibi geçiş bileti uygulaması adlarına ilişkin kurallara uygun olmalıdır.

IBM MQ Yöneticiler ve RACF yöneticileri, kullanılacak geçerli uygulama adlarını kabul etmelidir. RACF denetimcisi, PTKTDATA sınıfında, erişim verilecek tüm uygulamaların kullanıcı kimliklerine okuma erişimi veren bir tanımlama yaratmalıdır. IBM MQ yöneticisi, geçiş bileti kimlik doğrulaması için kullanılacak uygulama adını belirten gerekli STGCLASS tanımlarını oluşturmalı ya da değiştirmelidir.

İlgili bilgiler için *Script (MQSC) Command Reference* belgesine bakın.

Alanlar

MQIIH yapısı aşağıdaki alanları içerir; alanlar **alfabetik sıradada** açıklanır:

IIAUT (8 baytlık karakter dizgisi)

RACF parolası ya da geçiş bileti.

Bu isteğe bağlıdır; belirtildiyse, güvenlik bağlamı sağlamak üzere IMS ' e gönderilen bir UTOKEN oluşturmak için MQMD güvenlik bağlamındaki kullanıcı kimliğiyle birlikte kullanılır. Belirtilmezse, kullanıcı kimliği doğrulanmadan kullanılır. Bu, bir kimlik doğrulayıcının var olmasını gerektirebilecek RACF anahtarlarının ayarına bağlıdır.

İlk bayt boş değerli ya da boş ise bu değer dikkate alınmaz. Aşağıdaki özel değer kullanılabilir:

IAUNON

Kimlik doğrulaması yok.

Bu alanın uzunluğu LNAUTH tarafından verilir. Bu alanın ilk değeri IAUNON.

IICMT (1 byte karakter dizgisi)

Kesinleştirme kipi.

IMS kesinleştirme kiplerine ilişkin ek bilgi için *OTMA Reference* adlı yayına bakın. Değer aşağıdakilerden biri olmalıdır:

ICMCTS

Kesinleştir ve gönder.

Bu mod, çıktının çift kuyruğa alındığını, ancak bölgenin kullanım sürelerinin kısıldığını gösterir. Hızlı yol ve etkileşimli hareketler bu kiple çalışamaz.

ICMSTC

Gönder ve kesinleştir.

Bu alanın ilk değeri ICMCTS 'dir.

IICSI (10 basamaklı işaretli tamsayı)

Ayrılmış.

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri 0 'dir.

IIENC (10 basamaklı işaretli tamsayı)

Ayrılmış.

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri 0 'dir.

IIFLG (10 basamaklı işaretli tamsayı)

Bayraklar.

Değer şöyle olmalıdır:

IINONE

Bayrak yok.

Bu alanın ilk değeri IINONE.

IIFMT (8 baytlık karakter dizgisi)

IBM MQ MQIIH ' yi izleyen verilerin biçim adı.

MQIIH yapısını izleyen verilerin IBM MQ biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *MDFMT* alanıyla aynıdır.

Bu alanın uzunluğu LNFMT tarafından verilir. Bu alanın ilk değeri FMNONE.

IILEN (10 basamaklı işaretli tamsayı)

MQIIH yapısının uzunluğu.

Değer şöyle olmalıdır:

IILEN1

IMS bilgi üstbilgisi yapısının uzunluğu.

Bu alanın ilk değeri IILEN1' dir.

IILTO (8 baytlık karakter dizgisi)

Mantıksal uçbirim geçersiz kılma.

Bu, GÇ PCB alanına yerleştirilir. İsteğe bağlıdır; belirtilmezse, TPIPE adı kullanılır. İlk bayt boşsa ya da boş değerliyse dikkate alınmaz.

Bu alanın uzunluğu LNLTOV tarafından verilir. Bu alanın ilk değeri 8 boş karakterdir.

IIMMN (8 baytlık karakter dizgisi)

İleti biçimi hizmetleri eşlemi adı.

Bu, GÇ PCB alanına yerleştirilir. Bu isteğe bağlıdır. Girişte MID 'yi temsil eder, çıkışta MOD' yi temsil eder. İlk bayt boş değerli ya da boş ise dikkate alınmaz.

Bu alanın uzunluğu LNMFMN tarafından verilir. Bu alanın ilk değeri 8 boş karakterdir.

IIRFM (8 baytlık karakter dizgisi)

IBM MQ yanıt iletilisinin biçim adı.

Bu, yürürlükteki iletiye yanıt olarak gönderilecek yanıt iletilisinin IBM MQ biçim adıdır. Bu kodu kodlamaya ilişkin kurallar, MQMD ' deki *MDFMT* alanıyla aynıdır.

Bu alanın uzunluğu LNFMT tarafından verilir. Bu alanın ilk değeri FMNONE.

IIRSV (1 baytlık karakter dizgisi)

Ayrılmış.

Bu ayrılmış bir alandır; boş olmalıdır.

IISEC (1 baytlık karakter dizgisi)

Güvenlik kapsamı.

Bu, gerekli IMS güvenlik işlemlerini gösterir. Aşağıdaki değerler tanımlanır:

ISSCHK

Güvenlik kapsamını denetleyin.

Denetim bölgesinde bir ACEE oluşturulur, ancak bağımlı bölgede oluşturulmaz.

ISSFUL

Tam güvenlik kapsamı.

Denetim bölgesinde ön belleğe alınmış bir ACEE oluşturulur ve bağımlı bölgede ön belleğe alınmamış bir ACEE oluşturulur. ISSFUL kullanıyorsanız, ACEE 'nin oluşturulduğu kullanıcı kimliğinin bağımlı bölgede kullanılan kaynaklara erişimi olduğundan emin olmanız gerekir.

Bu alan için ISSCHK ve ISSFUL belirtilmezse, ISSCHK varsayılır.

Bu alanın ilk değeri ISSCHK 'dir.

IISID (4 baytlık karakter dizisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

IISIDV

IMS bilgi üstbilgisi yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri IISIDV 'dir.

IITID (16 baytlık bit dizisi)

İşlem eşgörünümü tanıtıcısı.

Bu alan, IMS 'in çıkış iletileri tarafından kullanıldığından ilk girişte yoksayıdır. *IITST* , *ITSIC* olarak ayarlanırsa, IMS 'in iletileri doğru etkileşimde ilintilendirmesini sağlamak için sonraki girişte ve sonraki tüm girişlerde bu belirtilmelidir. Aşağıdaki özel değer kullanılabilir:

İTON

Hareket eşgörünümü tanıtıcısı yok.

Bu alanın uzunluğu LNTIID tarafından verilir. Bu alanın ilk değeri ITINON.

IITST (1 baytlık karakter dizisi)

İşlem durumu.

Bu, IMS etkileşim durumunu gösterir. Etkileşim olmadığı için ilk girişte bu yoksayıdır. Sonraki girişlerde, bir etkileşimin etkin olup olmadığını gösterir. Çıkışta IMStarafından ayarlanır. Değer aşağıdakilerden biri olmalıdır:

İTSIK

Sohbet sırasında.

İTSNIK

Sohbette değil.

ITSARC

Hareket durumu verilerini mimari biçimde döndürür.

Bu değer yalnızca IMS /DISPLAY TRAN komutuyla kullanılır. Bu, hareket durumu verilerinin karakter formu yerine IMS mimari biçiminde döndürülmesine neden olur. Daha fazla ayrıntı için bkz. [IMS hareket programlarını IBM MQ aracılığıyla yazma](#).

Bu alanın ilk değeri ITSNIC 'dir.

IIVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

IIVER1

IMS bilgi üstbilgisi yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

IIVERC

IMS bilgi üstbilgisi yapısının geçerli sürümü.

Bu alanın ilk değeri IIVER1' dir.

Başlangıçtaki değerler

Çizelge 705. MQIIH ' deki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
IISID	IISIDV	' IIH↵ '
IIVER	IIVER1	1
IILEN	IILEN1	84
IIENC	Yok	0
IICSI	Yok	0
IIFMT	FMNONE	Boşluklar
IIFLG	IINONE	0
IILTO	Yok	Boşluklar
IIMMN	Yok	Boşluklar
IIRFM	FMNONE	Boşluklar
IIAUT	IAUNON	Boşluklar
IITID	İTON	Boş Değerler
IITST	İTSNIK	' '
IICMT	ICMCTS	' 0 '
IISEC	ISSCHK	' C '
IIRSV	Yok	' '

Notlar:

1. ↵ simgesi, tek bir boş karakteri temsil eder.

RPG bildirimi

```
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'0000000000000000-0000000000000000')
D				
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 IBM i üzerinde MQIMPO (İleti özelliği seçeneklerini sorgularken sor)

MQIMPO yapısı, uygulamaların iletilerin özelliklerinin nasıl sorulduğunu denetleyen seçenekleri belirtmesine olanak sağlar.

Genel Bakış

Amaç: Yapı, MQINQMP çağrısındaki bir giriş değiştirmesidir.

Karakter kümesi ve kodlama: MQIMPO içindeki veriler, uygulamanın (ENNAT) karakter kümesinde ve kodlamasında olmalıdır.

- [“Alanlar” sayfa 1086](#)
- [“Başlangıçtaki değerler” sayfa 1092](#)
- [“RPG bildirimi” sayfa 1092](#)

Alanlar

MQIMPO yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

IPOPT (10 basamaklı işaretli tamsayı)

Aşağıdaki seçenekler MQINQMP işlemini denetler. Bu seçeneklerden birini ya da birkaçını belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyse). Geçerli olmayan seçenek birleşimleri belirtilir; diğer tüm birleşimler geçerlidir.

Değer verileri seçenekleri: Aşağıdaki seçenekler, özellik iletimden alındığında değer verilerinin işlenmesiyle ilgilidir.

IPCVAL

Bu seçenek, özelliğin değerinin, MQINQMP çağrısının *Value* alanındaki özellik değerini döndürmesinden önce belirtilen *IPREQCSI* ve *IPREQENC* değerlerine uyacak şekilde dönüştürülmesini ister.

- Dönüştürme başarılı olursa, MQINQMP çağrısından dönüşte *IPRETCSI* ve *IPRETENC* alanları *IPREQCSI* ve *IPREQENC* ile aynı değere ayarlanır.
- Dönüştürme başarısız olursa, ancak MQINQMP çağrısı hatasız tamamlanmazsa, özellik değeri dönüştürülmeden döndürülür.

Özellik bir dizgiyse, *IPRETCSI* ve *IPRETENC* alanları, dönüştürülmemiş dizginin karakter kümesine ve kodlamasına ayarlanır.

Bu durumda tamamlanma kodu CCWARN olur; neden kodu RC2466. Özellik imleci, döndürülen özelliğe ilerler.

Özellik değeri dönüştürme sırasında genişler ve **Value** parametresinin boyutunu aşarsa, CCFAIL tamamlanma koduyla değer dönüştürülmeden döndürülür; neden kodu RC2469olarak ayarlanır.

MQINQMP çağrısının **DataLength** değiştirgesi, uygulamanın dönüştürülen özellik değerini barındırmak için gereken arabellek büyüklüğünü saptamasını sağlamak için özellik değerinin dönüştürüleceği uzunluğu döndürür. Özellik imleci değiştirilmedi.

Bu seçenek ayrıca şunları da ister:

- Özellik adı bir joker karakter içeriyorsa, ve
- **IPRETNAMECHRP** alanı, döndürülen ad için bir adres ya da görelî konum ile ilk kullanıma hazırlanır.

Daha sonra, döndürülen ad **IPREQCSI** ve **IPREQENC** değerlerine uygun olacak şekilde dönüştürülür.

- Dönüştürme başarılı olursa, **IPRETNAMECHRP** alanının **VSCCSID** alanı ve döndürülen adın kodlaması, **IPREQCSI** ve **IPREQENC** giriş değerine ayarlanır.
- Dönüştürme başarısız olursa, ancak MQINQMP çağrısı hata ya da uyarı olmadan tamamlanmazsa, döndürülen ad dönüştürülmez. Bu durumda tamamlanma kodu CCWARN olur; neden kodu RC2492.

Özellik imleci, döndürülen özelliğe ilerler. Hem değer hem de ad dönüştürülmezse RC2466 döndürülür.

Döndürülen ad dönüştürme sırasında genişler ve **RequestedName** alanının **VSBufsize** alanının boyutunu aşarsa, döndürülen dizgi dönüştürülmemiş olarak bırakılır ve tamamlanma kodu CCFAIL olur ve neden kodu RC2465olarak ayarlanır.

MQCHARV yapısının **VSLength** alanı, uygulamanın dönüştürülen özellik değerini barındırmak için gereken arabellek büyüklüğünü saptamasına izin vermek için özellik değerinin dönüştürüleceği uzunluğu döndürür. Özellik imleci değiştirilmedi.

IPCTYP

Bu seçenek, özelliğin değerinin yürürlükteki veri tipinden MQINQMP çağrısının **Type** değiştirgesinde belirtilen veri tipine dönüştürülmesini ister.

- Dönüştürme başarılı olursa, MQINQMP çağrısının döndürülmesinde **Type** değiştirgesi değişmez.
- Dönüştürme başarısız olursa, ancak MQINQMP çağrısı hatasız olarak tamamlanırsa, çağrı RC2470nedeniyle başarısız olur. Özellik imleci değiştirilmedi.

Veri tipinin dönüştürülmesi, dönüştürme sırasında değerin genişletilmesine neden olursa ve dönüştürülen değer **Value** parametresinin boyutunu aşarsa, CCFAIL tamamlanma kodu ile değer dönüştürülmeden döndürülür ve neden kodu RC2469olarak ayarlanır.

MQINQMP çağrısının **DataLength** değiştirgesi, uygulamanın dönüştürülen özellik değerini barındırmak için gereken arabellek büyüklüğünü saptamasını sağlamak için özellik değerinin dönüştürüleceği uzunluğu döndürür. Özellik imleci değiştirilmedi.

MQINQMP çağrısının **Type** parametresinin değeri geçerli değilse, çağrı RC2473nedeniyle başarısız olur.

İstenen veri tipi dönüşümü desteklenmiyorsa, çağrı başarısız olur; neden: RC2470. Aşağıdaki veri tipi dönüşümleri desteklenir:

Çizelge 706. Desteklenen veri tipi dönüşümleri	
Özellik veri tipi	Desteklenen hedef veri tipleri
TYPBOL	TYPSTR, TYPI8, TYPI16, TYPI32, TYPI64
TYPBST	TYPSTR
TYPI8	TYPSTR, TYPI16, TYPI32, TYPI64
TYPI16	TYPSTR, TYPI32, TYPI64

Çizelge 706. Desteklenen veri tipi dönüştürmeleri (devamı var)

Özellik veri tipi	Desteklenen hedef veri tipleri
TYPI32	TYPSTR, TYPI64
TYPI64	TYPSTR
TYPF32	TYPSTR, TYPF64
TYPF64	TYPSTR
TYPSTR	TYPBOL, TYPI8, TYPI16, TYPI32, TYPI64, TYPF32, TYPF64
TYPNUL	Yok

Desteklenen dönüştürmeleri düzenleyen genel kurallar şunlardır:

- Dönüştürme sırasında veri kaybı olmaması koşuluyla, sayısal özellik değerleri bir veri tipinden diğerine dönüştürülebilir.
Örneğin, TYPI32 veri tipine sahip bir özelliğin değeri, TYPI64 veri tipine sahip bir değere dönüştürülebilir, ancak TYPI16 veri tipine sahip bir değere dönüştürülemez.
- Herhangi bir veri tipindeki bir özellik değeri dizgiye dönüştürülebilir.
- Dizginin dönüştürme için doğru biçimlenmesi koşuluyla, dizgi özelliği değeri başka bir veri tipine dönüştürülebilir. Bir uygulama doğru biçimlenmemiş bir dizgi özelliği değerini dönüştürmeyi denerse, IBM MQ RC2472 neden kodunu döndürür.
- Bir uygulama desteklenmeyen bir dönüştürme girişiminde bulunursa, IBM MQ RC2470 neden kodunu döndürür.

Bir özellik değerini bir veri tipinden diğerine dönüştürmeye ilişkin kurallar şunlardır:

- Bir TYPBOL özellik değeri bir dizgiye dönüştürülürken, TRUE değeri "TRUE" dizgisine ve false değeri "FALSE" dizgisine dönüştürülür.
- Bir TYPBOL özellik değeri sayısal veri tipine dönüştürülürken TRUE değeri bir, FALSE değeri sıfıra dönüştürülür.
- Bir dizgi özelliği değeri TYPBOL değerine dönüştürülürken, "TRUE" ya da "1" dizgisi TRUE değerine ve "FALSE" ya da "0" dizgisi FALSE değerine dönüştürülür.

"TRUE" ve "FALSE" terimlerinin büyük ve küçük harfe duyarlı olmadığını unutmayın.

Başka dizgiler dönüştürülemiyor; IBM MQ , RC2472 neden kodunu döndürür.

- Bir dizgi özelliği değerini TYPI8, TYPI16, TYPI32 ya da TYPI64 veri tipli bir değere dönüştürürken, dizginin biçimi şu olmalıdır:

```
[blanks][sign]digits
```

Dizginin bileşenlerinin anlamları aşağıdaki gibidir:

blanks

İsteğe bağlı baştaki boş karakterler

sign

İsteğe bağlı bir artı işareti (+) ya da eksi işareti (-) karakteri.

digits

Bitişik sayı karakterleri sırası (0-9). En az bir basamak karakteri bulunmalıdır.

Sayı karakterleri dizgisinden sonra, dizilim rakam karakterleri olmayan diğer karakterleri içerebilir, ancak bu karakterlerin ilkine ulaşılar ulaşılmaz dönüştürme durur. Dizginin bir ondalık tamsayıyı temsil ettiği varsayılır.

IBM MQ , dizgi doğru biçimlendirilmediyse, RC2472 neden kodunu döndürür.

- Bir dizgi özelliği değerini TYPF32 ya da TYPF64 veri tipindeki bir değere dönüştürürken, dizginin şu biçimi olmalıdır:

```
[blanks][sign]digits[.digits][e_char[e_sign]e_digits]
```

Dizginin bileşenlerinin anlamları aşağıdaki gibidir:

blanks

İsteğe bağlı baştaki boş karakterler

sign

İsteğe bağlı bir artı işareti (+) ya da eksi işareti (-) karakteri.

digits

Bitişik sayı karakterleri sırası (0-9). En az bir basamak karakteri bulunmalıdır.

e_char

"E" ya da "e" olan üstel bir karakter.

e_sign

Üstel için isteğe bağlı artı işareti (+) ya da eksi işareti (-) karakteri.

e_digits

Üs için bitişik sayı karakterleri sırası (0-9). Dizgi bir üstel karakter içeriyorsa, en az bir basamak karakteri bulunmalıdır.

Sayı karakterleri sırasından ya da bir üstel karakteri gösteren isteğe bağlı karakterlerden sonra, dizgi rakam karakteri olmayan diğer karakterleri içerebilir, ancak bu karakterlerin ilkine ulaşıldığında dönüştürme durur. Dizginin, 10 üslü bir ondalık kayan nokta sayısını temsil ettiği varsayılır.

IBM MQ , dizgi doğru biçimlendirilmediyse, RC2472 neden kodunu döndürür.

- Sayısal bir özellik değeri dizgiye dönüştürülürken, değer, o değere ilişkin ASCII karakterini içeren dizgiye değil, değer ondalık sayı olarak dizgi gösterimine dönüştürülür. Örneğin, 65 tamsayısı "A" dizgisine değil, "65" dizgisine dönüştürülür.
- Bayt dizilimi özellik değeri dizgiye dönüştürülürken, her bayt, baytı gösteren iki onaltılı karaktere dönüştürülür. Örneğin, {0xF1, 0x12, 0x00, 0xFF} bayt dizisi "F11200FF" dizgisine dönüştürülür.

IPQLEN

Özellik değerinin tipini ve uzunluğunu sorgulayın. Uzunluk, MQINQMP çağrısının **DataLength** değiştirgesinde döndürülür. Özellik değeri döndürülmedi.

Bir *ReturnedName* arabelleği belirtilirse, MQCHARV yapısının *VSLength* alanı özellik adının uzunluğuyla doldurulur. Özellik adı döndürülmez.

Yineleme seçenekleri: Aşağıdaki seçenekler, genel arama karakteri içeren bir ad kullanılarak özellikler üzerinde yineleme ile ilgilidir.

IPINQF

Belirtilen adla eşleşen ilk özelliği sorgulayın. Bu çağrıdan sonra, döndürülen özellik üzerinde bir imleç oluşturulur.

Bu varsayılan değerdir.

IPINQC seçeneği, gerekiyorsa, aynı özelliği yeniden sorgulamak için bir MQINQMP çağrısıyla birlikte kullanılabilir.

Tek bir özellik imleci olduğunu unutmayın; bu nedenle, MQINQMP çağrısında belirtilen özellik adı ilk durumuna getirilirse, geçici çizelge ilk durumuna getirilir.

Bu seçenek aşağıdaki seçeneklerden biriyle geçerli değildir:

IPINQN

IPINQC

IPINQN

Belirtilen adla eşleşen sonraki özelliğe ilişkin bilgi isteminde, özellik imlecinden aramaya devam edin. İmleç, döndürülen özelliğe ilerler.

Bu, belirtilen ada ilişkin ilk MQINQMP çağrısıya, belirtilen adla eşleşen ilk özellik döndürülür.

IPINQC seçeneği, gerekiyorsa, aynı özelliği yeniden sorgulamak için bir MQINQMP çağrısıyla birlikte kullanılabilir.

Geçici çizelgenin altındaki özellik silindiye, MQINQMP silinen özelliğin ardından eşleşen bir sonraki özelliği döndürür.

Bir yineleme devam ederken genel arama karakteriyle eşleşen bir özellik eklenirse, özellik yineleme tamamlanırken döndürülebilir ya da döndürülemeyebilir. Özellik, yineleme IPINQF kullanılarak yeniden başlatıldıktan sonra döndürülür.

Yineleme devam ederken silinen joker karakterle eşleşen bir özellik, silindikten sonra döndürülmez.

Bu seçenek aşağıdaki seçeneklerden biriyle geçerli değildir:

IPINQF
IPINQC

IPINQC

Özellik imlecinin gösterdiği özelliğin değerini alın. Özellik imlecinin gösterdiği özellik, IPINQF ya da IPINQN seçeneği kullanılarak en son sorulmuş özelliktir.

İleti tanıtıcısı yeniden kullanıldığında, ileti tanıtıcısı MQGET çağrısındaki MQGMO ' nun *MsgHandle* alanında belirtildiğinde ya da ileti tanıtıcısı bir MQPUT çağrısındaki MQPMO yapısının *OriginalMsgHandle* ya da *NewMsgHandle* alanlarında belirtildiğinde özellik geçici çizelgesi ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında ya da özellik imleci tarafından gösterilen özellik silindiğinde bu seçenek kullanılırsa, çağrı tamamlanma kodu CCFAIL ve neden RC2471 ile başarısız olur.

Bu seçenek aşağıdaki seçeneklerden biriyle geçerli değildir:

IPINQF
IPINQN

Daha önce açıklanan seçeneklerden hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

IPNONE.

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar.

IPNONE, program belgelerine yardımcı olur; bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmaz, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri IPINQF 'dir.

IPREQCSI (10 basamaklı işaretli tamsayı)

Değer bir karakter dizgisiyse, sorulmuş özellik değerinin dönüştürüleceğini karakter kümesi. Bu, IPCVAL ya da IPCTYP belirtildiğinde *ReturnedName* ' nin dönüştürüleceği karakter takımınıdır.

Bu alanın ilk değeri CSAPL 'dir.

IPREQENC (10 basamaklı işaretli tamsayı)

Bu, IPCVAL ya da IPCTYP belirtildiğinde, sorulmuş özellik değerinin dönüştürüleceği kodlamadır.

Bu alanın ilk değeri ENNAT ' tır.

IPRE1 (10 basamaklı işaretli tamsayı)

Bu ayrılmış bir alandır. Bu alanın ilk değeri boş bir karakterdir.

IPRETCSI (10 basamaklı işaretli tamsayı)

Çıkışta, MQINQMP çağrısının **Type** parametresi TYPSTR ise bu değer, döndürülen değer karakter kümesidir.

IPCVAL seçeneği belirtilirse ve dönüştürme başarılı olursa, dönüşte *ReturnedCCSID* alanı, geçirilen değerle aynıdır.

Bu alanın ilk değeri sıfırdır.

IIDC (10 basamaklı işaretli tamsayı)

Çıkışta bu, döndürülen değer kodlamasıdır.

IPCVAL seçeneği belirtilirse ve dönüştürme başarılı olursa, dönüşte *ReturnedEncoding* alanı, geçirilen değerle aynıdır.

Bu alanın ilk değeri ENNAT ' tır.

IPRETNAMCHRP (10 basamaklı işaretli tamsayı)

Sorulmuş özelliğin gerçek adı.

Girişte, MQCHARV yapısının *VSPtr* ya da *VSOffset* alanı kullanılarak bir dizgi arabelleği iletilir. Dizgi arabelleğinin uzunluğu, MQCHARV yapısının *VSBuFSIZE* alanı kullanılarak belirtilir.

MQINQMP çağrısından döndükten sonra, dizgi arabelleği adı tam olarak içerecek kadar uzun olduğu sürece, dizgi arabelleği, sorulmuş özelliğin adıyla tamamlanır. MQCHARV yapısının *VSLength* alanı, özellik adının uzunluğuyla doldurulur. MQCHARV yapısının *VSCCSID* alanı, döndürülen adın karakter kümesini göstermek için doldurulur.

Bu bir giriş/çıkış alanıdır. Bu alanın ilk değeri MQCHARV_DEFAULT.

IPSID (10 basamaklı işaretli tamsayı)

Bu, yapı tanıtıcısıdır. Değer şöyle olmalıdır:

IPSIDV

Sorma iletisi özellik seçenekleri yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri IPSIDV ' dir.

IPTYP (10 basamaklı işaretli tamsayı)

Özelliğin veri tipinin dizgi gösterimi.

Özellik bir MQRFH2 üstbilgisinde belirtildiyse ve MQRFH2 dt özniteliği tanınmıyorsa, özelliğin veri tipini saptamak için bu alan kullanılabilir. *TypeString* , kodlanmış karakter kümesi 1208 'de (UTF-8) döndürülür ve özelliğin dt özniteliğinin değerinin tanınmayan ilk sekiz baytı olur.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri C programlama dilindeki boş dizgi ve diğer programlama dillerindeki 8 boş karakterdir.

IPVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

IPVER1

Sorma iletisi özellik seçenekleri yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

IPVERC

Sorma iletisi özellik seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri IPVER1' dir.

Başlangıçtaki değerler

Çizelge 707. MQIPMO 'daki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
IPSID	IPSIDV	'IMPO'
IPVER	IPVER1	1
IPOPT	IPINQF	
IPREQENC	GİRİŞ	
IPREQCSI	CSAPL	
IPRETENC	GİRİŞ	
IPRETCSI	0	
IPRE1	0	
IPRETAMCHRP		
IPTYP		boşluklar

RPG bildirimi

```
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 IPRETNAMCHRC      61   64I 0 INZ(-3)
D*
D* Property data type as a string
D IPTYP             65   72  INZ
```

IBM i IBM i üzerinde MQMD (İleti tanımlayıcı)

Genel Bakış

Amaç: MQMD yapısı, bir ileti gönderme ve alma uygulamaları arasında dolaşırken uygulama verileriyle birlikte gönderilen denetim bilgilerini içerir. Yapı, MQGET, MQPUT ve MQPUT1 çağrılarında bir giriş/çıkış değiştirgesidir.

Sürüm: MQMD ' nin yürürlükteki sürümü MDVER2. Yapının yalnızca daha yeni sürümlerinde bulunan alanlar, izleyen açıklamalarda bu şekilde tanımlanır.

Sağlanan COPY dosyası, ortam tarafından desteklenen ancak MDVER alanının ilk değeri MDVER1olarak ayarlanmış MQMD ' nin en son sürümünü içeriyor. version-1 yapısında var olmayan alanları kullanmak için uygulamanın MDVER alanını gerekli sürümün sürüm numarasına ayarlaması gerekir.

version-1 yapısına ilişkin bir bildirim MQMD1adıyla kullanılabilir.

Karakter kümesi ve kodlama: MQMD ' deki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama IBM MQ MQI clientolarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Gönderen ve alan kuyruk yöneticileri farklı karakter kümeleri ya da kodlamalar kullanıyorsa, MQMD ' deki veriler otomatik olarak dönüştürülür. Uygulamanın MQMD ' yi dönüştürmesi gerekli değildir.

- [“MQMD ' nin farklı sürümlerini kullanma” sayfa 1093](#)
- [“İleti bağlamı” sayfa 1094](#)
- [“İleti süre bitimi” sayfa 1094](#)
- [“Alanlar” sayfa 1094](#)
- [“Başlangıçtaki değerler” sayfa 1132](#)
- [“RPG bildirimi” sayfa 1133](#)

MQMD ' nin farklı sürümlerini kullanma

version-2 MQMD, genellikle version-1 MQMD ' nin kullanılmasıyla ve ileti verilerinin MQMDE yapısıyla önceden düzeltilmesiyle eşdeğerdir. Ancak, MQMDE yapısındaki tüm alanların varsayılan değerleri varsa, MQMDE atlanabilir. version-1 MQMD ve MQMDE, daha sonra bu kısımda açıklandığı gibi kullanılır.

- MQPUT ve MQPUT1 çağrılarında, uygulama bir version-1 MQMD 'si sağlarsa, uygulama isteğe bağlı olarak ileti verilerine bir MQMDE ile önek verebilir ve MQMD 'deki MDFMT alanını bir MQMDE' nin var olduğunu göstermek için FMMDE olarak ayarlayabilir. Uygulama bir MQMDE sağlamazsa, kuyruk yöneticisi MQMDE ' deki alanlar için varsayılan değerleri varsayar.

Not: version-2 MQMD 'de var olan, ancak version-1 MQMD' de olmayan alanlar, MQPUT ve MQPUT1 çağrılarındaki giriş/çıkış alanlarıdır. Ancak, kuyruk yöneticisi MQPUT ve MQPUT1 çağrılarının çıkışında MQMDE ' deki eşdeğer alanlarda herhangi bir değer döndürmez; uygulama bu çıkış değerlerini gerektiriyorsa, bir version-2 MQMD kullanmalıdır.

- MQGET çağrısında, uygulama version-1 MQMD sağlarsa, kuyruk yöneticisi MQMDE ile döndürülen iletiyi önekler, ancak yalnızca MQMDE ' deki alanlardan birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa. MQMD 'deki MDFMT alanı, bir MQMDE' nin var olduğunu gösteren FMMDE değerine sahip olacaktır.

Kuyruk yöneticisinin MQMDE ' deki alanlar için kullandığı varsayılan değerler, [Çizelge 709 sayfa 1132](#) içinde gösterilen bu alanların başlangıç değerleriyle aynıdır.

Bir ileti iletim kuyruğunda olduğunda, MQMD ' deki bazı alanlar belirli değerlere ayarlanır; ayrıntılar için bkz. [“IBM i üzerinde MQXQH \(İletim kuyruğu üstbilgisi\)” sayfa 1225](#) .

İleti bağlamı

MQMD ' deki bazı alanlar ileti bağlamını içerir. Genellikle:

- *Kimlik bağlamı* , iletiyi ilk olarak koyan uygulamayla ilgilidir
- *Kaynak bağlamı* , iletiyi en son koyan uygulamayla ilgilidir
- *Kullanıcı bağlamı* , iletiyi ilk olarak koyan uygulamayla ilgilidir.

Bu iki uygulama aynı uygulama olabilir, ancak farklı uygulamalar da olabilir (örneğin, bir ileti bir uygulamadan diğerine iletildiğinde).

Kimlik ve kaynak bağlamın genellikle daha önce açıklanan anlamları olsa da, MQMD ' deki her iki bağlam alanı tipinin içeriği, ileti konduğunda belirtilen PM* seçeneklerine bağlıdır. Sonuç olarak, kimlik bağlamı ilk olarak iletiyi koyan uygulamayla ilgili olmak zorunda değildir ve kaynak bağlamı, iletiyi en son koyan uygulamayla ilgili olmak zorunda değildir-bu, uygulama paketinin tasarımına bağlıdır.

İleti bağlamını, yani ileti kanalı aracısını (MCA) hiçbir zaman değişmeyen bir uygulama sınıfı vardır. Uzak kuyruk yöneticilerinden ileti alan MCA ' lar MQPUT ya da MQPUT1 çağrısında PMSETA bağlam seçeneğini kullanır. Bu, alan MCA 'nın gönderen MCA' dan gelen iletiyle birlikte gönderilen ileti bağlamını tam olarak korumasını sağlar. Ancak sonuç, kaynak bağlamın en son iletiyi (alan MCA) koyan uygulamayla değil, bunun yerine iletiyi koyan daha önceki bir uygulamayla (muhtemelen kaynak uygulamanın kendisiyle) ilgili olmasıdır.

Daha fazla bilgi için bkz. [İleti bağlamı](#).

İleti süre bitimi

Yüklenen bir kuyruқта süresi dolmuş olan iletiler (açılmış olan bir kuyruk), süresi sona erdikten sonra makul bir süre içinde otomatik olarak kuyruktan kaldırılır. Bu IBM MQ yayınının diğer bazı yeni özellikleri, yüklenen kuyrukların önceki ürün sürümünden daha az sıklıkta taranmasına neden olabilir, ancak yüklenen kuyruklardaki süresi dolmuş iletiler her zaman, süresi dolduktan sonra makul bir süre içinde kaldırılır.

Alanlar

MQMD yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

MDACC (32 baytlık bit dizgisi)


Hesap simgesi.

Bu, iletinin *kimlik bağlamının* bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.





MDACC , bir uygulamanın, iletinin sonucu olarak yapılan işlerin uygun şekilde ücretlendirilmesine neden olmasını sağlar. Kuyruk yöneticisi bu bilgileri bir bit dizgisi olarak işler ve içeriğini denetlemez.

Kuyruk yöneticisi bu bilgileri oluşturduğunda, aşağıdaki gibi ayarlanır:

- Alanın ilk baytı, izleyen baytlarda var olan muhasebe bilgilerinin uzunluğuna ayarlanır; bu uzunluk, sıfır ile 30 aralığındadır ve ilk baytta ikili tamsayı olarak saklanır.
- İkinci ve sonraki baytlar (uzunluk alanı tarafından belirlendiği şekilde), ortama uygun muhasebe bilgilerine ayarlanır.

–  z/OS ' de muhasebe bilgileri aşağıdaki şekilde ayarlanır:

- z/OS toplu iş için, JES JOB kartındaki ya da EXEC kartındaki bir JES ACCT deyimindeki muhasebe bilgileri (virgül ayırıcıları X'FF ' olarak değiştirilir). Gerekirse, bu bilgiler 31 bayta kesilir.

- TSO için, kullanıcının hesap numarası.
 - CICS için, LU 6.2 iş birimi tanıtıcısı (UEPUOWDS) (26 bayt).
 - IMS için, 8 karakterlik PSB adı 16 karakterlik IMS kurtarma simgesiyle bitleştirilir.
 -  IBM işletim tarihinde, muhasebe bilgileri işin muhasebe koduna ayarlanır.
 -   AIX and Linux' da, muhasebe bilgileri ASCII karakterlerinde sayısal kullanıcı kimliğine ayarlanır.
 -  Windows üzerinde, muhasebe bilgileri sıkıştırılmış biçimde bir Windows NT güvenlik tanıtıcısına (SID) ayarlanır. SID, MDUID alanında saklanan kullanıcı kimliğini benzersiz olarak tanımlar. SID MDACC alanında saklandığında, 6 baytlık Tanıtıcı Yetkisi (SID ' nin üçüncü ve sonraki baytlarında bulunur) atlanır. Örneğin, Windows NT SID 28 bayt uzunluğdaysa, SID bilgilerinin 22 baytı MDACC alanında saklanır.
- Son bayt, aşağıdaki değerlerden biri olan muhasebe simgesi tipine ayarlanır:

ÖZNTLK

CICS LUOW tanıtıcısı.

ATTDOS

PC DOS varsayılan muhasebe simgesi.

ÖZNTLKT

Windows güvenlik tanıtıcısı.

ATT400

IBM i hesap simgesi.

ATTUNX

AIX and Linux sayısal tanıtıcı.

ATTUSR

Kullanıcı tanımlı hesap simgesi.

ATTUNK

Bilinmeyen muhasebe simgesi tipi.

Muhasebe simgesi tipi, yalnızca aşağıdaki ortamlarda belirttik bir değere ayarlanır:

-  AIX

-  IBM i

-  Windows

ve bu sistemlere bağlı IBM MQ MQI clients için.

Diğer ortamlarda, muhasebe simgesi tipi ATTUNK değerine ayarlanır. Bu ortamlarda MDPAT alanı, alınan muhasebe simgesi tipini çıkarmak için kullanılabilir.

- Diğer tüm baytlar ikili sıfır olarak ayarlanır.

MQPUT ve MQPUT1 çağrılar için bu, **PMO** değiştirilmesinde PMSETI ya da PMSETA belirtildiyse bir giriş/çıkış alanıdır. PMSETI ya da PMSETA belirtilmezse, girişte bu alan yoksayılr ve yalnızca çıkış alanıdır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerinin denetleme başlıklı](#) konuya bakın.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen MDACC ögesini içerir. Bu, alıkonan yayınlarla ilgili daha ayrıntılı bilgi için iletiyle birlikte tutulan MDACC değeri olur ("[IBM i üzerinde MQPMO \(Put-message options\)](#)" sayfa 1154 içindeki PMRET açıklamasına bakın), ancak ileti abonelere gönderilen tüm yayınlarda MDACC değerini geçersiz kılacak bir değer sağladıkları için ileti abonelere yayın olarak gönderildiğinde MDACC olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen ikili sıfırdır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır.

Bu alan, kuyruk yöneticisinin karakter kümesine dayalı olarak herhangi bir çeviriye tabi değildir; alan, karakter dizgisi olarak değil, bit dizgisi olarak işlenir.

Kuyruk yöneticisi bu alandaki bilgilerle hiçbir şey yapmaz. Bilgileri muhasebe amacıyla kullanmak istiyorsa, uygulama bilgiyi yorumlamalıdır.

MDACC alanı için aşağıdaki özel değer kullanılabilir:

YOK

Hesap simgesi belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

Bu alanın uzunluğu LNACT tarafından verilir. Bu alanın ilk değeri ACNONE.

MDAID (32 baytlık karakter dizilimi)

Kimlikle ilgili uygulama verileri.

Bu, iletinin *kimlik bağlamının* bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetlemebaşlıklı](#) konuya bakın.

MDAID , uygulama grubu tarafından tanımlanan bilgilerdir ve ileti ya da oluşturucusu hakkında ek bilgi sağlamak için kullanılabilir. Kuyruk yöneticisi bu bilgileri karakter verileri olarak kabul eder, ancak biçimini tanımlamaz. Kuyruk yöneticisi bu bilgileri oluşturduğunda tamamen boş olur.

MQPOT ve MQPUT1 çağrıları için bu, **PMO** değiştirgesinde PMSETI ya da PMSETA belirtildiyse bir giriş/çıkış alanıdır. Boş bir karakter varsa, boş değer ve aşağıdaki karakterler kuyruk yöneticisi tarafından boşluklara dönüştürülür. PMSETI ya da PMSETA belirtilmezse, girişte bu alan yoksayılr ve yalnızca çıkış alanıdır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetlemebaşlıklı](#) konuya bakın.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen MDAID ögesini içerir. Bu değer, alıkonursa iletiyle birlikte tutulan MDAID değeri olur (alıkonan yayınlarla ilgili daha fazla ayrıntı için PMRET açıklamasına bakın), ancak kendilerine gönderilen tüm yayınlarda MDAID değerini geçersiz kılacak bir değer sağladıkları için ileti abonelere yayın olarak gönderildiğinde MDAID olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen boştur.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu LNAIDD tarafından verilir. Bu alanın ilk değeri 32 boş karakterdir.

MDAOD (4 baytlık karakter dizilimi)

Kaynak ile ilgili uygulama verileri.

Bu, iletinin *kaynak bağlamının* bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetlemebaşlıklı](#) konuya bakın.

MDAOD , uygulama grubu tarafından tanımlanan ve iletinin kökenine ilişkin ek bilgi sağlamak için kullanılabilecek bilgilerdir. Örneğin, kimlik verilerinin güvenilir olup olmadığını belirtmek için uygun kullanıcı yetkisiyle çalışan uygulamalar tarafından ayarlanabilir.

Kuyruk yöneticisi bu bilgileri karakter verileri olarak kabul eder, ancak biçimini tanımlamaz. Kuyruk yöneticisi bu bilgileri oluşturduğunda tamamen boş olur.

MQPOT ve MQPUT1 çağrıları için, **PMO** değiştirgesinde PMSETA belirtildiyse bu bir giriş/çıkış alanıdır. Alan içinde boş bir karakteri izleyen bilgiler atılır. Boş karakter ve aşağıdaki karakterler, kuyruk yöneticisi tarafından boşluklara dönüştürülür. PMSETA belirtilmezse, girişte bu alan yoksayılr ve salt çıkış alanıdır.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen MDAOD ögesini içerir. Bu değer, alıkonursa iletiyle birlikte tutulan MDAOD değeri olur (alıkonan yayınlarla ilgili daha fazla ayrıntı için PMRET açıklamasına bakın), ancak kendilerine gönderilen tüm yayınlarda MDAOD değerini geçersiz kılacak bir değer sağladıkları için ileti abonelere yayın olarak gönderildiğinde MDAOD olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen boştur.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu LNAORD tarafından verilir. Bu alanın ilk değeri 4 boş karakterdir.

MDBOC (10 basamaklı işaretli tamsayı)

Geri dönüş sayacı.

Bu sayı, MQGET çağrısının bir iş biriminin parçası olarak daha önce kaç kez döndürüldüğünü ve daha sonra kaç kez geri çekildiğini gösterir. İleti içeriğine dayalı işleme hatalarının saptanmasında uygulamaya yardımcı olarak sağlanır. Sayım, GMBRW* seçeneklerinden herhangi birini belirten MQGET çağrılarını dışlar.

Bu sayının doğruluğu **HardenGetBackout** kuyruk özneteliğinden etkilenir; bkz. [“Kuyruklara ilişkin öznetelikler” sayfa 1345.](#)

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. MQPUT ve MQPUT1 çağrıları için yoksayılr. Bu alanın ilk değeri 0 'dir.

MDCID (24 baytlık bit dizgisi)

İlinti tanıtıcısı.

Bu, uygulamanın bir iletiyi başka bir iletiyle ilişkilendirmek ya da iletiyi uygulamanın gerçekleştirdiği diğer işle ilişkilendirmek için kullanabileceği bir bayt dizgisidir. İlinti tanıtıcısı, iletinin kalıcı bir özelliğidir ve kuyruk yöneticisinin yeniden başlatması boyunca devam eder. İlinti tanıtıcısı bir karakter dizgisi değil bayt dizgisi olduğundan, ileti bir kuyruk yöneticisinden diğerine aktığında, ilinti tanıtıcısı karakter kümeleri arasında dönüştürülmez.

MQPUT ve MQPUT1 çağrıları için uygulama herhangi bir değeri belirtebilir. Kuyruk yöneticisi bu değeri iletiyle iletir ve ileti için alma isteğini gönderen uygulamaya teslim eder.

Uygulama PMNCID belirtirse, kuyruk yöneticisi iletiyle birlikte gönderilen benzersiz bir ilinti tanıtıcısı oluşturur ve MQPUT ya da MQPUT1 çağrısının çıkışında gönderen uygulamaya döndürülür.

Oluşturulan bu ilinti tanıtıcısı korunursa iletiyle birlikte alıkonur ve ileti, MQSUB çağrısında iletilen MQSD ' de SDCID alanında CINONE belirten abonelere yayın olarak gönderildiğinde ilinti tanıtıcısı olarak kullanılır.

Alıkonan yayınlarla ilgili daha fazla ayrıntı için bkz. [“IBM i üzerinde MQPMO \(Put-message options\)” sayfa 1154](#)

Kuyruk yöneticisi ya da bir ileti kanalı aracısı bir rapor iletisi oluşturduğunda, MDCID alanını özgün iletinin ROCMTC ya da ROPCI MDREP alanıyla belirtilen şekilde ayarlar. Rapor iletileri oluşturan uygulamalar da bunu yapmalıdır.

MQGET çağrısı için MDCID , kuyruktan alınacak belirli bir iletiyi seçmek için kullanılacak beş alandan biridir. Bu alana ilişkin değerlerin nasıl belirleneceğine ilişkin ayrıntılar için MDMID alanının açıklamasına bakın.

İlinti tanıtıcısı olarak CINONE belirtilmesi MOCORI belirtilmemesiyle aynı etkiye sahiptir; yani, herhangi bir ilinti tanıtıcısı eşleşir.

MQGET çağrısındaki **GMO** değiştirgesinde GMMUC seçeneği belirtilirse, bu alan yoksayılr.

Bir MQGET çağrısından döndüğünüzde, MDCID alanı döndürülen iletinin ilinti tanıtıcısına (varsa) ayarlanır.

Aşağıdaki özel değerler kullanılabilir:

TAMAMDİR.

İlinti tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

SINEMALAR

İleti, yeni bir oturumun başlangıcını belirtir.

Bu değer, CICS bridge tarafından yeni bir oturum başlangıcı, yani yeni bir ileti dizisinin başlangıcı olarak tanınır.

MQGET çağrısı için bu bir giriş/çıkış alanıdır. MQPUT ve MQPUT1 çağrıları için, PMNCID belirtilmediyse bu bir giriş alanıdır ve PMNCID belirtildiyse bir çıkış alanıdır. Bu alanın uzunluğu LNCID tarafından verilir. Bu alanın ilk değeri CINONE.

MDCSI (10 basamaklı işaretli tamsayı)

Bu, iletideki karakter verilerinin karakter kümesi tanıtıcısını belirtir.

Not: MQMD 'deki karakter verileri ve çağrılarda parametre olan diğer IBM MQ veri yapıları, kuyruk yöneticisinin karakter kümesinde olmalıdır. Bu, kuyruk yöneticisinin **CodedCharSetId** özneliği tarafından tanımlanır; bu özneliğin ayrıntıları için bkz. [“IBM i üzerindeki kuyruk yöneticisine ilişkin öznelikler” sayfa 1376](#).

Aşağıdaki özel değerler kullanılabilir:

CSQM

Kuyruk yöneticisinin karakter takımı tanıtıcısı.

İletideki karakter verileri kuyruk yöneticisinin karakter takımındaki karakterdir.

MQPUT ve MQPUT1 çağrılarında, kuyruk yöneticisi iletiyle birlikte gönderilen MQMD 'de bu değeri kuyruk yöneticisinin gerçek karakter kümesi tanıtıcısına değiştirir. Sonuç olarak, MQGET çağrısı CSQM değerini hiçbir zaman döndürmez.

CSINHT

Bu yapının karakter kümesi tanıtıcısını edin.

İletideki karakter verileri bu yapıyla aynı karakter kümesinde; bu, kuyruk yöneticisinin karakter kümesidir. (Yalnızca MQMD için, CSINHT CSQM ile aynı anlama sahiptir).

Kuyruk yöneticisi, iletiyle birlikte gönderilen MQMD 'deki bu değeri MQMD' nin gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı CSINHT değerini döndürmez.

MQMD 'deki MDPAT alanının değeri ATBRKR ise, CSINHT kullanılamaz.

CSEMBD

Gömülü karakter kümesi tanıtıcısı.

İletideki karakter verileri, ileti verilerinin kendisinde bulunan tanıtıcıya sahip bir karakter kümesinde bulunur. Verilerin farklı bölümlerine uygulanarak, ileti verilerinin içine yerleştirilmiş herhangi bir sayıda karakter kümesi tanıtıcısı olabilir. Bu değer, karakter kümelerinin karışımındaki verileri içeren PCF iletileri için kullanılmalıdır. PCF iletilerinin biçim adı FMPCF.

Bu değeri yalnızca MQPUT ve MQPUT1 çağrılarında belirtin. MQGET çağrısında belirtilirse, iletinin dönüştürülmesini engeller.

MQPUT ve MQPUT1 çağrılarında, kuyruk yöneticisi iletiyle birlikte gönderilen MQMD 'deki CSQM ve CSINHT değerlerini önceden açıklandığı gibi değiştirir, ancak MQPUT ya da MQPUT1 çağrısında belirtilen MQMD' yi değiştirmez. Belirtilen değer üzerinde başka bir denetim gerçekleştirilmez.

İletileri alan uygulamalar bu alanı uygulamanın beklediği değerle karşılaştırmalıdır; değerler farklıysa, uygulamanın iletideki karakter verilerini dönüştürmesi gerekebilir.

MQGET çağrısında GMCONV seçeneği belirtilirse, bu alan bir giriş/çıkış alanıdır. Uygulama tarafından belirlenen değer, gerekirse ileti verilerinin dönüştürülmesi gereken kodlanmış karakter takımı tanıtıcısıdır. Dönüştürme başarılı ya da gereksizse, değer değişmez (CSQM ya da CSINHT değerinin gerçek değere dönüştürülmesi dışında). Dönüştürme başarısız olursa, MQGET çağrısından sonraki değer, uygulamaya döndürülen dönüştürülmemiş iletinin kodlanmış karakter kümesi tanıtıcısını gösterir.

Tersi durumda, bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş alanıdır. Bu alanın ilk değeri CSQM 'dir.

MDENC (10 basamaklı işaretli tamsayı)

İleti verilerinin sayısal kodlaması.

Bu, iletideki sayısal verilerin sayısal kodlamasını belirtir; MQMD yapısındaki sayısal veriler için geçerli değildir. Sayısal kodlama, ikili tamsayılar, paketlenmiş ondalık tamsayılar ve kayan noktalı sayılar için kullanılan gösterimi tanımlar.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Kuyruk yöneticisi, alanın geçerli olup olmadığını denetlemez. Aşağıdaki özel değer tanımlandı:

GIRIŞ

Yerli makine kodlaması.

Kodlama, uygulamanın çalıştığı programlama dili ve makine için varsayılan değerdir.

Not: Bu değişmez değeri, programlama diline ve çevreye bağlıdır. Bu nedenle, uygulamaların, uygulamanın çalışacağı ortama uygun üstbilgi, makro, COPY ya da INCLUDE dosyaları kullanılarak derlenmesi gerekir.

İletileri koyan uygulamaların olağan durumda ENNAT belirtmesi gerekir. İletileri alan uygulamalar bu alanı ENNAT değeriyle karşılaştırmalıdır; değerler farklıysa, uygulamanın iletideki sayısal verileri dönüştürmesi gerekebilir. GMCONV seçeneği, kuyruk yöneticisinden iletiyi MQGET çağrısının işlenmesinin bir parçası olarak dönüştürmesini istemek için kullanılabilir.

MQGET çağrısında GMCONV seçeneği belirtilirse, bu alan bir giriş/çıkış alanıdır. Uygulama tarafından belirtilen değer, gerekirse ileti verilerinin dönüştürülmesi gereken kodlamadır. Dönüştürme başarılı ya da gereksizse, değer değişmez. Dönüştürme başarısız olursa, MQGET çağrısından sonraki değer, uygulamaya döndürülen dönüştürülmemiş iletinin kodlamasını gösterir.

Diğer durumlarda, bu alan MQGET çağrısına ilişkin bir çıkış alanıdır ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri ENNAT ' tır.

MDEXP (10 basamaklı işaretli tamsayı)

Mesaj ömrü.

Bu, iletiyi koyan uygulama tarafından belirlenen, saniyenin onda biri olarak ifade edilen bir zaman dönemidir. İleti, bu süre dolmadan önce hedef kuyruktan kaldırılmamışsa atılmaya uygun hale gelir.

Değer, iletinin hedef kuyruğa harcandığı süreyi ve uzak bir kuyruğa yerleştirilmesi durumunda, herhangi bir ara iletim kuyruğunda harcandığı süreyi yansıtacak şekilde azaltılır. Bunlar önemliyse, iletim sürelerini yansıtmak için ileti kanalı ajanları tarafından da azaltılabilir. Benzer şekilde, bu iletiyi başka bir kuyruğa ileten bir uygulama, iletiyi önemli bir süre alıkoysa, gerekirse değeri azaltılabilir. Ancak, süre bitim süresi yaklaşık olarak kabul edilir ve değer küçük zaman aralıklarını yansıtmak için azaltılmasına gerek yoktur.

İleti MQGET çağrısı kullanılarak bir uygulama tarafından alındığında, MDEXP alanı hala devam eden özgün süre bitim süresini gösterir.

Bir iletinin süre bitim tarihi geçtikten sonra, kuyruk yöneticisi tarafından atılmaya uygun hale gelir. Yürürlükteki somutlamalarda, süresi dolmamış bir iletiyi döndüren bir göz atma ya da göz atmama MQGET çağrısı oluştuğunda ileti atılır. Örneğin, MQGMO ' daki GMMO alanı için FIFO sıralı kuyruğundan MONONE okuma değeri ayarlanmış bir göz atmama MQGET çağrısı, süresi dolan tüm iletilerin ilk süresi dolmamış iletiye kadar atılmasına neden olur. Öncelikli sıralı bir kuyruğa, aynı çağrı, süresi dolan daha yüksek öncelikli iletileri ve süresi dolmamış ilk iletiden önce kuyruğa gelen aynı önceliğe sahip iletileri atar.

Süresi dolan bir ileti bir uygulamaya hiçbir zaman döndürülmez (göz atma ya da göz atmama bir MQGET çağrısıyla); bu nedenle, başarılı bir MQGET çağrısından sonra ileti tanımlayıcısının MDEXP alanındaki değer sıfırdan büyük ya da özel değer EIULIM.

Bir ileti uzak kuyruğa konursa, ileti hedef kuyruğa ulaşmadan önce, ileti bir ara iletim kuyruğundayken süre bitimine uğrayabilir (ve atılabilir).

İleti ROEXP* rapor seçeneklerinden birini belirtmişse, süresi dolan bir ileti atıldığında bir rapor oluşturulur. Bu seçeneklerden hiçbiri belirtilmezse, böyle bir rapor oluşturulmaz; iletinin bu zaman döneminden sonra artık ilgili olmadığı varsayılır (belki de daha sonraki bir iletinin yerine geçmesinden dolayı).

İletileri süre bitimine dayalı olarak gönderen diğer herhangi bir program, istenirse uygun bir rapor iletisi de göndermelidir.

Not:

1. MDEXP sıfır zamanlı bir ileti konursa, MQPUT ya da MQPUT1 çağrısı başarısız olur; neden kodu RC2013; bu durumda rapor iletisi oluşturulmaz.
2. Süresi dolmuş bir ileti daha sonraya kadar atılamayabileceğinden, kuyrukta süresi dolmuş olan ve bu nedenle alınmaya uygun olmayan iletiler olabilir. Yine de bu iletiler, derinlik tetikleme de dahil olmak üzere, tüm amaçlar için kuyruktaki iletilerin sayısına dahil olur.
3. Bir süre sonu raporu, istenirse, ileti atıldığında değil, gerçekten atıldığında oluşturulur.
4. Süresi dolan bir iletinin atılması ve istenirse, süre bitim raporunun oluşturulması, ileti bir iş birimi içinde çalışan bir MQGET çağrısının sonucu olarak atılmak üzere zamanlanmış olsa da, hiçbir zaman uygulamanın iş biriminin bir parçası değildir.
5. Süresi dolmuş bir ileti bir iş birimi içindeki bir MQGET çağrısı tarafından alınırsa ve iş birimi daha sonra geriletirse, ileti yeniden alınmadan önce atılabilir.
6. Süresi dolmuş bir ileti GMLK ile bir MQGET çağrısı tarafından kilitlenirse, ileti GMMUC ile bir MQGET çağrısıyla alınmadan önce atılabilir; bu durumda, sonraki MQGET çağrısında neden kodu RC2034 döndürülür.
7. Süre bitimi sıfırdan büyük olan bir istek iletisi alındığında, uygulama yanıt iletisini gönderirken aşağıdaki işlemlerden birini gerçekleştirebilir:
 - İstek iletisinden yanıt iletisine kalan süre bitimini kopyalayın.
 - Yanıt iletisindeki süre bitimini sıfırdan büyük belirtik bir değere ayarlayın.
 - Yanıt iletisindeki süre bitimini EIULIM olarak ayarlayın.

Yapılacak işlem, uygulama paketinin tasarımına bağlıdır. Ancak, iletileri bir teslim edilmeyen ileti kuyruğuna koymak için varsayılan işlem, iletinin kalan süre bitimini korumak ve azaltmaya devam etmek olmalıdır.

8. Tetikleyici iletiler her zaman EIULIM ile oluşturulur.
9. MDFMT adı FMXQH olan bir ileti (normalde bir iletim kuyruğunda), MQXQH içinde ikinci bir ileti tanımlayıcısına sahiptir. Bu nedenle, kendisiyle ilişkilendirilmiş iki MDEXP alanı vardır. Bu durumda aşağıdaki ek puanlar dikkate alınmalıdır:
 - Bir uygulama uzak kuyruğa bir ileti koyduğunda, kuyruk yöneticisi ilk olarak iletiyi yerel bir iletim kuyruğuna yerleştirir ve uygulama iletisi verilerinin başına MQXQH yapısını ekler. Kuyruk yöneticisi, iki MDEXP alanının değerlerini uygulama tarafından belirtilenle aynı olacak şekilde ayarlar.

Bir uygulama bir iletiyi doğrudan yerel iletim kuyruğuna koyarsa, ileti verileri zaten bir MQXQH yapısıyla başlamalıdır ve biçim adı FMXQH olmalıdır (ancak kuyruk yöneticisi bunu uygulamaz). Bu durumda, uygulamanın bu iki MDEXP alanının değerlerini aynı olacak şekilde ayarlamasına gerek yoktur. (Kuyruk yöneticisi, MQXQH içindeki MDEXP alanının geçerli bir değer içerdiğini ya da ileti verilerinin bunu içerecek kadar uzun olduğunu denetlemez.)
 - Bir kuyruktan MDFMT adı FMXQH olan bir ileti alındığında (bu normal ya da bir iletim kuyruğu olsun), kuyruk yöneticisi bu MDEXP alanlarının her ikisini de kuyrukta beklerken harcanan süre ile belirler. İleti verileri MDEXP alanını MQXQH ' ye dahil edecek kadar uzun değilse hata oluşmaz.
 - Kuyruk yöneticisi, iletinin atılmaya uygun olup olmadığını sınamak için ayrı ileti tanımlayıcısında (MQXQH yapısına gömülü ileti tanımlayıcısında değil) MDEXP alanını kullanır.
 - Bu nedenle, iki MDEXP alanının ilk değerleri farklıysa, ileti alındığında ayrı ileti tanımlayıcısındaki MDEXP süresi sıfırdan büyük olabilir (bu nedenle, ileti atılabilir değildir), ancak MQXQH ' deki MDEXP alanına göre süre geçmiştir. Bu durumda MQXQH ' deki MDEXP alanı sıfır olarak ayarlanır.

Şu özel değer tanınır:

EIULIM

Sınırsız yaşam süresi.

İletinin sınırsız süre bitimi var.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri EIULIM 'dir.

MDFB (10 basamaklı işaretli tamsayı)

Geribildirim ya da neden kodu.

Bu, raporun niteliğini belirtmek için MTRPRT tipinde bir iletiyle birlikte kullanılır ve yalnızca bu ileti tipiyle anlamlıdır. Alan, FB* değerlerinden birini ya da RC* değerlerinden birini içerebilir. Geribildirim kodları aşağıdaki gibi gruplanır:

FBNONE

Geribildirim sağlanmadı.

FBSFST

Sistem tarafından oluşturulan geribildirim için en düşük değer.

FBSLST

Sistem tarafından oluşturulan geribildirim için en yüksek değer.

FBSLST aracılığıyla sistem tarafından oluşturulan geribildirim kodları aralığı, daha sonra bu bölümde (FB*) listelenen genel geribildirim kodlarını ve ileti hedef kuyruğa yerleştirilemediğinde ortaya çıkabilecek neden kodlarını (RC*) içerir.

FBAFST

Uygulama tarafından oluşturulan geribildirim için en düşük değer.

FBALST

Uygulama tarafından oluşturulan geribildirim için en yüksek değer.

Rapor iletileri oluşturan uygulamalar, kuyruk yöneticisi ya da ileti kanalı aracısı tarafından oluşturulan rapor iletilerinin benzetimini yapmak istemedikçe, sistem aralığındaki (FBQUIT dışında) geribildirim kodlarını kullanmamalıdır.

MQPUT ya da MQPUT1 çağrılarında, belirtilen değer FBNONE olmalı ya da sistem aralığı ya da uygulama aralığı içinde olmalıdır. Bu, MDMT değeri ne olursa olsun denetlenir.

Genel geribildirim kodları:

FBCOA

Hedef kuyruğa varış onayı (bkz. ROCOA).

FBCOD

Alan uygulamaya teslim onayı (bkz. ROCOD).

FBEXP

İletinin süresi doldu.

İleti, süresi dolmadan önce hedef kuyruktan kaldırılmadığı için atıldı.

FBPAN

Pozitif eylem bildirim (bkz. ROPAN).

FBNAN

Negatif işlem bildirim (bkz. RONAN).

FBIT

Uygulama sona ermelidir.

Bu, çalışmakta olan bir uygulama programının eşgörünümlerinin sayısını denetlemek için bir iş yükü zamanlama programı tarafından kullanılabilir. Bu geribildirim kodunu içeren bir MTRPRT iletilerinin uygulama programının bir eşgörünümüne gönderilmesi, o örneğe işlenmeyi durdurması gerektiğini belirtir. Ancak, bu kurala uymak uygulama için bir sorundur; kuyruk yöneticisi tarafından uygulanmaz.

IMS-bridge feedback codes: IMS köprüsü sıfır olmayan bir IMS-OTMA algılama kodu aldığı anda, IMS köprüsü algılama kodunu onaltıdan ondalığa dönüştürür, FBIERR (300) değerini ekler ve sonucu yanıt

iletisinin MDFB alanına yerleştirir. Bu, bir IMS-OTMA hatası ortaya çıktığında FBIFST (301)-FBILST (399) aralığında bir değer içeren geribildirim koduyla sonuçlanır.

IMS köprüsü tarafından aşağıdaki geribildirim kodları oluşturulabilir:

FBDLZ

Veri uzunluğu sıfır.

İletinin uygulama verilerinde bir kesim uzunluğu sıfırdı.

FBDLN

Veri uzunluğu negatif.

İletinin uygulama verilerinde bir kesim uzunluğu negatif.

FBDLTB

Veri uzunluğu çok büyük.

İletinin uygulama verilerinde kesim uzunluğu çok büyüktü.

FBBUFO

Arabellek taşması.

Uzunluk alanlarından birinin değeri, verilerin ileti arabelleğine taşmasına neden olur.

FBLOB1

Bir hata ile uzunluk.

Uzunluk alanlarından birinin değeri bir byte çok kısa.

FBIIH.

MQIIH yapısı geçerli değil ya da eksik.

MQMD ' deki MDFMT alanı FMIMSbelirtiyor, ancak ileti geçerli bir MQIIH yapısıyla başlamıyor.

FBNAFI

Kullanıcı kimliği IMSiçinde kullanım için yetkili değil.

İleti tanımlayıcı MQMD ' de bulunan kullanıcı kimliği ya da MQIIH yapısındaki IIAUT alanında bulunan parola, IMS köprüsü tarafından gerçekleştirilen geçerlilik denetiminde başarısız oldu. Sonuç olarak, ileti IMS' e iletilmedi.

FBIERR

IMStarafından beklenmeyen bir hata döndürüldü.

IMSBeklenmeyen bir hata döndürdü. Hatayla ilgili daha fazla bilgi için IMS köprüsünün bulunduğu sistemdeki IBM MQ hata günlüğüne bakın.

FBIFST

IMStarafından oluşturulan geribildirim için en düşük değer.

IMStarafından oluşturulan geribildirim kodları FBIFST (300) ile FBILST (399) aralığını kaplar. IMS-OTMA algılama kodunun kendisi MDFB eksi FBIERR 'dir.

FILST (FBILST)

IMStarafından oluşturulan geribildirim için en yüksek değer.

CICS-bridge geribildirim kodları: CICS bridgetarafından aşağıdaki geribildirim kodları oluşturulabilir:

FBCAAB

Uygulama olağandışı bitti.

İletide belirtilen uygulama programı olağandışı bitti. Bu geribildirim kodu yalnızca MQDLH yapısının DLREA alanında oluşur.

FBCANLAR

Uygulama başlatılmıyor.

İletide belirtilen uygulama programına ilişkin EXEC CICS LINK başarısız oldu. Bu geribildirim kodu yalnızca MQDLH yapısının DLREA alanında oluşur.

FBCBRF

CICS bridge olađan hata işlemeyi tamamlamadan olađandışı bitti.

FBCCESE

Karakter takımı tanıtıcısı geçersiz.

FBCIHE

CICS bilgi üstbilgisi yapısı eksik ya da geçersiz.

FBCCAE

CICS iletişim alanı uzunluđu geçersiz.

FBCCE

İlinti tanıtıcısı geçersiz.

FBCDLQ

Gitmeyen ileti kuyruđu kullanılmıyor.

CICS bridge görevi, bu isteđin yanıtını teslim edilmeyen iletiler kuyruđuna kopyalayamadı. İstek geri çekildi.

FBCENE

Kodlama geçerli deđil.

FBCIN

CICS bridge beklenmeyen bir hatayla karşılaştı.

Bu geribildirim kodu yalnızca MQDLH yapısının DLREA alanında oluşur.

FBCNTA

Kullanıcı kimliđi yetkili deđil ya da parola geçersiz.

Bu geribildirim kodu yalnızca MQDLH yapısının DLREA alanında oluşur.

FBCUBO

İş birimi geri çekildi.

Aşađıdaki nedenlerden biri nedeniyle iş birimi geriletildi:

- Aynı iş birimi içindeki başka bir istek işlenirken bir hata saptandı.
- İş birimi devam ederken bir CICS olađandışı sonu oluştu.

FBCUWE

İş birimi denetim alanı CIUOW geçersiz.

MQ neden kodları: Kural dışı durum raporu iletileri için, MDFB bir MQ neden kodu içerir. Olası neden kodları şunlardır:

RC2051

(2051, X'803 ') Kuyruk için çağrılarını engelleme.

RC2053

(2053, X'805 ') Kuyruk zaten ileti sayısı üst sınırını içeriyor.

RC2035

(2035, X'7F3') Erişim için yetkili deđil.

RC2056

(2056, X'808 ') Diskte kuyruk için kullanılabilir alan yok.

RC2048

(2048, X'800 ') Kuyruk kalıcı iletileri desteklemiyor.

RC2031

(2031, X'7EF') Kuyruk yöneticisi için ileti uzunluđu üst sınırından fazla.

RC2030

(2030, X'7EE') İleti uzunluđu kuyruk için üst sınırdan fazla.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş alanıdır. Bu alanın ilk deđeri FBNONE 'dir.

MDFMT (8 baytlık karakter dizgisi)

İleti verilerinin biçim adı.

Bu, iletiyi gönderen kişinin iletideki verilerin niteliğini alıcıya göstermek için kullanabileceği bir addır. Kuyruk yöneticisinin karakter takımındaki tüm karakterler ad için belirtilebilir, ancak adın aşağıdakiler ile sınırlanması önerilir:

- Büyük A-Z arası
- 0-9 arası sayısal rakamlar

Başka karakterler kullanılırsa, adı gönderen ve alan kuyruk yöneticilerinin karakter kümeleri arasında çevirmek mümkün olmayabilir.

Ad, alanın uzunluğuna kadar boşluklarla ya da alan sonundan önce adı sonlandırmak için kullanılan boş karakterle doldurulacaktır; boş değer ve sonraki karakterler boşluk olarak kabul edilir. Başında ya da içinde boşluk olan bir ad belirlemeyin. MQGET çağrısı için, kuyruk yöneticisi alanın uzunluğuna kadar boşluklarla doldurarak adı döndürür.

Kuyruk yöneticisi, adın önceden açıklanan önerilere uygun olup olmadığını denetlemez.

Büyük ve küçük harf karışık olarak "MQ" ile başlayan adlar, kuyruk yöneticisi tarafından tanımlanan anlamlara sahiptir; kendi biçimleriniz için bu harflerle başlayan adları kullanmamalısınız. Kuyruk yöneticisi yerleşik biçimleri şunlardır:

FMNONE

Biçim adı yok.

Verilerin türü tanımlanmamış. Bu, ileti GMCONV seçeneği kullanılarak bir kuyruktan alındığında verilerin dönüştürülemeyeceği anlamına gelir.

MQGET çağrısında GMCONV belirtilirse ve iletideki verilerin karakter kümesi ya da kodlaması **MSGDSC** değiştirilmesinde belirtilenden farklıysa, ileti aşağıdaki tamamlanma ve neden kodlarıyla (başka hata olmadığı varsayılarak) döndürülür:

- İletinin başında FMNONE verileri varsa, tamamlanma kodu CCWARN ve neden kodu RC2110 .
- FMNONE verileri iletinin sonundaysa (bir ya da daha çok MQ üstbilgi yapısının önünde), tamamlanma kodu CCOK ve neden kodu RCNONE. MQ üstbilgi yapıları, bu durumda istenen karakter kümesine ve kodlamaya dönüştürülür.

FMADMN

Komut sunucusu isteği/yanıt iletisi.

İleti, programlanabilir komut biçimindeki (PCF) bir komut sunucusu isteği ya da yanıt iletisidir. MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir. Programlanabilir komut biçimi iletilerinin kullanılmasına ilişkin ek bilgi için [Programlanabilir Komut Biçimlerinin Kullanılması](#) başlıklı konuya bakın.

FMCICS

CICS bilgi üstbilgisi.

İleti verileri, uygulama verilerinin izlediği CICS bilgi üstbilgisi MQCIH ile başlar. Uygulama verilerinin biçim adı, MQCIH yapısındaki CIFMT alanı tarafından verilir.

FMCM1

1 komut yanıt iletisi yazın.

İleti, nesne sayısını, tamamlanma kodunu ve neden kodunu içeren bir MQSC komut sunucusu yanıt iletisidir. MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMCM2

2 komut yanıt iletisi yazın.

İleti, istenen nesnelere ilişkin bilgileri içeren bir MQSC komut sunucusu yanıt iletisidir. MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMDLH

Ölmeyen harf üstbilgisi.

İleti verileri, giden harf üstbilgisi MÖDLH ile başlar. Özgün iletideki veriler MÖDLH yapısını hemen izler. Özgün ileti verilerinin biçim adı, MÖDLH yapısındaki DLFMT alanı tarafından verilir; bu yapının ayrıntıları için bkz. "MÖDLH (Ölü harf üstbilgisi)- IBM i" sayfa 1049 . MÖGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

MDFMT FMDLH içeren iletiler için COA ve COD raporları oluşturulmaz.

FMDH

Dağıtım listesi üstbilgisi.

İleti verileri dağıtım listesi üstbilgisi MÖDH ile başlar; bu, MÖOR ve MÖPMR kayıtlarının dizilerini içerir. Dağıtım listesi üstbilgisini ek veriler izleyebilir. MÖDH yapısındaki DHFMT alanı tarafından verilen ek verilerin biçimi (varsa); bu yapının ayrıntıları için bkz. "IBM i üzerinde MÖDH (Dağıtım üstbilgisi)" sayfa 1044 . MÖGET çağrısında GMCONV seçeneği belirtilirse, FMDH biçimli iletiler dönüştürülebilir.

FMEVNT

Olay iletisi.

İleti, oluşan bir olayı bildiren bir MÖ olay iletisidir. Olay iletileri programlanabilir komutlarla aynı yapıya sahiptir; bu yapıyla ilgili daha fazla bilgi için bkz. Komutlar ve yanıtlar için yapılar. Olaylarla ilgili bilgi için Olay izlemebaşlıklı konuya bakın.

Version-1 olay iletileri, MÖGET çağrısında GMCONV seçeneği belirtilirse dönüştürülebilir.

FMIMS

IMS bilgi üstbilgisi.

İleti verileri, uygulama verilerinin izlediği IMS bilgi üstbilgisi MÖIIH ile başlar. Uygulama verilerinin biçim adı, MÖIIH yapısındaki *IIFMT* alanı tarafından verilir. MÖGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMIMVS

IMS değişken dizgisi.

İleti, 11zzccc biçiminde bir dizgi olan IMS değişken dizgisidir; burada:

11

IMS değişken dizgi ögesinin toplam uzunluğunu belirten 2 baytlık bir alandır. Bu uzunluk, 11 (2 bayt) uzunluğuna, zz (2 bayt) uzunluğuna ve karakter dizgisinin kendisinin uzunluğuna eşittir. 11 , MDENC alanı tarafından belirtilen kodlamada 2 baytlık bir ikili tamsayıdır.

zz

IMS için önemli olan işaretleri içeren 2 baytlık bir alandır. zz , iki adet 1 baytlık bit dizilim alanından oluşan ve gönderenden alıcıya değiştirilmeden iletilen bir bayt dizgisidir (yani, zz herhangi bir dönüştürmeye tabi değildir).

ccc

11-4 karakterlerini içeren değişken uzunluklu bir karakter dizilimidir. ccc , MDCSI alanı tarafından belirtilen karakter kümesinde bulunur.

MÖGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMMDE

İleti tanımlayıcı uzantısı.

İleti verileri, ileti tanımlayıcı uzantısı MÖMDE ile başlar ve isteğe bağlı olarak diğer veriler (genellikle uygulama iletisi verileri) tarafından izlenir. MÖMDE 'yi izleyen verilerin biçim adı, karakter kümesi ve kodlaması, MÖMDE' deki MEFMT, MECSI ve MEENC alanları tarafından verilir. Bu yapının ayrıntıları için bkz. "IBM i üzerinde MÖMDE (İleti tanımlayıcı uzantısı)" sayfa 1134 . MÖGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMPCF

Programlanabilir komut biçiminde (PCF) kullanıcı tanımlı ileti.

İleti, programlanabilir komut biçimi (PCF) iletisinin yapısına uyan kullanıcı tanımlı bir iletidir. MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir. Programlanabilir komut biçimi iletilerinin kullanılmasına ilişkin ek bilgi için [Programlanabilir Komut Biçimlerinin Kullanılması](#) başlıklı konuya bakın.

FMRMH

Başvuru iletisi üstbilgisi.

İleti verileri, başvuru iletisi üstbilgisi MQRMH ile başlar ve isteğe bağlı olarak diğer veriler tarafından takip edilir. Verilerin biçim adı, karakter kümesi ve kodlaması, MQRMH 'deki RMFMT, RMCSive RMENC alanları tarafından verilir. Bu yapının ayrıntıları için bkz. [“IBM i üzerinde MQRMH \(Başvuru iletisi üstbilgisi\)” sayfa 1180](#) . MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMRFH

Kurallar ve biçimlendirme üstbilgisi.

İleti verileri, kurallar ve biçimleme üstbilgisi MQRFH ile başlar ve isteğe bağlı olarak diğer veriler tarafından takip edilir. Verilerin biçim adı, karakter kümesi ve kodlaması (varsa), MQRFH 'deki RFFMT, RFCSive RFENC alanları tarafından verilir. MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMRFH2

Kurallar ve biçimleme üstbilgisi sürüm 2.

İleti verileri, version-2 kuralları ve MQRFH2 üstbilgisiyle başlar ve isteğe bağlı olarak diğer veriler tarafından takip edilir. İsteğe bağlı verilerin biçim adı, karakter kümesi ve kodlaması (varsa), MQRFH2 içindeki RF2FMT, RF2CSive RF2ENC alanları tarafından verilir. MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMSTR

Tamamen karakterlerden oluşan ileti.

Uygulama iletisi verileri bir SBCS dizilimi (tek baytlık karakter takımı) ya da bir DBCS dizilimi (çift baytlık karakter takımı) olabilir. MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMTM

Tetikleyici mesaj.

İleti, MQTM yapısı tarafından açıklanan bir tetikleyici iletidir; bu yapının ayrıntıları için bkz. [“MQTM-Tetikleyici iletisi” sayfa 1215](#) . MQGET çağrısında GMCONV seçeneği belirtilirse, bu biçimdeki iletiler dönüştürülebilir.

FMWIH

İş bilgileri üstbilgisi.

İleti verileri, iş bilgileri üstbilgisi MQWIH ile başlar ve bunu uygulama verileri izler. Uygulama verilerinin biçim adı, MQWIH yapısındaki WIFMT alanı tarafından verilir.

FMXQH

İletim kuyruğu üstbilgisi.

İleti verileri, iletim kuyruğu üstbilgisi MQXQH ile başlar. Özgün iletideki veriler MQXQH yapısını hemen izler. Özgün ileti verilerinin biçim adı, iletim kuyruğu üstbilgisi MQXQH 'nin bir parçası olan MQMD yapısındaki MDFMT alanı tarafından verilir. Bu yapının ayrıntıları için bkz. [“IBM i üzerinde MQXQH \(İletim kuyruğu üstbilgisi\)” sayfa 1225](#) .

MDFMT FMXQH içeren iletiler için COA ve COD raporları oluşturulmaz.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın uzunluğu LNFMT tarafından verilir. Bu alanın ilk değeri FMNONE.

MDGID (24 baytlık bit dizgisi)

Grup tanıtıcısı.

Bu, fiziksel iletinin ait olduđu belirli ileti grubunu ya da mantıksal iletiyi tanımlamak için kullanılan bir bayt dizilimidir. MDGID , ileti için bölümlenmeye izin verildiğinde de kullanılır. Bu durumların tümünde, MDGID boş olmayan bir değere sahiptir ve MDMFL alanında aşağıdaki işaretlerden biri ya da daha fazlası ayarlanır:

- MFMIG
- MFLMIG
- MFSEG
- MFLSEG
- MFSEGA

Bu işaretlerin hiçbiri ayarlanmazsa, MDGID özel boş değer GINONE ' a sahiptir.

Bu alanın, aşağıdaki durumlarda MQPUT ya da MQGET çağrısındaki uygulama tarafından ayarlanması gerekmez:

- MQPUT çağrısında PMLOGO belirtildi.
- MQGET çağrısında MOGRPI belirtilmedi.

Rapor iletisi olmayan iletiler için bu çağrıları kullanmayı düşünün. Ancak, uygulama daha fazla denetim gerektiriyorsa ya da çağrı MQPUT1ise, uygulamanın MDGID ' in uygun bir değere ayarlandığından emin olması gerekir.

İleti grupları ve bölümler yalnızca grup tanıtıcısı benzersizse doğru olarak işlenebilir. Bu nedenle, uygulamalar kendi grup tanıtıcılarını oluşturmamalıdır; Bunun yerine, uygulamalar aşağıdakilerden birini yapmalıdır:

- PMLOGO belirtilirse, kuyruk yöneticisi, mantıksal iletinin grup ya da bölümündeki ilk ileti için otomatik olarak benzersiz bir grup tanıtıcısı oluşturur ve mantıksal iletinin grup ya da bölümlerinde kalan iletiler için bu grup tanıtıcısını kullanır; bu nedenle, uygulamanın özel bir işlem gerçekleştirmesi gerekmez. Bu yordamı kullanmayı düşünün.
- PMLOGO belirtilmezse, uygulama, mantıksal iletinin grubundaki ya da bölümündeki bir ileti için ilk MQPUT ya da MQPUT1 çağrısında MDGID değerini GINONE olarak ayarlayarak, kuyruk yöneticisinden grup tanıtıcısını oluşturmasını istemelidir. Bu çağrıdan çıkışta kuyruk yöneticisi tarafından döndürülen grup tanıtıcısı, mantıksal iletinin gruptaki ya da bölümlerindeki kalan iletiler için kullanılmalıdır. Bir ileti grubu bölümlenmiş iletiler içeriyorsa, gruptaki tüm kesimler ve iletiler için aynı grup tanıtıcısı kullanılmalıdır.

PMLOGO belirtilmediğinde, mantıksal ileti grupları ve bölümlerindeki iletiler herhangi bir sıraya konabilir (örneğin, ters sırada), ancak grup tanıtıcısı bu iletilerin herhangi biri için yayınlanan ilk MQPUT ya da MQPUT1 çağrısı tarafından ayrılmalıdır.

MQPUT ve MQPUT1 çağrılarında girişte, kuyruk yöneticisi PMOPTiçinde ayrıntılı olarak açıklanan değeri kullanır. MQPUT ve MQPUT1 çağrılarının çıkışında, kuyruk yöneticisi bu alanı, açılan nesne bir dağıtım listesi değil de tek bir kuyruksa, iletiyle birlikte gönderilen değere ayarlar, ancak açılan nesne bir dağıtım listesiye değiştirilmeden bırakır. İkinci durumda, uygulamanın oluşturulan grup tanıtıcılarını bilmesi gerekiyorsa, uygulamanın PRGID alanını içeren MQPMR kayıtlarını sağlaması gerekir.

MQGET çağrısına girişte, kuyruk yöneticisi Çizelge 1' de ayrıntılı olarak açıklanan değeri kullanır. MQGET çağrısının çıkışında, kuyruk yöneticisi bu alanı alınan iletinin değerine ayarlar.

Aşağıdaki özel değer tanımlandı:

GINONE

Grup tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır. Bu, mantıksal ileti kesimleri değil, gruplarda olmayan ve bölümlenmeye izin verilmeyen iletiler için kullanılan değerdir.

Bu alanın uzunluğu LNGID tarafından verilir. Bu alanın ilk değeri GINONE. MDVER MDVER2değerinden küçükse bu alan yoksayılır.

MDMFL (10 basamaklı işaretli tamsayı)

İleti işaretleri.

Bunlar, iletinin özniteliklerini belirleyen ya da işlenmesini denetleyen işaretlerdir. İşaretler aşağıdaki kategorilere ayrılır:

- Bölümlenme işareti
- Durum işaretleri

Bunlar da sırayla anlatılıyor.

Bölümlenme işareti: Bir ileti kuyruk için çok büyük olduğunda, iletiyi kuyruğa koyma girişimi genellikle başarısız olur. Bölümlenme, kuyruk yöneticisinin ya da uygulamanın iletiyi segmentler adı verilen daha küçük parçalara ayırdığı ve her kesimi ayrı bir fiziksel ileti olarak kuyruğa yerleştirdiği bir tekniktir. İletiyi alan uygulama, bölümleri birer birer alabilir ya da kuyruk yöneticisinden bölümleri MQGET çağrısıyla döndürülen tek bir iletide yeniden birleştirmesini isteyebilir. İkincisi, MQGET çağrısında GMCMPM seçeneği belirtilerek ve iletinin tamamını barındıracak kadar büyük bir arabellek sağlanarak elde edilir. (GMCMPM seçeneğine ilişkin ayrıntılar için bkz. "[IBM i üzerinde MQGMO \(Get-message seçenekleri\)](#)" sayfa 1060 .) Bir iletinin bölümlenmesi gönderen kuyruk yöneticisinde, bir ara kuyruk yöneticisinde ya da hedef kuyruk yöneticisinde gerçekleşebilir.

Bir iletinin bölümlenmesini denetlemek için aşağıdakilerden birini belirtebilirsiniz:

MFSEGI

Segmentasyon engellenmiş.

Bu seçenek, iletinin kuyruk yöneticisi tarafından bölümlere ayrılmasını önler. Zaten bir bölüm olan bir ileti için belirtilirse, bu seçenek kesimin daha küçük bölümlere bölünmesini önler.

Bu işaretin değeri ikili sıfır. Bu varsayılandır.

MFSEGA

Bölümlenmeye izin verilir.

Bu seçenek, iletinin kuyruk yöneticisi tarafından bölümlere ayrılmasını sağlar. Zaten bir bölüm olan bir ileti için belirtilirse, bu seçenek kesimin daha küçük bölümlere bölünmesine izin verir. MFSEGA, MFSEG ya da MFLSEG ayarlanmadan ayarlanabilir.

Kuyruk yöneticisi bir iletiyi kestiğinde, kuyruk yöneticisi her kesimle birlikte gönderilen MQMD 'nin kopyasında MFSEG işaretini açar, ancak MQPUT ya da MQPUT1 çağrısında uygulama tarafından sağlanan MQMD' de bu işaretlerin ayarlarını değiştirmez. Mantıksal iletideki son bölüm için kuyruk yöneticisi, bölümlerle birlikte gönderilen MQMD ' de MFLSEG işaretini de açar.

Not: Mesajlar MFSEGA ile konduğunda ancak PMLOGO olmadan bakım gereklidir. İleti:

- Bir bölüm değil, ve
- Bir grupta değil, ve
- İftira edilmiyor.

Her bir MQPUT ya da MQPUT1 çağrısının her bir iletiye ilişkin kuyruk yöneticisi tarafından benzersiz bir grup tanıtıcısı oluşturulmasını sağlamak için, uygulamanın MDGID alanını GINONE olarak ilk durumuna getirmeyi unutmayın. Bu yapılmazsa, ilgisiz iletiler yanlışlıkla aynı grup tanıtıcısıyla sonuçlanabilir; bu da daha sonra yanlış işlenmesine neden olabilir. MDGID alanının ne zaman sıfırlanması gerektiğine ilişkin ek bilgi için MDGID alanı ve PMLOGO seçeneği açıklamalarına bakın.

Kuyruk yöneticisi, kesimlerin (artı gerekli olabilecek üstbilgi verilerinin) kuyruğa sığmasını sağlamak için iletileri gerektiği şekilde bölümlere ayırır. Ancak, kuyruk yöneticisi tarafından oluşturulan bir kesimin büyüklüğü için alt sınır vardır ve yalnızca bir iletiden yaratılan son kesim bu sınırdan daha küçük olabilir. (Uygulama tarafından oluşturulan bir kesimin boyutu için alt sınır bir bayttır.) Kuyruk yöneticisi tarafından oluşturulan bölümler eşit uzunlukta olmayabilir. Kuyruk yöneticisi iletiyi aşağıdaki gibi işler:

- Kullanıcı tanımlı biçimler, 16 baytın katları olan sınırlarda bölünür. Bu, kuyruk yöneticisinin 16 byte 'tan (son bölüm dışında) küçük kesimler oluşturmadığı anlamına gelir.
- FMSTR dışındaki yerleşik biçimler, mevcut verilerin niteliğine uygun noktalarda bölünür. Ancak, kuyruk yöneticisi hiçbir zaman bir MQ üstbilgi yapısının ortasında bir iletiyi bölmez. Bu, tek bir MQ üstbilgi yapısını içeren bir kesimin kuyruk yöneticisi tarafından daha fazla bölünemeyeceği ve sonuç olarak, bu ileti için olası kesim büyüklüğü alt sınırının 16 byte 'tan fazla olduğu anlamına gelir.

Kuyruk yöneticisi tarafından oluşturulan ikinci ya da sonraki bölüm aşağıdakilerden biriyle başlar:

- MQ üstbilgi yapısı
- Uygulama ileti verilerinin başlangıcı
- Uygulama ileti verileri arasında yarı yol
- FMSTR, varolan verilerin niteliğine bakılmaksızın bölünür (SBCS, DBCS ya da karma SBCS/ DBCS). Dizilim DBCS ya da karışık SBCS/DBCS olduğunda, bir karakter kümesinden diğerine dönüştürülemeyen kesimler oluşabilir. Kuyruk yöneticisi, FMSTR iletilerini hiçbir zaman 16 bayttan (son bölüm dışında) küçük bölümlere ayırmaz.
- Her kesimin MQMD ' deki MDFMT, MDCSive MDENC alanları, kuyruk yöneticisi tarafından kesimin başında var olan verileri doğru olarak tanımlamak üzere ayarlanır; biçim adı yerleşik bir biçimin adı ya da kullanıcı tanımlı bir biçimin adı olur.
- MDOFF değeri sıfırdan büyük olan bölümlerin MQMD ' deki MDREP alanı aşağıdaki gibi değiştirilir:
 - Her rapor tipi için, rapor seçeneği RO* D ise, ancak kesim kullanıcı verilerinin ilk 100 bayttan herhangi birini (yani, mevcut olabilecek herhangi bir MQ üstbilgi yapısını izleyen veriler) içeremezse, rapor seçeneği RO* olarak değiştirilir.

Kuyruk yöneticisi önceki kuralları izler, ancak aksi takdirde iletileri beklenmedik şekilde böler; bir iletinin nerede bölündüğü hakkında varsayımda bulunmayın

Kalıcı iletiler için, kuyruk yöneticisi yalnızca bir iş birimi içinde bölümlenme gerçekleştirebilir:

- MQPUT ya da MQPUT1 çağrısı kullanıcı tanımlı bir iş birimi içinde çalışıyorsa, o iş birimi kullanılır. Çağrı, bölümlenme işlemi boyunca partway başarısız olursa, kuyruk yöneticisi, başarısız olan çağrı nedeniyle kuyruğa yerleştirilen bölümleri kaldırır. Ancak, başarısızlık iş biriminin başarıyla kesinleştirilmesini engellemez.
- Çağrı kullanıcı tanımlı bir iş birimi dışında çalışıyorsa ve kullanıcı tanımlı bir iş birimi yoksa, kuyruk yöneticisi çağrı süresi boyunca bir iş birimi yaratır. Çağrı başarılı olursa, kuyruk yöneticisi iş birimini otomatik olarak kesinleştirir (uygulamanın bunu yapması gerekmez). Çağrı başarısız olursa, kuyruk yöneticisi iş birimini geri alır.
- Çağrı kullanıcı tanımlı bir iş biriminin dışında çalışıyorsa, ancak kullanıcı tanımlı bir iş birimi varsa, kuyruk yöneticisi bölümlenme gerçekleştiremez. İleti bölümlenme gerektirmiyorsa, çağrı yine de başarılı olabilir. Ancak ileti bölümlenme gerektiriyorsa, çağrı RC2255neden koduyla başarısız olur.

Kalıcı olmayan iletiler için kuyruk yöneticisi, bölümlenme gerçekleştirmek için bir iş biriminin kullanılabilir olmasını gerektirmez.

Kesimlere bölünebilecek iletilerin veri dönüşümüne özel önem verilmelidir:

- Veri dönüştürme işlemi yalnızca MQGET çağrısında alan uygulama tarafından gerçekleştirilirse ve uygulama GMCMPM seçeneğini belirtirse, veri dönüştürme çıkışına çıkışın tam ileti geçirilir ve iletinin bölümlenmiş olması çıkışa görünür olmaz.
- Alan uygulama bir kerede bir bölüm alırsa, veri dönüştürme çıkışı bir kerede bir bölümü dönüştürmek için çağrılır. Bu nedenle çıkış, diğer bölümlerdeki verilerden bağımsız olarak bir kesimdeki verileri dönüştürebilmelidir.

İletideki verilerin niteliği, 16 baytlık sınırlardaki verilerin isteğe bağlı olarak bölümlenmesi, çıkış tarafından dönüştürülemeyen bölümlerle ya da biçim FMSTR ve karakter kümesi DBCS ya da karışık SBCS/DBCS ise, gönderen uygulamanın daha fazla bölümlenmeyi engellemek için MFSEGI

belirterek bölümleri yaratması ve yerleştirmesi gerekir. Bu şekilde, gönderen uygulama, her bir kesimin veri dönüştürme çıkışının kesimi başarıyla dönüştürmesini sağlamak için yeterli bilgi içerdiğinden emin olabilir.

- Gönderen ileti kanalı aracısı (MCA) için gönderen dönüştürmesi belirtilirse, MCA yalnızca mantıksal ileti kesimleri olmayan iletileri dönüştürür; MCA hiçbir zaman bölüm olan iletileri dönüştürmeye çalışmaz.

Bu işaret, MQPUT ve MQPUT1 çağrılarında bir giriş işareti ve MQGET çağrısında bir çıkış işaretiyle gösterilir. İkinci çağrıda kuyruk yöneticisi, işaretin değerini MQGMO 'daki GMSEG alanına yansıtır.

Bu işaretin ilk değeri MFSEGI 'dir.

Durum işaretleri: Fiziksel iletinin bir ileti grubuna ait olup olmadığını, mantıksal iletinin bir parçası olup olmadığını gösteren işaretler. MQPUT ya da MQPUT1 çağrısında aşağıdakilerden biri ya da daha fazlası belirtilebilir ya da MQGET çağrısı tarafından döndürülebilir:

MFMI

İleti bir grubun üyesi.

MFLMI

İleti, bir gruptaki son mantıksal iletidir.

Bu işaret ayarlanırsa, kuyruk yöneticisi iletiyle birlikte gönderilen MQMD kopyasında MFMI 'yi açar, ancak MQPUT ya da MQPUT1 çağrısında uygulama tarafından sağlanan MQMD' de bu işaretlerin ayarlarını değiştirmez.

Bir grubun tek bir mantıksal iletiden oluşması geçerlidir. Bu durumda, MFLMI ayarlanır, ancak MDSEQ alanında bir değeri olur.

MFSE

İleti, mantıksal iletinin bir bölümüdür.

MFSE, MFLSE olmadan belirlendiğinde, kesimdeki uygulama iletisi verilerinin uzunluğu (var olabilecek MQ üstbilgi yapılarının uzunlukları dışında) en az bir olmalıdır. Uzunluk sıfır ise, MQPUT ya da MQPUT1 çağrısı RC2253neden koduyla başarısız olur.

MFLSE

İleti, mantıksal iletinin son bölümüdür.

Bu işaret ayarlanırsa, kuyruk yöneticisi iletiyle birlikte gönderilen MQMD kopyasında MFSE 'yi açar, ancak MQPUT ya da MQPUT1 çağrısında uygulama tarafından sağlanan MQMD' de bu işaretlerin ayarlarını değiştirmez.

Bu, mantıksal bir iletinin tek bir bölümden oluşması için geçerlidir. Bu durumda, MFLSE ayarlanır, ancak MDOFF alanı sıfır değerine sahiptir.

MFLSE belirlendiğinde, kesimdeki uygulama iletisi verilerinin uzunluğunun (var olabilecek üstbilgi yapılarının uzunlukları dışında) sıfır olması için izin verilir.

Uygulama, iletiler yerleştirilirken bu işaretlerin doğru ayarlandığından emin olmalıdır. PMLOGO belirtilirse ya da kuyruk tanıtıcısı için önceki MQPUT çağrısında belirtildiyse, işaretlerin ayarları kuyruk tanıtıcısı için kuyruk yöneticisi tarafından tutulan grup ve bölüm bilgileriyle tutarlı olmalıdır. PMLOGO belirtildiğinde kuyruk tanıtıcısı için ardışık MQPUT çağrıları için aşağıdaki koşullar geçerlidir:

- Geçerli bir grup ya da mantıksal ileti yoksa, tüm bu işaretler (ve bunların birleşimleri) geçerlidir.
- MFMI belirlendikten sonra, MFLMI belirleninceye kadar açık kalmalıdır. Bu koşul karşılanmazsa, çağrı RC2241 neden koduyla başarısız olur.
- MFSE belirlendikten sonra, MFLSE belirleninceye kadar açık kalmalıdır. Bu koşul karşılanmazsa, çağrı RC2242 neden koduyla başarısız olur.
- MFMI olmadan MFSE belirlendikten sonra, MFLSE belirleninceye kadar MFMI kapalı kalmalıdır. Bu koşul karşılanmazsa, çağrı RC2242 neden koduyla başarısız olur.

Tablo 1 , işaretlerin geçerli birleşimlerini ve çeşitli alanlar için kullanılan değerleri gösterir.

Bu işaretler, MÖPUT ve MÖPUT1 çağrılarına ilişkin giriş işaretleri ve MÖGET çağrısına ilişkin çıkış işaretleridir. İkinci çağrıda, kuyruk yöneticisi işaretlerin değerlerini MÖGMO 'daki GMGST ve GMSST alanlarına da yansıtır.

Varsayılan işaretler: İletinin varsayılan öznitelikleri olduğunu belirtmek için aşağıdaki belirtilebilir:

MFNONE

İleti işareti yok (varsayılan ileti öznitelikleri).

Bu, kesimlere ayırmayı engeller ve iletinin bir grupta olmadığını ve mantıksal iletinin bir parçası olmadığını gösterir. MFNONE, program belgelerine yardımcı olmak için tanımlanır. Bu işaretin başka bir işaretiyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

MDMFL alanı alt alanlara bölümlenir; ayrıntılar için bkz. [“IBM i üzerinde rapor seçenekleri ve ileti işaretleri” sayfa 1409.](#)

Bu alanın ilk değeri MFNONE. MDVER MDVER2değerinden küçükse bu alan yoksayılır.

MDMID (24 baytlık bit dizgisi)

İleti tanıtıcısı.

Bu, bir iletiyi diğerinden ayırmak için kullanılan bir bayt dizilimidir. Genellikle, kuyruk yöneticisi tarafından izin verilmemesine rağmen, iki iletinin aynı ileti tanıtıcısına sahip olmaması gerekir. İleti tanıtıcısı, iletinin kalıcı bir özelliğidir ve kuyruk yöneticisinin yeniden başlatması boyunca devam eder. İleti tanıtıcısı bir karakter dizgisi değil bayt dizgisi olduğundan, ileti bir kuyruk yöneticisinden diğerine aktığında ileti tanıtıcısı karakter kümeleri arasında dönüştürülmez.

MÖPUT ve MÖPUT1 çağrıları için, uygulama tarafından MINONE ya da PMNMID belirtilirse, kuyruk yöneticisi ileti konduğunda benzersiz bir ileti tanıtıcısı oluşturur ve iletiyle birlikte gönderilen ileti tanımlayıcısına yerleştirir. Kuyruk yöneticisi, gönderen uygulamaya ait ileti tanımlayıcısında bu ileti tanıtıcısını da döndürür. Uygulama, belirli iletilere ilişkin bilgileri kaydetmek ve uygulamanın diğer bölümlerinden gelen sorgulara yanıt vermek için bu değeri kullanabilir.

Kuyruk yöneticisi tarafından oluşturulan bir MDMID , 4 baytlık bir ürün tanıtıcısından (ASCII ya da EBCDIC 'de AMQ→ ya da CSQ→ ; burada → tek bir boş karakteri temsil eder) ve ardından benzersiz bir dizginin ürüne özgü somutlamasından oluşur. IBM MQ içinde bu, kuyruk yöneticisi adının ilk 12 karakterini ve sistem saatinden türetilmiş bir değeri içerir. Bu nedenle, ileti tanıtıcılarının benzersiz olduğundan emin olmak için, iletişim kurabilen tüm kuyruk yöneticilerinin ilk 12 karakterde farklı adları olmalıdır. Benzersiz bir dizgi oluşturma yeteneği, sistem saatinin geriye doğru değiştirilmemesine de bağlıdır. Kuyruk yöneticisi tarafından oluşturulan bir ileti tanıtıcısının uygulama tarafından çoğaltılması olasılığını ortadan kaldırmak için uygulama, ASCII ya da EBCDIC 'de (X'41'-X'49 've X'C1'-X'C9') A-I aralığındaki ilk karakterlerle tanıtıcı oluşturmaktan kaçınılmalıdır. Ancak, uygulamanın bu aralıklardaki ilk karakterlerle tanıtıcı oluşturmaması engellenmez.

İleti bir konuya konuluyorsa, kuyruk yöneticisi yayınlanan her ileti için gerektiği gibi benzersiz ileti tanıtıcıları oluşturur. Uygulama PMNMID değerini belirtirse, kuyruk yöneticisi çıkışta döndürmek için benzersiz bir ileti tanıtıcısı oluşturur. MINONE uygulama tarafından belirtilirse, çağrıdan döndüğünüzde MQMD 'deki MDMID alanının değeri değişmez.

Alıkonan yayınlarla ilgili daha fazla ayrıntı için [PMOPT](#) içindeki PMRET açıklamasına bakın.

İleti bir dağıtım listesine yerleştiriliyorsa, kuyruk yöneticisi gerektiği şekilde benzersiz ileti tanıtıcıları oluşturur; ancak, MINONE ya da PMNMID belirtilmiş olsa da, MQMD 'deki MDMID alanının değeri çağrıdan dönüşte değişmez. Uygulamanın kuyruk yöneticisi tarafından oluşturulan ileti tanıtıcılarını bilmesi gerekiyorsa, uygulamanın PRMID alanını içeren MQPMR kayıtlarını sağlaması gerekir.

Gönderen uygulama, ileti tanıtıcısı için MINONE dışında belirli bir değer de belirtebilir; bu, kuyruk yöneticisinin benzersiz bir ileti tanıtıcısı oluşturmamasını durdurur. Bir iletiyi ileten bir uygulama, özgün iletinin ileti tanıtıcısını yaymak için bu olanağı kullanabilir.

Kuyruk yöneticisinin kendisi aşağıdaki durumlar dışında bu alanı kullanmaz:

- Daha önce açıklandığı gibi, istenirse benzersiz bir değer oluşturun

- İletiyeye ilişkin alma isteğini veren uygulamaya değeri sağlayın
- Değeri, bu ileti hakkında oluşturduğu herhangi bir rapor iletilisinin MDCID alanına kopyalayın (MDREP seçeneklerine bağlı olarak)

Kuyruk yöneticisi ya da bir ileti kanalı aracısı bir rapor iletilisi oluşturduğunda, MDMID alanını özgün iletilinin (RONMI ya da ROPMI) MDREP alanı tarafından belirtilen şekilde ayarlar. Rapor iletileri oluşturan uygulamalar da bunu yapmalıdır.

MQGET çağrısı için MDMID , kuyruktan alınacak belirli bir iletiyi seçmek için kullanılacak beş alandan biridir. Olağan durumda MQGET çağrısı kuyruktaki sonraki iletiyi döndürür, ancak belirli bir ileti gerekiyorsa, bu, herhangi bir birleşimde beş seçim ölçütünden biri ya da daha fazlası belirtilerek elde edilebilir; bu alanlar şunlardır:

- MDMID
- MDCID
- MDGID
- MDSEQ
- MDOFF

Uygulama bu alanlardan birini ya da daha fazlasını gerekli değerlere ayarlar ve MQGMO ' nun GMMO alanındaki ilgili MO* eşleşme seçeneklerini, bu alanların seçim ölçütü olarak kullanılması gerektiğini gösterecek şekilde ayarlar. Yalnızca bu alanlarda belirtilen değerleri içeren iletiler alma adaylarıdır. GMMO alanının varsayılan değeri (uygulama tarafından değiştirilmediyse), ileti tanıtıcısı ve ilinti tanıtıcısıyla eşleşmektedir.

Olağan durumda, döndürülen ileti, kuyrukta seçim ölçütlerini karşılayan ilk iletidir. Ancak GMBRWN belirtilirse, döndürülen ileti seçim ölçütlerini karşılayan bir sonraki iletidir; bu iletiye ilişkin tarama, imlecin yürürlükteki konumundan sonra gelen iletiyle başlar.

Not: Kuyruk, seçim ölçütlerini karşılayan bir ileti için sırayla taranır; bu nedenle, alma süreleri, seçim ölçütü belirtilmemesinden daha yavaş olur; özellikle de uygun bir ileti bulunmadan önce çok sayıda ileti taranması gerekirse.

Seçim ölçütlerinin çeşitli durumlarda nasıl kullanıldığına ilişkin ek bilgi için [Tablo 1](#) başlıklı konuya bakın.

İleti tanıtıcısı olarak MINONE değerinin belirlenmesi, MOMSGI belirlenmemesiyle aynı etkiye sahiptir; diğer bir deyişle, herhangi bir ileti tanıtıcısı eşleşir.

MQGET çağrısındaki **GMO** değiştirilmesinde GMMUC seçeneği belirtilirse bu alan yoksayılr.

Bir MQGET çağrısından döndüğünüzde, MDMID alanı döndürülen iletilinin ileti tanıtıcısına (varsa) ayarlanır.

Aşağıdaki özel değer kullanılabilir:

MINON

İleti tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

Bu, MQGET, MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş/çıkış alanıdır. Bu alanın uzunluğu LNMID tarafından verilir. Bu alanın ilk değeri MINONE.

MDMT (10 basamaklı işaretli tamsayı)

İleti tipi.

Bu, iletilinin tipini gösterir. İleti tipleri aşağıdaki gibi gruplanır:

MTSFST (TSFST)

Sistem tarafından tanımlanan ileti tipleri için en düşük değer.

MTSLST

Sistem tarafından tanımlanan ileti tipleri için en yüksek değer.

Sistem aralığı içinde tanımlı olan değerler şunlardır:

MTDGRM

Yanıt gerektirmeyen ileti.

İleti, yanıt gerektirmeyen bir ileti.

MTRQST

Yanıt gerektiren ileti.

İleti, yanıt gerektiren bir iletidir.

Yanıtın gönderilmesi gereken kuyruğun adı MDRQ alanında belirtilmelidir. MDREP alanı, yanıtın MDMID ve MDCID ögesinin nasıl ayarlanacağını gösterir.

MTRPLY

Daha önceki bir istek iletisini yanıtlayın.

İleti, önceki bir istek iletisine (MTRQST) verilen yanıttır. İleti, istek iletisinin MDRQ alanı tarafından belirtilen kuyruğa gönderilmelidir. Yanıtın MDMID ve MDCID ' nin nasıl ayarlanacağını denetlemek için isteğin MDREP alanı kullanılmalıdır.

Not: Kuyruk yöneticisi, istek-yanıt ilişkisini uygulamaz; bu bir uygulama sorumluluğudur.

MTRPRT

Rapor iletisi.

İleti, genellikle başka bir iletiyle ilgili beklenen ya da beklenmeyen bir oluşumu bildiriyor (örneğin, geçerli olmayan veriler içeren bir istek iletisi alındı). İleti, özgün iletinin ileti tanımlayıcısının MDRQ alanı tarafından belirtilen kuyruğa gönderilmelidir. MDFB alanı, raporun doğasını gösterecek şekilde ayarlanmalıdır. Rapor iletisinin MDMID ve MDCID ögesinin nasıl ayarlanacağını denetlemek için özgün iletinin MDREP alanı kullanılabilir.

Kuyruk yöneticisi ya da ileti kanalı aracısı tarafından oluşturulan rapor iletileri her zaman MDRQ kuyruğuna gönderilir; MDFB ve MDCID alanları daha önce açıklandığı gibi ayarlanır.

Sistem aralığı içindeki diğer değerler MQI ' in sonraki sürümlerinde tanımlanabilir ve MQPUT ve MQPUT1 çağrılarını tarafından hatasız olarak kabul edilir.

Uygulama tanımlı değerler de kullanılabilir. Bunlar aşağıdaki aralık içinde olmalıdır:

MTAFST

Uygulama tanımlı ileti tipleri için en düşük değer.

MTALST

Uygulama tanımlı ileti tipleri için en yüksek değer.

MQPUT ve MQPUT1 çağrılarını için, MDMT değeri sistem tanımlı aralık içinde ya da uygulama tanımlı aralık içinde olmalıdır; değilse, çağrı başarısız olur; neden kodu RC2029.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarınına ilişkin bir giriş alanıdır. Bu alanın ilk değeri MTDGRM 'dir.

MDOFF (10 basamaklı işaretli tamsayı)

Mantıksal iletinin başlangıcından fiziksel iletideki verilerin görelî konumu.

Bu, fiziksel iletideki verilerin, veri formunun bir parçası olduğu mantıksal iletinin başlangıcından itibaren bayt cinsinden görelî konumudur. Bu verilere *segment*denir. Görelî konum, 0-999 999 999 999 aralığındadır. Mantıksal iletinin bir parçası olmayan fiziksel bir iletinin görelî konumu sıfır.

Bu alanın, aşağıdaki durumlarda MQPUT ya da MQGET çağrısındaki uygulama tarafından ayarlanması gerekmez:

- MQPUT çağrısında PMLOGO belirtildi.
- MQGET çağrısında MOOFFS belirtilmedi.

Bunlar, rapor iletisi olmayan iletiler için bu çağrılarını kullanmanın önerilen yollarıdır. Ancak, uygulama bu koşullara uymuyorsa ya da çağrı MQPUT1 ise, uygulamanın MDOFF ' in uygun bir değere ayarlandığından emin olması gerekir.

MQPUT ve MQPUT1 çağrılarında girişte, kuyruk yöneticisi Tablo 1 içinde ayrıntılı olarak açıklanan değeri kullanır. MQPUT ve MQPUT1 çağrılarının çıkışında, kuyruk yöneticisi bu alanı iletiyle birlikte gönderilen değere ayarlar.

Mantıksal bir iletinin bir kesimine ilişkin rapor iletisi raporlaması için, MDOLN alanı (OLUNDF değilse), kuyruk yöneticisi tarafından tutulan kesim bilgilerindeki görelî konumu güncellemek için kullanılır.

MQGET çağrısına girişte, kuyruk yöneticisi Çizelge 1 ' de ayrıntılı olarak açıklanan değeri kullanır. MQGET çağrısının çıkışında, kuyruk yöneticisi bu alanı alınan iletinin değerine ayarlar.

Bu alanın ilk değeri sıfırdır. MDVER MDVER2 değerinden küçükse bu alan yoksayılr.

MDOLN (10 basamaklı işaretli tamsayı)

Özgün iletinin uzunluğu.

Bu alan yalnızca bölüm olan rapor iletileri için geçerlidir. Rapor iletisinin ilişkili olduğu ileti bölümünün uzunluğunu belirler; kesimin bir parçasını oluşturduğu mantıksal iletinin uzunluğunu ya da rapor iletisindeki verilerin uzunluğunu belirtmez.

Not: Bölüm olan bir ileti için bir rapor iletisi oluştururken, kuyruk yöneticisi ve ileti kanalı aracısı, özgün iletideki MDGID, MDSEQ, MDOFF ve *MDMFL* alanlarını rapor iletisi için MQMD ' ye kopyalar. Sonuç olarak, rapor iletisi de bir bölümdür. Rapor iletileri oluşturan uygulamaların da aynısını yapması ve MDOLN alanının doğru şekilde ayarlandığından emin olması önerilir.

Aşağıdaki özel değer tanımlandı:

OLUNDF

İletinin özgün uzunluğu tanımlanmadı.

MDOLN , MQPUT ve MQPUT1 çağrılarındaki bir giriş alanıdır, ancak uygulama tarafından sağlanan değer yalnızca belirli durumlarda kabul edilir:

- Konan ileti bir bölümse ve aynı zamanda bir rapor iletisiyse, kuyruk yöneticisi belirtilen değeri kabul eder. Değer şöyle olmalıdır:
 - Kesim son kesim değilse sıfırdan büyük
 - Kesim son kesim ise sıfırdan küçük değil
 - İletide var olan verilerin uzunluğundan az değil

Bu koşullar karşılanmazsa, çağrı RC2252 neden koduyla başarısız olur.

- Yerleştirilmekte olan ileti bir bölümse, ancak bir rapor iletisi değilse, kuyruk yöneticisi alanı yoksayar ve bunun yerine uygulama iletisi verilerinin uzunluğunu kullanır.
- Diğer tüm durumlarda, kuyruk yöneticisi alanı yoksayar ve bunun yerine OLUNDF değerini kullanır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır.

Bu alanın ilk değeri OLUNDF 'dir. MDVER MDVER2 değerinden küçükse bu alan yoksayılr.


MDPAN (28 baytlık karakter dizgisi)

İletiyi koyan uygulamanın adı.

Bu, iletinin *kaynak bağlamının* bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için İleti bağlamı ve Bağlam bilgilerinin denetleme başlıklı konuya bakın.

MDPAN biçimi MDPAT değerine bağlıdır.

Bu alan kuyruk yöneticisi tarafından ayarlandığında (PMSETA dışındaki tüm seçenekler için), ortam tarafından belirlenen değere ayarlanır:

-  z/OS' da, kuyruk yöneticisi aşağıdakileri kullanır:

- z/OS toplu iş için, JES JOB kartındaki 8 karakterlik iş adı
- TSO için 7 karakterlik TSO kullanıcı kimliği
- CICS için, 8 karakterli applid, ardından 4 karakterli tranid
- IMS için, 8 karakterlik IMS sistem tanıtıcısı ve ardından 8 karakterlik PSB adı
- XCF için, 8 karakterlik XCF grubu adı ve ardından 16 karakterlik XCF üye adı
- Kuyruk yöneticisi tarafından oluşturulan bir ileti için, kuyruk yöneticisi adının ilk 28 karakteri
- CICS olmadan dağıtılmış kuyruğa alma için, kanal başlatıcısının 8 karakterlik iş adı ve ardından, teslim edilmeyen iletiler kuyruğuna yerleştirilen modülün 8 karakterlik adı ve ardından 8 karakterlik bir görev tanıtıcısı gelir.
- MQSeries Java dil bağ tanımlarının IBM MQ for z/OS ile işlenmesi için z/OS UNIX System Services ortamı için oluşturulan adres alanının 8 karakterlik iş adı. Bu genellikle, sonuna tek bir sayısal karakter eklenmiş bir TSO kullanıcı kimliği olur.

Ad ya da adların her biri, alanın geri kalanında olduğu gibi, sağda boşluklarla doldurulur. Birden çok ad varsa, aralarında ayırıcı yoktur.

- **Windows** PC DOS ve Windows sistemlerinde kuyruk yöneticisi şunları kullanır:
 - CICS uygulaması için CICS işlem adı
 - CICS dışı bir uygulama için, yürütülür dosyanın tam olarak nitelenmiş adının en sağındaki 28 karakter
- **IBM i** IBM i' de, kuyruk yöneticisi tam olarak nitelenmiş iş adını kullanır.
- **Linux** **AIX** AIX and Linux' da, kuyruk yöneticisi aşağıdakileri kullanır:
 - CICS uygulaması için CICS işlem adı
 - CICS dışı bir uygulama için, kuyruk yöneticisi tarafından kullanılabilirse, yürütülür dosyanın tam olarak nitelenmiş adının en sağındaki 14 karakter ve tersi durumda boşluk (örneğin, AIX üzerinde)
- VSE/ESA' da, kuyruk yöneticisi 8 karakterli applid değerini ve ardından 4 karakterli tranid değerini kullanır.

MQPUT ve MQPUT1 çağrılarını için, **PMO** değiştirilmesinde PMSETA belirtildiyse bu bir giriş/çıkış alanıdır. Alan içinde boş bir karakteri izleyen bilgiler atılır. Boş karakter ve aşağıdaki karakterler, kuyruk yöneticisi tarafından boşluklara dönüştürülür. PMSETA belirtilmezse, girişte bu alan yoksayılr ve salt çıkış alanıdır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu LNPAN tarafından verilir. Bu alanın ilk değeri 28 boş karakterdir.

MDPAT (10 basamaklı işaretli tamsayı)

İletiyi koyan uygulamanın tipi.

Bu, iletinin **kaynak bağlamının** bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı ve Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

MDPAT aşağıdaki standart tiplerden birine sahip olabilir. Kullanıcı tanımlı tipler de kullanılabilir, ancak ATUFST-ATULST aralığındaki değerlerle sınırlandırılmalıdır.

ATAIX

AIX uygulaması (ATUNIX ile aynı değer).

ATBRKR

Borsacı.

ÖZNTLKCICS

CICS hareketi.

ATCICB

CICS bridge.

ATVSE.

CICS/VSE hareketi.

ATDOS

PC DOS üzerinde IBM MQ MQI client uygulaması.

ATDQM (Kalite Yönetimi)

Dağıtılmış kuyruk yöneticisi aracı.

ATGUAR

Tandem Guardian uygulaması (ATNSK ile aynı değer).

ÖZNTLKI

IMS uygulaması.

ATIMSB

IMS köprüsü.

ATJAVA

Java.

ATMVS

MVS ya da TSO uygulaması (ATZOS ile aynı değer).

ATNOT (NOT)

Lotus Notes Agent uygulaması.

ATNSK

Tandem NonStop Kernel uygulaması.

AT390

OS/390 uygulaması (ATZOS ile aynı değer).

AT400

IBM i uygulaması.

ÖZNKY

Kuyruk yöneticisi.

ATUNIX

UNIX uygulaması.

ATVOS

Stratus VOS uygulaması.

ATWIN.

16 bit Windows uygulaması.

ATWINT (NOKTA)

32 bit Windows uygulaması.

ATXCF

-XCF.

ATZOS

z/OS uygulaması.

ÖZNTLKF

Varsayılan uygulama tipi.

Bu, uygulamanın çalıştığı altyapıya ilişkin varsayılan uygulama tipidir.

Not: Bu değişimin değeri ortama özgüdür.

ATUNK

Bilinmeyen uygulama tipi.

Bu değer, başka bağlam bilgileri olsa da uygulama tipinin bilinmediğini belirtmek için kullanılabilir.

ATUFST

Kullanıcı tanımlı uygulama tipi için en düşük değer.

ATULST

Kullanıcı tanımlı uygulama tipi için en yüksek değer.

Aşağıdaki özel değer de oluşabilir:

ATNCON

İletide bağlam bilgisi yok.

Bu değer, bağlamı olmayan bir ileti konduğunda (PMNOC bağlam seçeneği belirtilirse) kuyruk yöneticisi tarafından ayarlanır.

Bir ileti alındığında, iletinin bağlamı olup olmadığına karar vermek için MDPAT sınanabilir (diğer bağlam alanlarından herhangi biri boş değilse, PMSETA kullanan bir uygulama tarafından MDPAT değerinin ATNCON olarak ayarlanmaması önerilir).

ATSIB

Başka bir IBM MQ ileti sistemi ürününde oluşturulan ve SIB (Service Integration Bus) köprüsü aracılığıyla gelen bir iletiyi gösterir.

Kuyruk yöneticisi bu bilgileri bir uygulama koymasının sonucu olarak oluşturduğunda, alan ortam tarafından belirlenen bir değere ayarlanır.

IBM i IBM i'ta alanın AT400; olarak ayarlandığını unutmayın; kuyruk yöneticisi IBM üzerinde ATCICS' i hiçbir zaman kullanmaz.

MQPUT ve MQPUT1 çağrılar için, **PMO** değiştirgesinde PMSETA belirtildiyse bu bir giriş/çıkış alanıdır. PMSETA belirtilmezse, girişte bu alan yoksayılr ve salt çıkış alanıdır.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen MDPAT ögesini içerir. Bu değer, alıkonursa iletiyle birlikte tutulan MDPAT değeri olur (alıkonan yayınlarla ilgili daha fazla ayrıntı için PMRET açıklamasına bakın), ancak kendilerine gönderilen tüm yayınlarda MDPAT değerini geçersiz kılacak bir değer sağladıkları için ileti abonelere yayın olarak gönderildiğinde MDPAT olarak kullanılmaz. İletinin bağlamı yoksa, alan ATNCON olarak ayarlanır.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın ilk değeri ATNCON.

MDPD (8 baytlık karakter dizilimi)

İletinin konma tarihi.

Bu, iletinin *kaynak bağlamının* bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı ve Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

Bu alanın kuyruk yöneticisi tarafından oluşturulduğu tarih için kullanılan biçim:

• YYYYAAAGG

karakterlerin temsil ettiği yer:

YYYY

yıl (dört sayısal basamak)

MM

yılın ayı (01-12)

GG

ayın günü (01-31)

Greenwich Ortalama Saati (GMT), sistem saatinin GMT ' ye doğru şekilde ayarlanmasına bağlı olarak MDPD ve MDPT alanları için kullanılır.

İleti bir iş biriminin parçası olarak konduysa, tarih, iş biriminin kesinleştirildiği tarih değil, iletinin konduğu tarihtir.

MQPUT ve MQPUT1 çağrılar için, **PMO** değiştirgesinde PMSETA belirtildiyse bu bir giriş/çıkış alanıdır. Alanın içeriği kuyruk yöneticisi tarafından denetlenmez; ancak, alan içindeki boş karakterin ardından gelen bilgiler atılır. Boş karakter ve aşağıdaki karakterler, kuyruk yöneticisi tarafından boşluklara dönüştürülür. PMSETA belirtilmezse, girişte bu alan yoksayılr ve salt çıkış alanıdır.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen MDPD ögesini içerir. Bu değer, alıkonursa iletiyle birlikte tutulan MDPD değeri olur (alıkonan yayınlarla ilgili daha fazla ayrıntı için PMRET açıklamasına bakın), ancak kendilerine gönderilen tüm yayınlarda MDPD değerini geçersiz kılacak bir değer sağladıkları için ileti abonelere yayın olarak gönderildiğinde MDPD olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen boştur.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu LNPDAT tarafından verilir. Bu alanın ilk değeri 8 boş karakterdir.

MDPER (10 basamaklı işaretli tamsayı)

İleti kalıcılığı.

Bu, iletinin sistem hatalarından kurtulup kurtulmadığını ve kuyruk yöneticisinin yeniden başlatılıp başlatılmadığını gösterir. MQPUT ve MQPUT1 çağrıları için değer aşağıdakilerden biri olmalıdır:

PEPER

İleti kalıcı.

Bu, iletinin sistem hatalarından kurtulup kuyruk yöneticisini yeniden başlatması anlamına gelir. İleti konduktan ve golf sopasının iş birimi kesinleştirildikten sonra (ileti bir iş biriminin parçası olarak konursa), ileti yardımcı bellekte korunur. İleti kuyruktan kaldırılıncaya ve alıcı iş birimi kesinleştirilinceye (ileti iş biriminin bir parçası olarak alındıysa) kadar orada kalır.

Uzak kuyruğa kalıcı bir ileti gönderildiğinde, iletinin bir sonraki kuyruk yöneticisine geldiği bilininceye kadar, hedef rotadaki her kuyruk yöneticisinde iletiyi tutmak için bir saklama ve iletme mekanizması kullanılır.

Kalıcı iletilerin yerleştirilemediği yer:

- Geçici dinamik kuyruklar
- Bağlaşım olanağı yapı düzeyinin üçten az olduğu ya da bağlaşım olanağı yapısının kurtarılamadığı paylaşılan kuyruklar.

Kalıcı iletiler, bağlaşım olanağı yapı düzeyi 3 ve bağlaşım olanağının kurtarılabildiği olduğu kalıcı dinamik kuyruklara, önceden tanımlanmış kuyruklara ve paylaşılan kuyruklara yerleştirilebilir.

PENPER

İleti kalıcı değil.

Bu, iletinin olağan durumda sistem hatalarından ya da kuyruk yöneticisini yeniden başlatmadan kurtulmadığı anlamına gelir. Bu, kuyruk yöneticisi yeniden başlatılırken yardımcı bellekte iletinin bozulmamış bir kopyası bulursa da geçerlidir.

Paylaşılan kuyrukların özel durumunda, kalıcı olmayan iletiler *yapma* kuyruk paylaşım grubundaki kuyruk yöneticilerinin yeniden başlatmalarına devam ediyor, ancak paylaşılan kuyruklardaki iletileri saklamak için kullanılan bağlaşım olanağının başarısızlıklarına dayanmıyor.

PEQDEF

İletinin varsayılan kalıcılığı var.

- Kuyruk bir küme kuyruğuyorsa, iletinin kalıcılığı, iletinin yerleştirildiği kuyruğun belirli eşgörünümüne sahip hedef kuyruk yöneticisinde tanımlanan **DefPersistence** özniteliğinden alınır. Genellikle, bir küme kuyruğunun tüm eşgörünümleri **DefPersistence** özniteliği için aynı değere sahiptir, ancak bu zorunlu değildir.

DefPersistence değeri, ileti hedef kuyruğa yerleştirildiğinde **MDPER** alanına kopyalanır. **DefPersistence** daha sonra değiştirilirse, kuyruğa önceden yerleştirilen iletiler bundan etkilenmez.

- Kuyruk bir küme kuyruğu değilse, hedef kuyruk yöneticisi uzak olsa bile, iletinin kalıcılığı yerel kuyruk yöneticisinde tanımlanan **DefPersistence** özniteliğinden alınır.

Kuyruk adı çözme yolunda birden çok tanımlama varsa, varsayılan kalıcılık, yoldaki ilk tanımlamada bu özniteliğin değerinden alınır. Bu şöyle olabilir:

- Diğer ad kuyruğu

- Yerel bir kuyruk
- Uzak kuyruğun yerel tanımlaması
- Kuyruk yöneticisi diğer adı
- Bir iletim kuyruğu (örneğin, DefXmitQName kuyruğu)

İleti konduğunda, **DefPersistence** değeri MDPER alanına kopyalanır. Daha sonra **DefPersistence** değiştirilirse, önceden konmuş olan iletiler etkilenmez.

Kalıcı ve kalıcı olmayan iletiler aynı kuyrukta bulunabilir.

Bir iletiyi yanıtlarken, uygulamalar genellikle istek iletilerinin kalıcılığını yanıt iletileri için kullanmalıdır.

Bir MQGET çağrısı için döndürülen değer PEPER ya da PENPER değeridir.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri PEQDEF 'dir.

MDPRI (10 basamaklı işaretli tamsayı)

İleti önceliği.

MQPUT ve MQPUT1 çağrıları için değer sıfırdan büyük ya da sıfıra eşit olmalıdır; sıfır en düşük önceliktir. Aşağıdaki özel değer de kullanılabilir:

PRQDEF

Kuyruk için varsayılan öncelik.

- Kuyruk bir küme kuyruğuyorsa, iletinin önceliği, iletinin yerleştirildiği kuyruğun belirli eşgörünümüne sahip olan hedef kuyruk yöneticisinde tanımlandığı şekilde **DefPriority** özniteliğinden alınır. Genellikle, bir küme kuyruğunun tüm eşgörünümleri **DefPriority** özniteliği için aynı değere sahiptir, ancak bu zorunlu değildir.

DefPriority değeri, ileti hedef kuyruğa yerleştirildiğinde MDPRI alanına kopyalanır. **DefPriority** daha sonra değiştirilirse, kuyruğa önceden yerleştirilen iletiler bundan etkilenmez.

- Kuyruk bir küme kuyruğu değilse, hedef kuyruk yöneticisi uzak olsa bile, iletinin önceliği yerel kuyruk yöneticisinde tanımlandığı şekilde **DefPriority** özniteliğinden alınır.

Kuyruk adı çözüm yolunda birden çok tanımlama varsa, varsayılan öncelik, yoldaki ilk tanımlamada bu özniteliğin değerinden alınır. Bu şöyle olabilir:

- Diğer ad kuyruğu
- Yerel bir kuyruk
- Uzak kuyruğun yerel tanımlaması
- Kuyruk yöneticisi diğer adı
- Bir iletim kuyruğu (örneğin, DefXmitQName kuyruğu)

İleti konduğunda, **DefPriority** değeri MDPRI alanına kopyalanır. Daha sonra **DefPriority** değiştirilirse, önceden konmuş olan iletiler etkilenmez.

MQGET çağrısıyla döndürülen değer her zaman sıfırdan büyük ya da sıfıra eşittir; PRQDEF değeri hiçbir zaman döndürülmez.

Yerel kuyruk yöneticisi tarafından desteklenen üst sınırdan daha yüksek bir önceliğe sahip bir ileti konursa (bu üst sınır **MaxPriority** kuyruk yöneticisi özniteliği tarafından verilir), ileti kuyruk yöneticisi tarafından kabul edilir, ancak kuyruk yöneticisinin en yüksek önceliğine göre kuyruğa yerleştirilir; MQPUT ya da MQPUT1 çağrısı CCWARN ve neden kodu RC2049 ile tamamlanır. Ancak MDPRI alanı, iletiyi koyan uygulama tarafından belirtilen değeri korur.

Bir iletiyi yanıtlarken, uygulamalar normalde yanıt iletileri için istek iletilerinin önceliğini kullanmalıdır. Diğer durumlarda, PRQDEF değerinin belirlenmesi, öncelik ayarlarının uygulama değiştirilmeden gerçekleştirilmesini sağlar.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın ilk değeri PRQDEF 'dir.

MDPT (8 baytlık karakter dizgisi)

İletinin konma zamanı.

Bu, iletinin **kaynak bağlamının** bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için İleti bağlamı ve Bağlam bilgilerini denetleme başlıklı konuya bakın.

Bu alanın kuyruk yöneticisi tarafından oluşturulduğu saat için kullanılan biçim:

- HHMMSSSTH

burada karakterler (sırayla) temsil edilir:

SS

saat (00-23)

MM

dakika (00-59)

SS

saniye (00-59; bkz. not)

T

saniyenin onda biri (0-9)

H

saniyenin yüzde biri (0-9)

Not: Sistem saati çok doğru bir zaman standardıyla eşitlenirse, nadir durumlarda MDPT içindeki saniyeler için 60 ya da 61 geri gönderilebilir. Bu, artık saniyeler genel zaman standardına eklendiğinde olur.

Greenwich Ortalama Saati (GMT), sistem saatinin GMT ' ye doğru şekilde ayarlanmasına bağlı olarak MDPD ve MDPT alanları için kullanılır.

İleti bir iş biriminin parçası olarak konduysa, zaman, iş biriminin kesinleştirildiği zaman değil, iletinin konduğu zamandır.

MQPUT ve MQPUT1 çağrıları için, **PMO** değiştirilmesinde PMSETA belirtildiyse bu bir giriş/çıkış alanıdır. Alanın içeriği kuyruk yöneticisi tarafından denetlenmez; ancak, alan içindeki boş karakterin ardından gelen bilgiler atılır. Boş karakter ve aşağıdaki karakterler, kuyruk yöneticisi tarafından boşluklara dönüştürülür. PMSETA belirtilmezse, girişte bu alan yoksa ve salt çıkış alanıdır.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa konduysa, iletiyle birlikte iletilen MDPT değerini içerir. Bu değer, alıkonursa iletiyle birlikte tutulan MDPT değeri olur (alıkonan yayınlarla ilgili daha fazla ayrıntı için PMRET açıklamasına bakın), ancak kendilerine gönderilen tüm yayınlarda MDPT değerini geçersiz kılacak bir değer sağladıkları için ileti abonelere yayın olarak gönderildiğinde MDPT olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen boştur.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu LNPTIM tarafından verilir. Bu alanın ilk değeri 8 boş karakterdir.

MDREP (10 basamaklı işaretli tamsayı)

Rapor iletileri için seçenekler.

Rapor iletisi, bir uygulamayı özgün iletiyle ilgili beklenen ya da beklenmeyen olaylar hakkında bilgilendirmek için kullanılan başka bir iletiyle ilgili bir iletidir. MDREP alanı, uygulamanın özgün iletisi göndererek hangi rapor iletilerinin gerekli olduğunu, uygulama iletisi verilerinin bunlara dahil edilip edilmeyeceğini ve ayrıca (hem raporlar hem de yanıtlar için) rapor ya da yanıt iletisindeki ileti ve ilinti tanıtıcılarının nasıl ayarlanacağını belirtmesini sağlar. Aşağıdaki rapor iletisi türlerinin herhangi biri (ya da hiçbiri) istenebilir:

- Kural dışı durum
- Süre Bitim Tarihi

- Variş sırasında onayla (COA)
- Teslimatı onayla (COD)
- Pozitif işlem bildirim (PAN)
- Negatif işlem bildirim (NAN)

Birden fazla rapor iletisi tipi gerekiyorsa ya da başka rapor seçenekleri gerekiyorsa, değerler bir araya eklenebilir (aynı sabiti bir kereden fazla ekleme).

Rapor iletisini alan uygulama, MQMD 'deki MDFB alanını inceleyerek raporun oluşturulma nedenini belirleyebilir; daha fazla ayrıntı için MDFB alanına bakın.

Bir konuya ileti koyarken rapor seçeneklerinin kullanılması, sıfır, bir veya daha fazla rapor iletisinin oluşturulmasına ve uygulamaya gönderilmesine neden olabilir. Bunun nedeni, yayın iletisinin sıfır, bir ya da daha fazla abone olan uygulamaya gönderilebilir olmasıdır.

Kural dışı durum seçenekleri: Kural dışı durum raporu iletisi istemek için aşağıdaki seçeneklerden birini belirtebilirsiniz.

ROACTIVITY

Gerekli etkinlik raporları

Bu rapor seçeneği, bu rapor seçeneği ayarlanmış bir ileti destekleyen uygulamalar tarafından işlendiğinde bir etkinlik raporu oluşturulmasını sağlar.

Bu rapor seçeneği ayarlanmış olan iletiler, seçeneği 'anlamasalar' bile, herhangi bir kuyruk yöneticisi tarafından kabul edilmelidir. Bu, rapor seçeneğinin, önceki kuyruk yöneticileri tarafından işlene bile, herhangi bir kullanıcı iletisinde ayarlanmasını sağlar. Bunu gerçekleştirmek için rapor seçeneği ROAUM alt alanına yerleştirilir.

Bir süreç (kuyruk yöneticisi ya da kullanıcı süreci), ROACT kümesi olan bir ileti üzerinde bir Etkinlik gerçekleştiriyorsa, bir etkinlik raporu oluşturmayı ve yerleştirmeyi seçebilir.

Etkinlik raporu seçeneği, herhangi bir iletinin rotasının bir kuyruk yöneticisi ağı boyunca izlenmesini sağlar. Rapor seçeneği herhangi bir geçerli kullanıcı iletisinde belirtilebilir ve anında ağ üzerinden iletinin rotasını hesaplamaya başlayabilirler. İletiyi oluşturan uygulama, etkinlik raporu oluşturmayı etkinleştiremezse, kuyruk yöneticisi yöneticileri tarafından sağlanan bir API geçiş çıkışı kullanılarak etkinleştirilebilir.

Etkinlik raporları için birkaç koşul geçerlidir:

1. Ağ üzerinde etkinlik raporları oluşturabilen daha az kuyruk yöneticisi varsa, rota daha az ayrıntılı olacaktır.
2. Etkinlik raporları, alınan rotayı belirlemek için kolayca 'sipariş edilebilir' olmayabilir.
3. Etkinlik raporları, istenen hedeflerine giden bir rota bulamayabilir.

SATIR

Kural dışı durum raporları gerekli.

Bu tip bir rapor, bir ileti başka bir kuyruk yöneticisine gönderildiğinde ve ileti belirtilen hedef kuyruğa teslim edilemediğinde bir ileti kanalı aracı tarafından oluşturulabilir. Örneğin, hedef kuyruk ya da bir ara iletim kuyruğu dolu olabilir ya da ileti kuyruk için çok büyük olabilir.

Kural dışı durum raporu iletisinin oluşturulması, özgün iletinin kalıcılığına ve özgün iletinin iletildiği ileti kanalının hızına (normal ya da hızlı) bağlıdır:

- Tüm kalıcı iletiler ve normal ileti kanallarında dolaşan kalıcı olmayan iletiler için, kural dışı durum raporu yalnızca, hata koşulu için gönderen uygulama tarafından belirtilen işlem başarıyla tamamlanabiliyorsa oluşturulur. Gönderen uygulama, hata koşulu ortaya çıktığında özgün iletinin atılmasını denetlemek için aşağıdaki işlemlerden birini belirtebilir:
 - RODLQ (bu, özgün iletinin gönderilmeyen iletiler kuyruğuna yerleştirilmesine neden olur).
 - RODISC (bu, özgün iletinin atılmasına neden olur).

Gönderen uygulama tarafından belirlenen işlem başarıyla tamamlanamazsa, özgün ileti iletim kuyruğunda bırakılır ve kural dışı durum raporu iletilmesi oluşturulmaz.

- Hızlı ileti kanallarında dolaşan kalıcı olmayan iletiler için, özgün ileti iletim kuyruğundan kaldırılır ve hata koşulu için belirtilen işlem başarıyla tamamlanamasa bile kural dışı durum raporu oluşturulur. Örneğin, RODLQ belirtilirse, ancak (örneğin) kuyruk dolu olduğu için özgün ileti teslim edilmeyen ileti kuyruğuna yerleştirilemezse, kural dışı durum raporu iletilmesi oluşturulur ve özgün ileti atılır.

Normal ve hızlı ileti kanalları hakkında daha fazla bilgi için bkz. [İleti kalıcılığı](#) .

Özgün iletiyi koyan uygulama, MQPUT ya da MQPUT1 çağrısının döndürdüğü neden kodu aracılığıyla sorunu zamanuymulu olarak bildirebiliyorsa, kural dışı durum raporu oluşturulmaz.

Uygulamalar, aldığı bir iletinin işlenemeyeceğini belirtmek için kural dışı durum raporları da gönderebilir (örneğin, hesabın kredi sınırını aşmasına neden olacak bir borç işlemi olduğundan).

Özgün iletideki ileti verileri rapor iletilmesine dahil edilmez.

Birden çok ROEXC, ROEXCD ve ROEXCF belirtmeyin.

ROEXCD

Veri gerektiren kural dışı durum raporları.

Bu, ROEXC ile aynıdır, ancak özgün iletideki uygulama iletilmesi verilerinin ilk 100 baytı rapor iletilmesine eklenir. Özgün ileti bir ya da daha fazla MQ üstbilgi yapısı içeriyorsa, bunlar 100 baytlık uygulama verilerine ek olarak rapor iletilmesine eklenir.

Birden çok ROEXC, ROEXCD ve ROEXCF belirtmeyin.

ROEXCF

Tam veri gerektiren kural dışı durum raporları.

Bu, özgün iletideki tüm uygulama iletilmesi verilerinin rapor iletilmesine eklenmesi dışında, ROEXC ile aynıdır.

Birden çok ROEXC, ROEXCD ve ROEXCF belirtmeyin.

Süre sonu seçenekleri: Bir süre bitimi rapor iletilmesi istemek için aşağıdaki seçeneklerden birini belirtebilirsiniz.

SATIR

Süre bitimi raporları gerekli.

Bu rapor tipi, ileti bir uygulamaya teslim edilmeden önce atılırsa kuyruk yöneticisi tarafından oluşturulur çünkü süre bitimi süresi geçmiştir (bkz. MDEXP alanı). Bu seçenek belirlenmezse, bu nedenle bir ileti atılırsa (ROEXC* seçeneklerinden biri belirtilse bile) rapor iletilmesi oluşturulmaz.

Özgün iletideki ileti verileri rapor iletilmesine dahil edilmez.

Birden çok ROEXP, ROEXPD ve ROEXPF belirtmeyin.

YADIR

Veri gerektiren süre bitimi raporları.

Bu, ROEXP ile aynıdır; ancak, özgün iletideki uygulama iletilmesi verilerinin ilk 100 baytı rapor iletilmesine eklenir. Özgün ileti bir ya da daha fazla MQ üstbilgi yapısı içeriyorsa, bunlar 100 baytlık uygulama verilerine ek olarak rapor iletilmesine eklenir.

Birden çok ROEXP, ROEXPD ve ROEXPF belirtmeyin.

ROEXPF

Tam veri gerektiren süre bitimi raporları.

Bu, özgün iletideki tüm uygulama iletilmesi verilerinin rapor iletilmesine dahil edilmesi dışında, ROEXP ile aynıdır.

Birden çok ROEXP, ROEXPD ve ROEXPF belirtmeyin.

Variş sırasında doğrulama seçenekleri: Bir variş sırasında doğrulama raporu iletisi istemek için aşağıdaki seçeneklerden birini belirtebilirsiniz.

ROCOA

Variş sırasında raporların onaylanması gerekiyor.

Bu tip bir rapor, ileti hedef kuyruğa yerleştirildiğinde, hedef kuyruğun sahibi olan kuyruk yöneticisi tarafından oluşturulur. Özgün iletideki ileti verileri rapor iletisine dahil edilmez.

İleti bir iş biriminin parçası olarak konursa ve hedef kuyruk yerel bir kuyruksa, kuyruk yöneticisi tarafından oluşturulan COA rapor iletisi yalnızca iş birimi kesinleştirildiğinde ve kesinleştirildiğinde alınabilir.

İleti tanımlayıcısındaki MDFMT alanı FMXQH ya da FMDLH ise bir COA raporu oluşturulmaz. Bu, ileti bir iletim kuyruğuna konursa ya da teslim edilemezse ve bir teslim edilmeyen ileti kuyruğuna konursa bir COA raporunun oluşturulmasını önler.

Birden çok ROCOA, ROCOAD ve ROCOAF belirtmeyin.

ROCOAD

Gerekli verilerle gelen raporları onaylayın.

Bu, ROCOA ile aynıdır; ancak, özgün iletideki uygulama iletisi verilerinin ilk 100 baytı rapor iletisine eklenir. Özgün ileti bir ya da daha fazla MQ üstbilgi yapısı içeriyorsa, bunlar 100 baytlık uygulama verilerine ek olarak rapor iletisine eklenir.

Birden çok ROCOA, ROCOAD ve ROCOAF belirtmeyin.

ROCOAF

Tam veri gerekli, variş sırasında raporları onaylayın.

Bu, özgün iletideki tüm uygulama iletisi verilerinin rapor iletisine dahil edilmesi dışında ROCOA ile aynıdır.

Birden çok ROCOA, ROCOAD ve ROCOAF belirtmeyin.

At ve süre bitimi seçeneklerini at: Rapor iletilerine ilişkin süre bitim zamanını ayarlamak ve işareti atmak için aşağıdaki seçeneği belirtebilirsiniz.

ROPDAE (BAE)

Rapor iletisi süre bitim zamanını ayarlayın ve işareti atın.

Bu seçenek, rapor iletilerinin ve yanıt iletilerinin süre bitimini devralmasını ve işareti (atılıp atılmayacağını) özgün iletilerinden atmasını sağlar. Bu seçenek ayarlandığında, iletileri raporla ve yanıtla:

1. RODISC işaretini (ayarlandıysa) edin.
2. İleti bir süre bitimi raporu değilse, iletinin kalan süre bitimini devralır. İleti bir süre bitimi raporuyorsa, süre bitimi 60 saniye olarak ayarlanır.

Bu seçenek belirlendiğinde, aşağıdakiler geçerli olur:

Not:

1. Rapor ve yanıt iletileri, bir atma işareti ve süre bitimi değeriyle oluşturulur ve sistem içinde kalamaz.
2. İzleme rotası iletilerinin, izleme rotası etkin olmayan kuyruk yöneticilerindeki hedef kuyruklara ulaşması engellenir.
3. İletişim bağlantıları kesilirse, kuyrukların teslim edilemeyen raporlarla doldurulması önlenir.
4. Komut sunucusu yanıtları, isteğin kalan süre bitimini devralır.

Teslimde doğrulama seçenekleri: Teslimde doğrulama raporu iletisi istemek için aşağıdaki seçeneklerden birini belirleyebilirsiniz.

ROCOD

Teslimatta onaylanması gereken raporlar var.

Bu tip bir rapor, bir uygulama iletiyi hedef kuyruktan, iletinin kuyruktan silinmesine neden olacak şekilde aldığı kuyruk yöneticisi tarafından oluşturulur. Özgün iletideki ileti verileri rapor iletisine dahil edilmez.

İleti bir iş biriminin parçası olarak alınır, rapor iletisi aynı iş birimi içinde oluşturulur; böylece, iş birimi kesinleştirilinceye kadar rapor kullanılamaz. İş birimi geri çekilirse, rapor gönderilmez.

İleti tanımlayıcıdaki MDFMT alanı FMDLH ise COD raporu oluşturulmaz. Bu, ileti teslim edilemezse ve teslim edilmeyen bir kuyruğa konursa bir COD raporunun oluşturulmasını önler.

Hedef kuyruk bir XCF kuyruğuysa ROCOD geçerli değildir.

Birden çok ROCOD, ROCODD ve ROCODF belirtmeyin.

ROCODD

Teslim edilme raporlarını gerekli verilerle onaylayın.

Bu, ROCOD ile aynıdır; ancak, özgün iletideki uygulama iletisi verilerinin ilk 100 baytı rapor iletisine eklenir. Özgün ileti bir ya da daha fazla MQ üstbilgi yapısı içeriyorsa, bunlar 100 baytlık uygulama verilerine ek olarak rapor iletisine eklenir.

Özgün ileti için MQGET çağrısında GMATM belirtilirse ve alınan ileti kesilirse, rapor iletisine yerleştirilen uygulama iletisi veri miktarı alt sınırı:

- Özgün iletinin uzunluğu
- 100 bayt.

Hedef kuyruk bir XCF kuyruğuysa ROCODD geçerli değildir.

Birden çok ROCOD, ROCODD ve ROCODF belirtmeyin.

ROCODF

Teslim edilmeye ilişkin raporları tam veriler gerekli olarak onaylayın.

Bu, ROCOD ile aynıdır, ancak özgün iletideki tüm uygulama iletisi verileri rapor iletisine eklenir.

Hedef kuyruk bir XCF kuyruğuysa ROCODF geçerli değildir.

Birden çok ROCOD, ROCODD ve ROCODF belirtmeyin.

Eylem-bildirim seçenekleri: Ayan uygulamanın pozitif eylem ya da negatif eylem rapor iletisi göndermesini istemek için aşağıdaki seçeneklerden birini ya da her ikisini belirtebilirsiniz.

ROPAN

Pozitif eylem bildirim raporları gerekli.

Bu tip bir rapor, iletiyi alan ve üzerinde işlem yapan uygulama tarafından oluşturulur. İletide istenen işlemin başarıyla gerçekleştirildiğini gösterir. Raporu oluşturan uygulama, herhangi bir verinin rapora dahil edilip edilmeyeceğini belirler.

Bu isteğin iletiyi alan uygulamaya iletilmesi dışında, kuyruk yöneticisi bu seçeneğe dayalı olarak herhangi bir işlem gerçekleştirmez. Uygunsa, raporu oluşturmak, alma uygulamasının sorumluluğundadır.

RONAN

Negatif eylem bildirim raporları gerekli.

Bu tip bir rapor, iletiyi alan ve üzerinde işlem yapan uygulama tarafından oluşturulur. İletide istenen işlemin başarıyla gerçekleştirilmediğini gösterir. Raporu oluşturan uygulama, herhangi bir verinin rapora dahil edilip edilmeyeceğini belirler. Örneğin, isteğin neden gerçekleştirilemediğini gösteren bazı verilerin eklenmesi istenebilir.

Bu isteğin iletiyi alan uygulamaya iletilmesi dışında, kuyruk yöneticisi bu seçeneğe dayalı olarak herhangi bir işlem gerçekleştirmez. Uygunsa, raporu oluşturmak, alma uygulamasının sorumluluğundadır.

Hangi koşulların olumlu bir eylemin karşılığı olduğunu ve negatif bir eylemin karşılığı olduğunu belirlemek, uygulamanın sorumluluğudur. Ancak, istek yalnızca kısmen gerçekleştirildiyse, istenirse

PAN raporu yerine bir NAN raporu oluşturulması önerilir. Ayrıca, her olası koşulun ya pozitif bir işleme ya da negatif bir işleme karşılık gelmesi önerilir, ancak her ikisine birden karşılık gelmemesi önerilir.

İleti tanıtıcısı seçenekleri: Rapor iletisinin (ya da yanıt iletisinin) MDMID ögesinin nasıl ayarlanacağını denetlemek için aşağıdaki seçeneklerden birini belirleyebilirsiniz.

RONMI

Yeni ileti tanıtıcısı.

Bu varsayılan eylemdir ve bu iletinin sonucu olarak bir rapor ya da yanıt oluşturulursa, rapor ya da yanıt iletisi için yeni bir MDMID oluşturulacağını belirtir.

ROPMI

İleti tanıtıcısını iletir.

Bu iletinin sonucu olarak bir rapor ya da yanıt üretilirse, bu iletinin MDMID raporu, raporun ya da yanıt iletisinin MDMID dosyasına kopyalanır.

Bir yayın iletisinin MsgId , yayının bir kopyasını alan her abone için farklı olacaktır ve bu nedenle rapora ya da yanıt iletisine kopyalanan MsgId her biri için farklı olacaktır.

Bu seçenek belirtilmezse, RONMI varsayılır.

İlinti tanıtıcısı seçenekleri: Rapor iletisinin (ya da yanıt iletisinin) MDCID ögesinin nasıl ayarlanacağını denetlemek için aşağıdaki seçeneklerden birini belirtebilirsiniz.

ROCMTC

İleti tanıtıcısını ilinti tanıtıcısına kopyalayın.

Bu varsayılan eylemdir ve bu iletinin sonucu olarak bir rapor ya da yanıt oluşturulursa, bu iletinin MDMID ögesinin raporun ya da yanıt iletisinin MDCID dosyasına kopyalanacağını belirtir.

Bir yayın iletisinin MsgId ' i, yayının bir kopyasını alan her abone için farklı olacaktır ve bu nedenle MsgId raporun ya da yanıt iletisinin CorrelId içine kopyalanması her biri için farklı olacaktır.

ROPCI

İlinti tanıtıcısını geçirin.

Bu iletinin sonucu olarak bir rapor ya da yanıt üretilirse, bu iletinin MDCID raporu, raporun ya da yanıt iletisinin MDCID dosyasına kopyalanır.

Bir yayın iletisinin MDCID ögesi, SOSCİD seçeneğini kullanmadığı ve MQSD ' deki SCDIC alanını CINONE olarak ayarlamadığı sürece bir aboneye özgüdür. Bu nedenle, raporun ya da yanıt iletisinin MDCID MDCID içine kopyalanması her biri için farklı olabilir.

Bu seçenek belirtilmezse, ROCMTC varsayılır.

Özgün iletide ROPMI ya da ROPCI seçeneklerinin ayarlanıp ayarlanmadığını denetlemek için isteklere yanıt ya da rapor iletileri oluşturan sunucular önerilir. Varsa, sunucular bu seçenekler için açıklanan işlemi gerçekleştirmelidir. İkisi de ayarlanmazsa, sunucular ilgili varsayılan işlemi gerçekleştirmelidir.

: Özgün ileti hedef kuyruğa teslim edilemediğinde iletinin yok edilmesini denetlemek için aşağıdaki seçeneklerden birini belirleyebilirsiniz. Bu seçenekler, yalnızca gönderen uygulama tarafından bir kural dışı durum raporu iletisi oluşturulmasına neden olacak durumlar için geçerlidir. Uygulama, kural dışı durum raporları istemesinden bağımsız olarak yok etme seçeneklerini ayarlayabilir.

RODLQ

İletiyi gitmeyen iletiler kuyruğuna yerleştirin.

Bu, varsayılan işlemidir ve ileti hedef kuyruğa teslim edilemezse, iletinin gitmeyen iletiler kuyruğuna yerleştirilmesi gerektiğini belirtir. Bu durum aşağıdaki durumlarda oluşur:

- Özgün iletiyi koyan uygulamaya, MQPUT ya da MQPUT1 çağrısının döndürdüğü neden koduyla sorunu zamanuyumlu olarak bildirilemezse. Gönderen tarafından istendiye, bir kural dışı durum raporu iletisi oluşturulur.
- Özgün iletiyi koyan uygulama bir konuya yerleştirildiğinde

Gönderen tarafından istendiye, bir kural dışı durum raporu iletisi oluşturulur.

RODISC

İletiyi at.

Bu, iletinin hedef kuyruğa teslim edilememesi durumunda atılması gerektiğini gösterir. Bu durum aşağıdaki durumlarda oluşur:

- Özgün iletiyi koyan uygulamaya, MQPUT ya da MQPUT1 çağrısının döndürdüğü neden koduyla sorunu zamanuyumlu olarak bildirilemezse. Gönderen tarafından istendiye, bir kural dışı durum rapor iletisi oluşturulur.
- Özgün iletiyi koyan uygulama bir konuya yerleştirildiğinde

Gönderen tarafından istendiye, bir kural dışı durum rapor iletisi oluşturulur.

Özgün iletinin gönderene geri döndürülmesi gerekiyorsa, özgün ileti gitmeyen iletiler kuyruğuna yerleştirilmeden, gönderenin ROEXCF ile RODISC belirtmesi gerekir.

Varsayılan seçenek: Rapor seçeneği gerekmiyorsa aşağıdakileri belirtebilirsiniz:

YAYINI

Rapor gerekmez.

Bu değer, başka seçenek belirtilmediğini belirtmek için kullanılabilir. RONONE, program belgelerine yardımcı olmak için tanımlanır. Bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Genel bilgiler:

1. Gerekli tüm rapor tipleri, özgün iletiyi gönderen uygulama tarafından özellikle istenmelidir. Örneğin, bir COA raporu istenir, ancak bir kural dışı durum raporu istenmezse, ileti hedef kuyruğa yerleştirildiğinde bir COA raporu oluşturulur, ancak ileti oraya geldiğinde hedef kuyruk doluyorsa kural dışı durum raporu oluşturulmaz. MDREP seçeneği ayarlanmazsa, kuyruk yöneticisi ya da ileti kanalı aracısı (MCA) tarafından rapor iletisi oluşturulmaz.

Yerel kuyruk yöneticisi bunları tanımasa da bazı rapor seçenekleri belirtilebilir; bu, seçenek hedef kuyruk yöneticisi tarafından işlendiğinde kullanışlıdır. Daha fazla ayrıntı için bkz. [“IBM i üzerinde rapor seçenekleri ve ileti işaretleri” sayfa 1409](#) .

Bir rapor iletisi istenirse, raporun gönderilmesi gereken kuyruğun adı MDRQ alanında belirtilmelidir. Bir rapor iletisi alındığında, raporun türü, ileti tanımlayıcısındaki MDFB alanı incelenerek belirlenebilir.

2. Bir rapor iletisi oluşturan kuyruk yöneticisi ya da MCA, rapor iletisini yanıt kuyruğuna koyamazsa (örneğin, yanıt kuyruğu ya da iletim kuyruğu dolu olduğu için), rapor iletisi teslim edilmeyen ileti kuyruğuna yerleştirilir. Bu da başarısız olursa ya da gitmeyen iletiler kuyruğu yoksa, yapılan işlem rapor iletisinin tipine bağlıdır:

- Rapor iletisi bir kural dışı durum raporuysa, kural dışı durum raporunun oluşturulmasına neden olan ileti iletim kuyruğunda bırakılır; bu, iletinin kaybolmamasını sağlar.
- Diğer tüm rapor tipleri için rapor iletisi atılır ve işleme normal şekilde devam eder. Bu, özgün iletinin güvenli bir şekilde teslim edilmesi (COA veya COD rapor iletileri için) ya da artık herhangi bir ilgi alanı olmaması (bir süre sonu rapor iletisi için) nedeniyle yapılır.

Bir rapor iletisi bir kuyruğa başarıyla yerleştirildiğinde (hedef kuyruk ya da bir ara iletim kuyruğu), ileti artık özel işleme konusu olmaz; ileti, diğer herhangi bir ileti gibi işlenir.

3. Rapor oluşturulduğunda, MDRQ kuyruğu açılır ve aşağıdaki durumlar dışında, iletiye ilişkin MQMD ' de MDUID yetkisi kullanılarak rapor iletisi konarak rapora neden olur:

- Alan MCA tarafından oluşturulan kural dışı durum raporları, MCA ' nın rapora neden olan iletiyi yerleştirmeye çalışırken kullandığı herhangi bir yetkiyle birlikte kullanılır. CDPa kanal özniteliği, kullanılan kullanıcı kimliğini belirler.
- Kuyruk yöneticisi tarafından oluşturulan COA raporları, rapora neden olan ileti, raporu oluşturan kuyruk yöneticisine yerleştirildiğinde kullanılan herhangi bir yetkiyle yerleştirilir. Örneğin, ileti MCA 'nın kullanıcı kimliği kullanılarak bir alıcı MCA tarafından konduysa, kuyruk yöneticisi COA raporunu MCA' nın kullanıcı kimliğini kullanarak koyar.

Rapor oluşturan uygulamalar normalde bir yanıt oluşturmak için kullandıklarıyla aynı yetkiyi kullanmalıdır; bu normalde özgün iletideki kullanıcı kimliğinin yetkisi olmalıdır.

Raporun uzak bir hedefe gitmesi gerekirse, gönderenler ve alıcılar, diğer iletilerde olduğu gibi raporun kabul edilip edilmeyeceğine karar verebilir.

4. Veri içeren bir rapor iletisi istenirse:

- Rapor iletisi, her zaman özgün iletiyi gönderen tarafından istenen veri miktarıyla oluşturulur. Rapor iletisi yanıt kuyruğu için çok büyükse, açıklanan işlem önceden gerçekleşir; rapor iletisi yanıt kuyruğuna sığmak için hiçbir zaman kesilmez.
- Özgün iletinin MDFMT değeri FMXQH ise, raporda yer alan veriler MQXQH ' yi içermez. Rapor verileri, özgün iletideki MQXQH ' nin ötesindeki verilerin ilk byte 'ıyla başlar. Bu, kuyruğun bir iletim kuyruğu olup olmadığını belirler.

5. Yanıt kuyruğunda bir COA, COD ya da süre bitim raporu iletisi alınır, özgün iletinin geldiği, teslim edildiği ya da süresinin dolduğu garanti edilir. Ancak, bu rapor iletilerinden biri ya da daha fazlası istenirse ve alınmazsa, aşağıdakilerden biri gerçekleşmiş olabileceğinden, tersi kabul edilemez:

- a. Bir bağlantı kapalı olduğundan rapor iletisi tutuldu.
- b. Rapor iletisi, bir ara iletim kuyruğunda ya da yanıt kuyruğunda (örneğin, kuyruk dolu ya da girişler için engellenmiş) bir engelleme koşulu bulunduğu için gecikir.
- c. Rapor iletisi, gönderilmeyen iletiler kuyruğunda.
- d. Kuyruk yöneticisi rapor iletisini oluşturmayı denediğinde, iletiyi uygun kuyruğa koyamadı ve gitmeyen iletiler kuyruğuna koyamadı, bu nedenle rapor iletisi oluşturulamadı.
- e. Bildirilmekte olan işlem (varış, teslim ya da süre bitimi) ile ilgili rapor iletisinin oluşturulması arasında kuyruk yöneticisinde bir hata oluştu. (COD rapor iletisi aynı iş birimi içinde oluşturulduğundan, uygulama bir iş birimi içindeki özgün iletiyi alırsa, COD rapor iletileri için bu gerçekleşmez.)

Kural dışı durum raporu iletileri, daha önce 1, 2 ve 3 nedenleriyle aynı şekilde tutulabilir. Bununla birlikte, bir MCA bir kural dışı durum raporu iletisi oluşturamazsa (rapor iletisi yanıt kuyruğuna ya da gönderilmeyen iletiler kuyruğuna konamaz), özgün ileti gönderenin iletim kuyruğunda kalır ve kanal kapanır. Bu durum, rapor iletisinin kanalın gönderildiği yerde mi, yoksa gönderildiği yerde mi oluşturulacağına bakılmaksızın ortaya çıkar.

6. Özgün ileti geçici olarak engellenirse (bir kural dışı durum raporu oluşturulmasıyla ve özgün ileti bir teslim edilmeyen ileti kuyruğuna konmasıyla sonuçlanıyorsa), ancak engelleme temizlenir ve bir uygulama özgün iletiyi teslim edilmeyen ileti kuyruğundan okur ve yeniden hedefine yerleştirirse, aşağıdaki durum oluşabilir:

- Bir kural dışı durum raporu iletisi oluşturulmasına rağmen, özgün ileti sonunda hedefine başarıyla ulaşır.
- Özgün ileti daha sonra başka bir engelle karşılaşılabileceği için, tek bir özgün iletiye göre birden fazla kural dışı durum rapor iletisi oluşturulur.

Bir konuya yerleştirilirken iletileri raporla:

1. Bir konuya ileti yerleştirilirken raporlar oluşturulabilir. Bu ileti, konunun tüm abonelerine gönderilir; bu durum sıfır, bir ya da daha fazla olabilir. Sonuç olarak çok sayıda rapor iletisi oluşturulabileceği için rapor seçeneklerini kullanmayı seçerken bu dikkate alınmalıdır.
2. Bir konuya ileti koyarken, iletinin bir kopyasının verileceği birçok hedef kuyruk olabilir. Bu hedef kuyruklardan bazılarında kuyruk dolu gibi bir sorun varsa, MQPUT ' nin başarıyla tamamlanması NPMGDLV ya da PMSGDLV ayarına bağlıdır (iletinin kalıcı olmasına bağlı olarak). Ayar, hedef kuyruğa ileti teslimi başarılı olacaksa (örneğin, kalıcı bir aboneye kalıcı bir iletidir ve PMSGDLV için ALL ya da ALLDUR değeri belirlenmişse), başarı aşağıdaki ölçütlerden biri olarak tanımlanır:
 - Abone kuyruğuna başarıyla yerleştirilmesi
 - Abone kuyruğu iletisi alamazsa, RODLQ ve Ölü mektup kuyruğuna başarılı bir şekilde konması
 - Abone kuyruğu iletisi alamazsa RODISC kullanımı.

İleti bölümleri için rapor iletileri:

1. Bölümlemeye izin verilen iletiler için rapor iletileri istenebilir (MFSEGA işaretinin açıklamasına bakın). Kuyruk yöneticisi iletiyi bölümlere ayırmayı gerekli bulursa, daha sonra ilgili koşulla karşılaşan her bölüm için bir rapor iletilisi oluşturulabilir. Bu nedenle başvurular, istenen her rapor iletilisi tipi için birden çok rapor iletilisi almaya hazırlanmalıdır. Rapor iletilisindeki MDGID alanı, birden çok raporu özgün iletilinin grup tanıtıcısıyla ilişkilendirmek için ve her bir rapor iletilisinin tipini tanımlamak için kullanılan MDFB alanı kullanılabilir.
2. GMLOGO, bölümlere ilişkin rapor iletilerini almak için kullanılıyorsa, ardışık MQGET çağrıları tarafından farklı tiplerde raporların döndürülebileceğini unutmayın. Örneğin, kuyruk yöneticisi tarafından kesimlere ayrılmış bir ileti için hem COA hem de COD raporları istenirse, rapor iletileri için MQGET çağrıları, tahmin edilemeyen bir şekilde serpiştirilmiş COA ve COD rapor iletilerini döndürebilir. Bu, GMCMPM seçeneği (isteğe bağlı olarak GMATM ile) kullanılarak önlenir. GMCMPM, kuyruk yöneticisinin aynı rapor tipine sahip rapor iletilerini yeniden birleştirmesine neden olur. Örneğin, ilk MQGET çağrısı özgün iletiyle ilgili tüm COA iletilerini yeniden birleştirir, ikinci MQGET çağrısı tüm COD iletilerini yeniden birleşebilir. İlk olarak yeniden birleştirilen ileti tipi, kuyrukta ilk olarak hangi tip rapor iletilisinin gerçekleştiğine bağlıdır.
3. Kendi bölümlerini yerleştirdikleri uygulamalar, her bir bölüm için farklı rapor seçenekleri belirtebilir. Ancak, aşağıdaki noktalar dikkate alınmalıdır:
 - Bölümler GMCMPM seçeneği kullanılarak alınırsa, yalnızca ilk bölümdeki rapor seçenekleri kuyruk yöneticisi tarafından belirlenir.
 - Kesimler birer birer alınırsa ve çoğu ROCOD* seçeneklerinden birine sahipse, ancak en az bir kesim sahip değilse, rapor iletilerini tek bir MQGET çağrısıyla almak için GMCMPM seçeneğini kullanmak ya da tüm rapor iletilerinin ne zaman geldiğini saptamak için GMASGA seçeneğini kullanmak mümkün değildir.
4. Bir MQ ağında, kuyruk yöneticilerinin farklı yetenekleri olabilir. Bölümlemeyi desteklemeyen bir kuyruk yöneticisi ya da MCA tarafından bir bölüm için bir rapor iletilisi oluşturulursa, kuyruk yöneticisi ya da MCA varsayılan olarak rapor iletilisine gerekli bölüm bilgilerini dahil etmez ve bu, raporun oluşturulmasına neden olan özgün iletilinin saptanmasını zorlaştırabilir. Bu zorluk, uygun RO* D ya da RO* F seçenekleri belirlenerek rapor iletilisiyle veri istenerek önlenir. Ancak, RO* D belirtilirse, rapor iletilisi bölümlemeyi desteklemeyen bir kuyruk yöneticisi ya da MCA tarafından oluşturulursa, rapor iletilisini alan uygulamaya 100 bayttan az uygulama iletilisi verisinin döndürülebileceğini unutmayın.

Bir rapor iletilisi için ileti tanımlayıcısının içeriği: Kuyruk yöneticisi ya da ileti kanalı aracısı (MCA) bir rapor iletilisi oluşturduğunda, ileti tanımlayıcısındaki alanları aşağıdaki değerlere ayarlar ve iletiyi normal şekilde yerleştirir.

Çizelge 708. Bir rapor iletilisi sistem tarafından oluşturulduğunda MQMD alanları için kullanılan değerler

MQMD 'deki Alan	Kullanılan değer
MDSID	MDSIDV
MDVER	MDVER2
MDREP	YAYINI
MDMT	MTRPRT
MDEXP	EIULIM
MDFB	Raporun niteliğine uygun olarak (FBCOA, FBCOD, FBEXP ya da RC* değeri)
MDENC	Özgün ileti tanımlayıcısından kopyalandı
MDCSI	Özgün ileti tanımlayıcısından kopyalandı
MDFMT	Özgün ileti tanımlayıcısından kopyalandı
MDPRI	Özgün ileti tanımlayıcısından kopyalandı

Çizelge 708. Bir rapor iletisi sistem tarafından oluşturulduğunda MQMD alanları için kullanılan değerler (devamı var)

MQMD 'deki Alan	Kullanılan değer
MDPER	Özgün ileti tanımlayıcısından kopyalandı
MDMID	Özgün ileti tanımlayıcısındaki rapor seçenekleriyle belirtildiği gibi
MDCID	Özgün ileti tanımlayıcısındaki rapor seçenekleriyle belirtildiği gibi
MDBOC	0
MDRQ	Boşluklar
MDRM	Kuyruk yöneticisinin adı
MDUID	PPMPASI seçeneği tarafından ayarlandığı şekilde
MDACC	PPMPASI seçeneği tarafından ayarlandığı şekilde
MDAID	PPMPASI seçeneği tarafından ayarlandığı şekilde
MDPAT	ATQM ya da ileti kanalı aracısı için uygun olduğu şekilde
MDPAN	Kuyruk yöneticisi adının ya da ileti kanalı aracısı adının ilk 28 baytı. IMS köprüsü tarafından oluşturulan rapor iletileri için bu alan, iletinin ilgili olduğu IMS sisteminin XCF grup adını ve XCF üye adını içerir.
MDPD	Rapor iletisinin gönderildiği tarih
MDPT	Rapor iletisinin gönderildiği zaman
MDAOD	Boşluklar
MDGID	Özgün ileti tanımlayıcısından kopyalandı
MDSEQ	Özgün ileti tanımlayıcısından kopyalandı
MDOFF	Özgün ileti tanımlayıcısından kopyalandı
MDMFL	Özgün ileti tanımlayıcısından kopyalandı
MDOLN	OLUNDF değilse özgün ileti tanımlayıcısından kopyalanır ve özgün ileti verilerinin uzunluğuna ayarlanır; tersi durumda,

Rapor oluşturan bir uygulamanın, aşağıdakiler dışında benzer değerler ayarlaması önerilir:

- MDRM alanı boş olarak ayarlanabilir (kuyruk yöneticisi, ileti konduğunda bunu yerel kuyruk yöneticisinin adıyla değiştirir).
- Bağlam alanları, normalde PMPASI olan bir yanıt için kullanılacak seçenek kullanılarak ayarlanmalıdır.

Rapor alanının çözümlenmesi: MDREP alanı alt alanlar içerir; bu nedenle, iletiyi gönderen kişinin belirli bir rapor isteyip istemediğini denetlemesi gereken uygulamalar [“IBM i üzerinde rapor alanını analiz etme” sayfa 1410](#) içinde açıklanan tekniklerden birini kullanmalıdır.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş alanıdır. Bu alanın ilk değeri RONONE.

MDRM (48 baytlık karakter dizgisi)

Yanıt kuyruğu yöneticisinin adı.

Bu, yanıt iletisinin ya da rapor iletisinin gönderilmesi gereken kuyruk yöneticisinin adıdır. MDRQ , bu kuyruk yöneticisinde tanımlanan bir kuyruğun yerel adıdır.

MDRM alanı boşsa, yerel kuyruk yöneticisi kuyruk tanımlamalarında **MDRQ** adını arar. Bu ada sahip bir uzak kuyruk yerel tanımlaması varsa, iletilen iletideki **MDRM** değeri, uzak kuyruk tanımındaki **RemoteQMgrName** özniteliğinin değeriyle değiştirilir ve alan uygulama ileti için bir MQGET çağrısı

yayınladığında ileti tanımlayıcısında bu değer döndürülür. Uzak kuyruğun yerel tanımlaması yoksa, iletilen MDRM yerel kuyruk yöneticisinin adıdır.

Ad belirtilirse, sonda boşluk olabilir; bu karakteri izleyen ilk boş karakter ve karakterler boşluk olarak işlenir. Ancak, adın kuyruk yöneticilerine ilişkin adlandırma kurallarına uygun olup olmadığını ya da bu adın gönderen kuyruk yöneticisi tarafından bilinip bilinmediğini denetlenmez; bu, iletilen iletide **MDRM** başkasıyla değiştirilirse, iletilen ad için de geçerlidir.

Yanıt kuyruğuna gerek yoksa, MDRM alanının boşluk olarak ayarlanması önerilir; alan ilk kullanıma hazırlanmamış olarak bırakılmamalıdır.

MQGET çağrısı için, kuyruk yöneticisi her zaman alanın uzunluğuna kadar boşluklarla doldurarak adı döndürür.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın uzunluğu LNQM tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

MDRQ (48 baytlık karakter dizilimi)

Yanıt kuyruğunun adı.

Bu, iletiye ilişkin alma isteğini yayınlayan uygulamanın MTRPLY ve MTRPRT iletilerini göndermesi gereken ileti kuyruğunun adıdır. Ad, MDRM ile tanımlanan kuyruk yöneticisinde tanımlanan bir kuyruğun yerel adıdır. Bu kuyruk bir model kuyruğu olmamalıdır; ancak, gönderen kuyruk yöneticisi ileti durumunda bunu doğrulamaz.

MQPUT ve MQPUT1 çağrıları için, MDMT alanında MTRQST değeri varsa ya da MDREP alanı tarafından herhangi bir rapor iletisi isteniyorsa bu alan boş olmamalıdır. Ancak, belirtilen (ya da yerine konan) değer, ileti tipi ne olursa olsun, ileti için alma isteğini veren uygulamaya iletir.

MDRM alanı boşsa, yerel kuyruk yöneticisi kendi kuyruk tanımlamalarında MDRQ adını arar. Bu ada sahip bir uzak kuyruk yerel tanımlaması varsa, iletilen iletideki MDRQ değeri, uzak kuyruk tanımındaki **RemoteQName** özniteliğinin değeriyle değiştirilir ve alan uygulama ileti için bir MQGET çağrısı yayınladığında ileti tanımlayıcısında bu değer döndürülür. Uzak kuyruğun yerel tanımlaması yoksa, MDRQ değişmez.

Ad belirtilirse, sonda boşluk olabilir; bu karakteri izleyen ilk boş karakter ve karakterler boşluk olarak işlenir. Ters durumda, adın kuyruklara ilişkin adlandırma kurallarını karşıladığı denetlenmez; iletilen iletide MDRQ değiştirilirse, bu da iletilen ad için geçerlidir. Yapılan tek denetim, koşullar gerektiriyorsa, bir adın belirtilmiş olmasıdır.

Yanıt kuyruğuna gerek yoksa, MDRQ alanının boşluk olarak ayarlanması önerilir; alan ilk kullanıma hazırlanmamış olarak bırakılmamalıdır.

MQGET çağrısı için, kuyruk yöneticisi her zaman alanın uzunluğuna kadar boşluklarla doldurarak adı döndürür.

Rapor iletisi gerektiren bir ileti teslim edilemezse ve rapor iletisi de belirtilen kuyruğa teslim edilemezse, hem özgün ileti hem de rapor iletisi teslim edilmeyen ileti kuyruğuna gider. "[IBM i üzerindeki kuyruk yöneticisine ilişkin öznitelikler](#)" sayfa 1376'de açıklanan **DeadLetterQName** özniteliğine bakın.

Bu, MQGET çağrısına ilişkin bir çıkış alanı ve MQPUT ve MQPUT1 çağrılarında ilişkin bir giriş alanıdır. Bu alanın uzunluğu LNQN tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

MDSEQ (10 basamaklı işaretli tamsayı)

Grup içindeki mantıksal iletinin sıra numarası.

Sıra numaraları 1 'den başlar ve gruptaki her yeni mantıksal ileti için en çok 999 999 999 'a kadar 1 artırılır. Bir grupta olmayan fiziksel bir iletinin sıra numarası 1 'dir.

Bu alanın, aşağıdaki durumlarda MQPUT ya da MQGET çağrısındaki uygulama tarafından ayarlanması gerekmez:

- MQPUT çağrısında PMLOGO belirtildi.
- MQGET çağrısında MOSEQN belirtilmedi.

Bunlar, rapor iletisi olmayan iletiler için bu çağrılarını kullanmanın önerilen yollarıdır. Ancak, uygulama daha fazla denetim gerektiriyorsa ya da çağrı MQPUT1 ise, uygulamanın MDSEQ ' in uygun bir değere ayarlandığından emin olması gerekir.

MQPUT ve MQPUT1 çağrılarında girişte, kuyruk yöneticisi Tablo 1 içinde ayrıntılı olarak açıklanan değeri kullanır. MQPUT ve MQPUT1 çağrılarının çıkışında, kuyruk yöneticisi bu alanı iletiyle birlikte gönderilen değere ayarlar.

MQGET çağrısına girişte, kuyruk yöneticisi Çizelge 1' de ayrıntılı olarak açıklanan değeri kullanır. MQGET çağrısının çıkışında, kuyruk yöneticisi bu alanı alınan iletinin değerine ayarlar.

Bu alanın ilk değeri birdir. MDVER MDVER2 değerinden küçükse bu alan yoksayıdır.

MDSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

MDSIDV

İleti tanımlayıcı yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MDSIDV 'dir.

MDUID (12 baytlık karakter dizgisi)

Kullanıcı kimliği.


Bu, iletinin *kimlik bağlamının* bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için İleti bağlamı ve Bağlam bilgilerini denetleme başlıklı konuya bakın.

MDUID , iletiyi oluşturan uygulamanın kullanıcı kimliğini belirtir. Kuyruk yöneticisi bu bilgileri karakter verileri olarak kabul eder, ancak biçimini tanımlamaz.

Bir ileti alındıktan sonra MDUID , sonraki bir MQOPEN ya da MQPUT1 çağrısının **OBJDSC** değiştirgesinin ODAU alanında kullanılabilir; böylece, açma işlemini gerçekleştiren uygulama yerine MDUID kullanıcısı için yetki denetimi gerçekleştirilir.

Kuyruk yöneticisi bir MQPUT ya da MQPUT1 çağrısı için bu bilgileri oluşturduğunda, kuyruk yöneticisi ortamdan belirlenen bir kullanıcı kimliğini kullanır.

Kullanıcı kimliği ortamdan belirlendiğinde:

-  z/OS' da, kuyruk yöneticisi aşağıdakileri kullanır:
 - Toplu iş için, JES JOB kartındaki ya da başlatılan görevdeki kullanıcı kimliği
 - TSO için, oturum açma kullanıcı kimliği
 - CICS için, görevle ilişkili kullanıcı kimliği
 - IMS için, kullanıcı kimliği uygulamanın tipine bağlıdır:
 - Bunun için:
 - İleti dışı BMP bölgeleri
 - İleti dışı IFP bölgeleri
 - Başarılı bir GU çağrısı yayınlamamış olan BMP ve ileti IFP bölgeleri
- Kuyruk yöneticisi, bölge JES JOB kartındaki kullanıcı kimliğini ya da TSO kullanıcı kimliğini kullanır. Bunlar boş değerli ya da boşsa, program belirtimi öbeğinin (PSB) adını kullanır.
- Bunun için:
 - Başarılı bir GU çağrısı yayınlayan BMP ve ileti IFP bölgeleri
 - MPP bölgeleri

Kuyruk yöneticisi aşağıdakilerden birini kullanır:

- İletiyile ilişkili oturum açmış kullanıcı kimliği

- Mantıksal uçbirim (LTERM) adı
- Bölge JES JOB kartındaki kullanıcı kimliği
- TSO kullanıcı kimliği
- PCB adı
- **IBM i** IBM i' da, kuyruk yöneticisi uygulama işiyle ilişkilendirilmiş kullanıcı tanımının adını kullanır.
- **Linux** **AIX** AIX and Linux' da, kuyruk yöneticisi aşağıdakileri kullanır:
 - Uygulamanın oturum açma adı
 - Oturum açma yoksa, işlemin etkin kullanıcı kimliği
 - Uygulama bir CICS işlemiyse, işlemle ilişkili kullanıcı kimliği
- VSE/ESA üzerinde, bu ayrılmış bir alandır.
- **Windows** Windows' da, kuyruk yöneticisi oturum açmış kullanıcı adının ilk 12 karakterini kullanır.

MQPOT ve MQPUT1 çağrıları için bu, **PMO** değiştirilmesinde PMSETI ya da PMSETA belirtildiyse bir giriş/çıkış alanıdır. Alan içinde boş bir karakteri izleyen bilgiler atılır. Boş karakter ve aşağıdaki karakterler, kuyruk yöneticisi tarafından boşluklara dönüştürülür. PMSETI ya da PMSETA belirtilmezse, girişte bu alan yoksayılır ve yalnızca çıkış alanıdır.

Bir MQPUT ya da MQPUT1 çağrısı başarıyla tamamlandıktan sonra, bu alan bir kuyruğa yerleştirildiyse, iletiyle birlikte iletilen MDUID ögesini içerir. Bu değer, alkonursa iletiyle birlikte tutulan MDUID değeri olur (alınan yayınlarla ilgili daha fazla ayrıntı için PMRET açıklamasına bakın), ancak kendilerine gönderilen tüm yayınlarda MDUID değerini geçersiz kılacak bir değer sağladıkları için ileti abonelere yayın olarak gönderildiğinde MDUID olarak kullanılmaz. İletinin bağlamı yoksa, alan tamamen boştur.

Bu, MQGET çağrısına ilişkin bir çıkış alanıdır. Bu alanın uzunluğu LNUID tarafından verilir. Bu alanın ilk değeri 12 boş karakterdir.

MDVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer aşağıdakilerden biri olmalıdır:

MDVER1

Version-1 ileti tanımlayıcı yapısı.

MDVER2

Version-2 ileti tanımlayıcı yapısı.

Not: version-2 MQMD kullanıldığında, kuyruk yöneticisi uygulama ileti verilerinin başında var olabilecek MQ üstbilgi yapılarında ek denetimler gerçekleştirir; ek ayrıntılar için MQPUT çağrısına ilişkin kullanım notlarına bakın.

Yalnızca yapının daha yeni sürümünde var olan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MDVERC

İleti tanımlayıcı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MDVER1' dir.

Başlangıçtaki değerler

Çizelge 709. MQMD 'deki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
MDSID	MDSIDV	'MD??'

Çizelge 709. MÇMD 'deki Alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
MDVER	MDVER1	1
MDREP	YAYINI	0
MDMT	MTDGRM	8
MDEXP	EIULIM	-1
MDFB	FBNONE	0
MDENC	GIRIŞ	Ortama bağlıdır
MDCSI	CSQM	0
MDFMT	FMNONE	Boşluklar
MDPRI	PRQDEF	-1
MDPER	PEQDEF	2
MDMID	MINON	Boş Değerler
MDCID	TAMAMDİR.	Boş Değerler
MDBOC	Yok	0
MDRQ	Yok	Boşluklar
MDRM	Yok	Boşluklar
MDUID	Yok	Boşluklar
MDACC	YOK	Boş Değerler
MDAID	Yok	Boşluklar
MDPAT	ATNCON	0
MDPAN	Yok	Boşluklar
MDPD	Yok	Boşluklar
MDPT	Yok	Boşluklar
MDAOD	Yok	Boşluklar
MDGID	GINONE	Boş Değerler
MDSEQ	Yok	1
MDOFF	Yok	0
MDMFL	MFNONE	0
MDOLN	OLUNDF	-1

Notlar:

1. - simgesi, tek bir boş karakteri temsil eder.

RPG bildirim

D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MÇMD 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 message data
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('00000000000000-
00000000000000000000-
000000000000')
D
D* Correlation identifier
D MDCID 73 96 INZ('00000000000000-
00000000000000000000-
000000000000')
D
D* Backout counter
D MDBOC 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('00000000000000-
00000000000000000000-
000000000000000000-
000000')
D
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('00000000000000-
00000000000000000000-
000000000000')
D
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

IBM i üzerinde MQMDE (İleti tanımlayıcı uzantısı)

Genel Bakış

Amaç: MQMDE yapısı, bazen uygulama ileti verilerinden önce gelen verileri tanımlar. Yapı, version-2 MQMD 'de var olan, ancak version-1 MQMD' de bulunmayan MQMD alanlarını içeriyor.

Biçim adı: FMMDE.

Karakter kümesi ve kodlama: MQMDE içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği ve C programlama dili için ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır.

MQMDE ' nin karakter kümesi ve kodlaması aşağıdaki alanlarda *MDCSI* ve *MDENC* alanlarına ayarlanmalıdır:

- MQMD (MQMDE yapısı ileti verilerinin başındadır) ya da
- MQMDE yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

MQMDE, kuyruk yöneticisinin karakter kümesinde ve kodlamasında yoksa, MQMDE kabul edilir, ancak yerine getirilmezse, MQMDE ileti verisi olarak işlenir.

Kullanım: Normal uygulamalar version-2 MQMD ' yi kullanmalıdır; bu durumda bir MQMDE yapısıyla karşılaşmazlar. Ancak, version-1 MQMD ' yi kullanmaya devam eden özel uygulamalar ve uygulamalar bazı durumlarda MQMDE ile karşılaşabilir. MQMDE yapısı aşağıdaki durumlarda oluşabilir:

- MQPUT ve MQPUT1 çağrılarında belirtildi
- MQGET çağrısı tarafından döndürüldü
- İletim kuyruklarındaki iletilerde
- [“MQPUT ve MQPUT1 çağrılarında MQMDE belirtildi” sayfa 1135](#)
- [“MQGET çağrısı tarafından döndürülen MQMDE” sayfa 1136](#)
- [“İletim kuyruklarındaki iletilerde MQMDE” sayfa 1136](#)
- [“Alanlar” sayfa 1136](#)
- [“Başlangıçtaki değerler” sayfa 1138](#)
- [“RPG bildirimi” sayfa 1139](#)

MQPUT ve MQPUT1 çağrılarında MQMDE belirtildi

MQPUT ve MQPUT1 çağrılarında, uygulama bir version-1 MQMD 'si sağlarsa, uygulama isteğe bağlı olarak ileti verilerine bir MQMDE ile örnek verebilir ve MQMD 'deki *MDFMT* alanını bir MQMDE ' nin var olduğunu göstermek için FMMDE olarak ayarlayabilir. Uygulama bir MQMDE sağlamazsa, kuyruk yöneticisi MQMDE 'deki alanlar için varsayılan değerleri varsayar. Kuyruk yöneticisinin kullandığı varsayılan değerler, yapının ilk değerleriyle aynıdır-bkz. [Çizelge 711 sayfa 1138](#).

Uygulama bir version-2 MQMD sağlarsa ve , uygulama ileti verilerini MQMDE ile örneklerse, yapılar [Çizelge 710 sayfa 1135](#) içinde gösterildiği gibi işlenir.

MQMD sürümü	version-2 alanlarının değerleri	MQMDE içindeki ilgili alanların değerleri	Kuyruk yöneticisi tarafından gerçekleştirilen işlem
1	-	Geçerli	MQMDE onurlandırıldı
2	Varsayılan	Geçerli	MQMDE onurlandırıldı
2	Varsayılan değil	Geçerli	MQMDE, ileti verisi olarak kabul edilir
1 ya da 2	Herhangi Biri	Geçerli değil	Arama uygun bir neden koduyla başarısız oluyor
1 ya da 2	Herhangi Biri	MQMDE yanlış karakter kümesinde ya da kodlamada ya da desteklenmeyen bir sürümde	MQMDE, ileti verisi olarak kabul edilir

Özel bir durum var. Uygulama bir kesim (MFSEG ya da MFLSEG işareti) olan bir iletiyi koymak için version-2 MQMD kullanıyorsa ve MQMD 'deki biçim adı FMDLH ise, kuyruk yöneticisi bir MQMDE yapısı

oluşturur ve bunu *arasında* MQDLH yapısını ve onu izleyen verileri ekler. Kuyruk yöneticisinin iletiyle birlikte sakladığı MQMD ' de version-2 alanları varsayılan değerlerine ayarlanır.

version-2 MQMD 'de var olan, ancak version-1 MQMD' de olmayan alanlar, MQPUT ve MQPUT1' de giriş/çıkış alanlarıdır. Ancak, kuyruk yöneticisi MQPUT ve MQPUT1 çağrılarının çıkışında MQMDE ' deki eşdeğer alanlarda herhangi bir değer döndürmez; uygulama bu çıkış değerlerini gerektiriyorsa, bir version-2 MQMD kullanmalıdır.

MQGET çağrısı tarafından döndürülen MQMDE

MQGET çağrısında, uygulama bir version-1 MQMD sağlarsa, kuyruk yöneticisi MQMDE ile döndürülen iletiyi örnekler, ancak MQMDE ' deki alanlardan birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa. Kuyruk yöneticisi, MQMD 'deki *MDFMT* alanını bir MQMDE' nin var olduğunu göstermek için FMMDE değerine ayarlar.

Uygulama **BUFFER** değiştirgesinin başında bir MQMDE sağlarsa, MQMDE yoksayılar. MQGET çağrısından döndüğünüzde, ileti için MQMDE ile değiştirilir (gerekliyse) ya da uygulama ileti verileri tarafından üzerine yazılır (MQMDE gerekli değilse).

MQGET çağrısı bir MQMDE döndürürse, MQMDE ' deki veriler genellikle kuyruk yöneticisinin karakter kümesinde ve kodlamasında bulunur. Ancak, MQMDE aşağıdaki durumda başka bir karakter kümesinde ve kodlamada olabilir:

- MQMDE, MQPUT ya da MQPUT1 çağrısında veri olarak kabul edildi (buna neden olabilecek durumlar için bkz. Çizelge 710 sayfa 1135).
- İleti, TCP bağlantısıyla bağlanan bir uzak kuyruk yöneticisinden alındı ve alan ileti kanalı aracısı (MCA) doğru ayarlanmadı (ek bilgi için [IBM MQ for IBM i nesnelerinin güvenliği](#) konusuna bakın).

İletim kuyruklarındaki iletilerde MQMDE

İletim kuyruklarındaki iletilerin başına, içinde version-1 MQMD içeren MQXQH yapısı eklenir. MQXQH yapısı ve uygulama iletileri arasında konumlandırılmış bir MQMDE de var olabilir; ancak, MQMDE ' deki alanlardan birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa bu durum tipik olarak ortaya çıkabilir.

Diğer IBM MQ üstbilgi yapıları, MQXQH yapısı ile uygulama iletileri arasında da oluşabilir. Örneğin, gitmeyen harf üstbilgisi MQDLH varsa ve ileti bir bölüm değilse, sıralama şöyle olur:

- MQXQH (version-1 MQMD 'si içerir)
- MQMDE
- MQDLH
- Uygulama iletileri verileri

Alanlar

MQMDE yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

MECSI (10 basamaklı işaretli tamsayı)

MQMDE ' yi izleyen verilerin karakter kümesi tanıtıcısı.

MQMDE yapısını izleyen verilerin karakter kümesi tanıtıcısını belirtir; MQMDE yapısının kendisindeki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Kuyruk yöneticisi bu alanın geçerli olup olmadığını denetlemez. Aşağıdaki özel değer kullanılabilir:

CSINHT

Bu yapının karakter kümesi tanıtıcısını edin.

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MÖGET çağrısı CSINHT değerini döndürmez.

MÖMD ' deki *MDPAT* alanının değeri *ATBRKR* ise, CSINHT kullanılamaz.

Bu alanın ilk değeri CSUNDF 'dir.

MEENC (10 basamaklı işaretli tamsayı)

MEENC (10 basamaklı işaretli tamsayı)

MÖMDE yapısını izleyen verilerin sayısal kodlamasını belirtir; MÖMDE yapısının kendisindeki sayısal veriler için geçerli değildir.

MÖPUT ya da MÖPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Kuyruk yöneticisi, alanın geçerli olup olmadığını denetlemez. Veri kodlamasıyla ilgili ek bilgi için [“IBM i üzerinde MÖMD \(İleti tanımlayıcı\)” sayfa 1093](#) içinde açıklanan *MDENC* alanına bakın.

Bu alanın ilk değeri ENNAT ' tır.

MEFLG (10 basamaklı işaretli tamsayı)

Genel bayraklar.

Aşağıdaki işaret belirtilebilir:

MEFNON

Bayrak yok.

Bu alanın ilk değeri MEFNON ' dur.

MEFMT (8 baytlık karakter dizgisi)

MÖMDE ' yi izleyen verilerin biçim adı.

MÖMDE yapısını izleyen verilerin biçim adını belirtir.

MÖPUT ya da MÖPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Kuyruk yöneticisi bu alanın geçerli olup olmadığını denetlemez. Biçim adlarıyla ilgili ek bilgi için [“IBM i üzerinde MÖMD \(İleti tanımlayıcı\)” sayfa 1093](#) içinde açıklanan *MDFMT* alanına bakın.

Bu alanın ilk değeri FMNONE.

MEGID (24 baytlık bit dizgisi)

Grup tanıtıcısı.

[“IBM i üzerinde MÖMD \(İleti tanımlayıcı\)” sayfa 1093](#) içinde açıklanan *MDGID* alanına bakın. Bu alanın ilk değeri GINONE.

MELEN (10 basamaklı işaretli tamsayı)

MÖMDE yapısının uzunluğu.

Aşağıdaki değer tanımlandı:

MELEN2

version-2 ileti tanımlayıcısı uzantı yapısının uzunluğu.

Bu alanın ilk değeri MELEN2' dir.

MEMFL (10 basamaklı işaretli tamsayı)

İleti işaretleri.

[“IBM i üzerinde MÖMD \(İleti tanımlayıcı\)” sayfa 1093](#) içinde açıklanan *MDMFL* alanına bakın. Bu alanın ilk değeri MFNONE.

MEOFF (10 basamaklı işaretli tamsayı)

Mantıksal iletinin başlangıcından fiziksel iletideki verilerin görelî konumu.

[“IBM i üzerinde MÖMD \(İleti tanımlayıcı\)” sayfa 1093](#) içinde açıklanan *MDOFF* alanına bakın. Bu alanın ilk değeri 0 'dir.

MEOLN (10 basamaklı işaretli tamsayı)

Özgün iletinin uzunluğu.

"IBM i üzerinde MQMD (İleti tanımlayıcı)" sayfa 1093 içinde açıklanan *MDOLN* alanına bakın. Bu alanın ilk değeri OLUNDF 'dir.

MESEQ (10 basamaklı işaretli tamsayı)

Grup içindeki mantıksal iletinin sıra numarası.

"IBM i üzerinde MQMD (İleti tanımlayıcı)" sayfa 1093 içinde açıklanan *MDSEQ* alanına bakın. Bu alanın ilk değeri 1 'dir.

MESID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

MESIDV

İleti tanımlayıcı uzantı yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri MESIDV ' dir.

MEVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

MEVER2

Version-2 ileti tanımlayıcısı uzantı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MEVERC (ORTALAMA)

İleti tanımlayıcı uzantı yapısının yürürlükteki sürümü.

Bu alanın ilk değeri MEVER2' dir.

Başlangıçtaki değerler

<i>Çizelge 711. MQMDE içindeki alanlar</i>		
Alan adı	Değişmezin adı	Değişmezin değeri
<i>MESID</i>	MESIDV	'MDE↵'
<i>MEVER</i>	MEVER2	2
<i>MELEN</i>	MELEN2	72
<i>MEENC</i>	GİRİŞ	Ortama bağlıdır
<i>MECSI</i>	CSUNDF	0
<i>MEFMT</i>	FMNONE	Boşluklar
<i>MEFLG</i>	MEFNON	0
<i>MEGID</i>	GINONE	Boş Değerler
<i>MESEQ</i>	Yok	1
<i>MEOFF</i>	Yok	0
<i>MEMFL</i>	MFNONE	0
<i>MEOLN</i>	OLUNDF	-1

Çizelge 711. MQMDE içindeki alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
Notlar:		
1. - simgesi, tek bir boş karakteri temsil eder.		

RPG bildirimi

```
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 MECSE 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('00000000000000-
00000000000000000000-
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

IBM i üzerinde MQMHBO (Arabellek seçeneklerine ileti tanıtıcısı)

İleti tanıtıcısını arabellek seçeneklerine tanımlayan yapı

Genel Bakış

Amaç: MQMHBO yapısı, uygulamaların ileti tanıtıcılarından arabelleklerin nasıl üretildiğini denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQMHBUF çağrısındaki bir giriş değişikliğidir.

Karakter kümesi ve kodlama: MQMHBO 'daki veriler, uygulamanın karakter kümesinde ve uygulamanın (ENNAT) kodlamasında olmalıdır.

- “Alanlar” sayfa 1139
- “Başlangıçtaki değerler” sayfa 1140
- “RPG bildirimi” sayfa 1140

Alanlar

MQMHBO yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

MBOPT (10 basamaklı işaretli tamsayı)

İleti tanıtıcısı-arabellek seçenekleri yapısı-MBOPT alanı.

Bu seçenekler MQMHBUF işlemini denetler.

Aşağıdaki seçeneği belirtmeniz gerekir:

MBPRRF

Özellikleri bir ileti tanıtıcısından arabelleğe dönüştürürken, bunları MQRFH2 biçimine dönüştürün. İsteğe bağlı olarak, aşağıdaki seçeneği de belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyse).

MBDLPR

Arabelleğe eklenen özellikler ileti tanıtıcısından silinir. Arama başarısız olursa, hiçbir özellik silinmez.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MBPRRF 'dir.

MBSID (10 basamaklı işaretli tamsayı)

İleti tanıtıcısı-arabellek seçenekleri yapısı-MBSID alanı.

Bu, yapı tanıtıcısıdır. Değer şöyle olmalıdır:

MBSIDV

Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısı tanıtıcısı.

Bu her zaman bir giriş alanıdır. Bu isMBSIDValanının ilk değeri.

MBVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

MBVER1

Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısının sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MAXIMO İş NESNESI

Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri MBVER1' dir.

Başlangıçtaki değerler

Çizelge 712. MQMHBO 'daki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
MVSID	MBSIDV	'MHBO'
MBVER	MBVER1	1
MBOPT	MBPRRF	

Notlar:

1. Boş değerli dizilim ya da boşluklar boş bir karakteri gösterir.

RPG bildirim

```
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 yapısı, ada göre bir nesne belirtmek için kullanılır.

Genel Bakış

Amaç: Aşağıdaki nesne tipleri geçerlidir:

- Kuyruk ya da dağıtım listesi
- Ad listesi
- Süreç tanımlaması
- Kuyruk yöneticisi
- Konu

Yapı, MQOPEN ve MQPUT1 çağrılarında bir giriş/çıkış değiştirgesidir.

Sürüm: MQOD ' un yürürlükteki sürümü ODVER4. Yapının yalnızca daha yeni sürümlerinde bulunan alanlar, izleyen açıklamalarda bu şekilde tanımlanır.

Sağlanan COPY dosyası, ortam tarafından desteklenen en son MQOD sürümünü içeriyor; ancak, *ODVER* alanının ilk değeri ODVER1olarak ayarlanmış. version-1 yapısında var olmayan alanları kullanmak için uygulamanın *ODVER* alanını gerekli sürümün sürüm numarasına ayarlaması gerekir.

Bir dağıtım listesini açmak için *ODVER* , ODVER2 ya da daha büyük olmalıdır.

Karakter kümesi ve kodlama: MQOD içindeki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

- [“Alanlar” sayfa 1141](#)
- [“Başlangıçtaki değerler” sayfa 1148](#)
- [“RPG bildirim” sayfa 1149](#)

Alanlar

MQOD yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

ODASI (40 baytlık bit dizgisi)

Diğer güvenlik tanıtıcısı.

Bu, uygun yetki denetimlerinin gerçekleştirilmesine izin vermek için *ODAU* ile yetkilendirme hizmetine iletilen bir güvenlik tanıtıcısıdır. *ODASI* yalnızca aşağıdaki durumda kullanılır:

- MQOPEN çağrısında OOALTU belirtildi ya da
- MQPUT1 çağrısında PMALTU belirtildi,

ve *ODAU* alanı, alanın ilk boş karakterine ya da sonuna kadar tamamen boş değildir.

ODASI alanı aşağıdaki yapıya sahiptir:

- İlk bayt, izleyen önemli verilerin uzunluğunu içeren ikili bir tamsayıdır; değer, uzunluk byte 'ının kendisini dışlar. Güvenlik tanıtıcısı yoksa, uzunluk sıfır olur.
- İkinci bayt, var olan güvenlik tanıtıcısının tipini gösterir; aşağıdaki değerler kullanılabilir:

SITWNT

Windows güvenlik tanıtıcısı.

DURUM

Güvenlik tanıtıcısı yok.

- Birinci baytın tanımladığı uzunluğa kadar olan üçüncü ve sonraki baytlar güvenlik tanıtıcısının kendisini içerir.

- Alandaki kalan bayt sayısı ikili sıfır olarak ayarlanır.

Aşağıdaki özel değer kullanılabilir:

SINONE.

Güvenlik tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNSCID tarafından verilir. Bu alanın ilk değeri SINONE. ODVER ODVER3değerinden küçükse bu alan yoksayılır.

ODAU (12 baytlık karakter dizgisi)

Diğer kullanıcı kimliği.

MQOPEN çağrısı için OOALTU ya da MQPUT1 çağrısı için PMALTU belirtilirse, bu alan, uygulamanın çalışmakta olduğu kullanıcı kimliği yerine, açma yetkisini denetlemek için kullanılacak diğer bir kullanıcı kimliğini içerir. Ancak, bazı denetimler yürürlükteki kullanıcı kimliğiyle (örneğin, bağlam denetimleri) gerçekleştirilmeye devam eder.

OOALTU ve PMALTU belirtilmezse ve bu alan, ilk boş karakter ya da alanın sonuna kadar tamamen boşsa, açma işlemi yalnızca belirlenen seçeneklerle bu nesneyi açmak için kullanıcı yetkisi gerekmediğinde başarılı olabilir.

OOALTU ya da PMALTU belirtilmezse, bu alan yoksayılır.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNUID tarafından verilir. Bu alanın ilk değeri 12 boş karakterdir.

ODDN (48 baytlık karakter dizgisi)

Dinamik kuyruk adı.

MQOPEN çağrısıyla yaratılacak dinamik kuyruğun adıdır. Bu yalnızca ODDN bir model kuyruğunun adını belirttiğinde geçerlidir; diğer tüm durumlarda ODDN yoksayılır.

Adda geçerli olan karakterler, yıldız işaretinin de geçerli olması dışında, ODDN için geçerli olan karakterlerle aynıdır. ODDN bir model kuyruğunun adıyla, boşluk (ya da ilk boş karakterden önce yalnızca boşlukların gösterildiği bir ad) geçerli değildir.

Addaki son boşluk olmayan karakter bir yıldız işaretiyse (*), kuyruk yöneticisi yıldız işaretini, kuyruk için oluşturulan adın yerel kuyruk yöneticisinde benzersiz olmasını garanti eden bir karakter dizgisiyle değiştirir. Bunun için yeterli sayıda karaktere izin vermek için, yıldız işareti yalnızca 1-33 arasındaki konumlarda geçerlidir. Yıldız işaretinden sonra boşluk ya da boş karakter dışında bir karakter olmamalıdır.

Yıldız işaretinin ilk karakter konumunda oluşması için geçerlidir; bu durumda ad yalnızca kuyruk yöneticisi tarafından oluşturulan karakterlerden oluşur.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNQN tarafından verilir. Bu alanın ilk değeri 'AMQ.*', boşluklarla doldurulur.

ODIDC (10 basamaklı işaretli tamsayı)

Açılmayan kuyrukların sayısı.

Bu, dağıtım listesindeki, başarıyla açılmayan kuyrukların sayısıdır. Varsa, dağıtım listesinde olmayan tek bir kuyruk açılırken de bu alan ayarlanır.

Not: Varsa, bu alan yalnızca MQOPEN ya da MQPUT1 çağrısındaki **CMPCOD** değiştirgesi CCOK ya da CCWARN ise ayarlanır; **CMPCOD** değiştirgesi CCFAIL ise ayarlanmaz.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0'dır. ODVER ODVER2değerinden küçükse bu alan yoksayılır.

ODKDC (10 basamaklı işaretli tamsayı)

Başarıyla açılan yerel kuyruk sayısı.

Bu, dağıtım listesindeki yerel kuyruklara çözülecek ve başarıyla açılan kuyrukların sayısıdır. Bu sayı, uzak kuyruklara çözülecek kuyrukları içermez (iletiyi saklamak için başlangıçta yerel bir iletim kuyruğu kullanılsa bile). Varsa, dağıtım listesinde olmayan tek bir kuyruk açılırken de bu alan ayarlanır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır. *ODVER* *ODVER2*değerinden küçükse bu alan yoksayılr.

ODMN (48 baytlık karakter dizgisi)

Nesne kuyruğu yöneticisi adı.

Bu, *ODON* nesnesinin tanımlandığı kuyruk yöneticisinin adıdır. Adda geçerli olan karakterler, *ODON* ile aynıdır (daha önce bakınız). İlk boş değere kadar ya da alanın sonuna kadar tümüyle boş bırakılan bir ad, uygulamanın bağlı olduğu kuyruk yöneticisini (yerel kuyruk yöneticisi) gösterir.

Aşağıdaki noktalar, belirtilen nesne tipleri için geçerlidir:

- *ODOT* *OTTOP*, *OTNLST*, *OTPRO* ya da *OTQM* ise, *ODMN* boş olmalı ya da yerel kuyruk yöneticisinin adı olmalıdır.
- *ODON* bir model kuyruğunun adıysa, kuyruk yöneticisi model kuyruğunun özniteliklerini içeren dinamik bir kuyruk yaratır ve *ODMN* alanında kuyruğun yaratıldığı kuyruk yöneticisinin adını döndürür; bu, yerel kuyruk yöneticisinin adıdır. Model kuyruğu yalnızca *MQOPEN* çağrısında belirtilebilir; *MQPUT1* çağrısında model kuyruğu geçerli değil.
- *ODON* bir küme kuyruğunun adıysa ve *ODMN* boşsa, *MQOPEN* çağrısının döndürdüğü kuyruk tanıtıcısı kullanılarak gönderilen iletilerin gerçek hedefi, kuyruk yöneticisi (ya da kuruluysa, küme iş yükü çıkışı) tarafından aşağıdaki gibi seçilir:
 - *OOBUDO* belirtilirse, kuyruk yöneticisi *MQOPEN* çağrısının işlenmesi sırasında küme kuyruğunun bir eşgörünümünü seçer ve bu kuyruk tanıtıcısı kullanılarak konan tüm iletiler o yönetim ortamına gönderilir.
 - *OOBNDN* belirtilirse, kuyruk yöneticisi bu kuyruk tanıtıcısını kullanan her ardışık *MQPUT* çağrısı için hedef kuyruğun farklı bir eşgörünümünü seçebilir (kümede farklı bir kuyruk yöneticisinde bulunur).

Uygulamanın bir küme kuyruğunun *belirli* bir eşgörünümüne (yani, kümedeki belirli bir kuyruk yöneticisinde bulunan bir kuyruk eşgörünümüne) ileti göndermesi gerekiyorsa, uygulama *ODMN* alanında o kuyruk yöneticisinin adını belirtmelidir. Bu, yerel kuyruk yöneticisini iletiyi belirtilen hedef kuyruk yöneticisine göndermeye zorlar.

- Açılmakta olan nesne bir dağıtım listesiye (yani, *ODREC* sıfırdan büyükse), *ODMN* boş değerli ya da boş değerli bir dizgi olmalıdır. Bu koşul karşılanmazsa, çağrı *RC2153*neden koduyla başarısız olur.

Bu, *ODON* bir model kuyruğunun adı ve diğer tüm durumlarda yalnızca giriş alanı olduğunda *MQOPEN* çağrısına ilişkin bir giriş/çıkış alanıdır. Bu alanın uzunluğu *LNQMN* tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

ODON (48 baytlık karakter dizgisi)

Nesne adı.

Bu, *ODMN* ile tanımlanan kuyruk yöneticisinde tanımlandığı şekilde, nesnenin yerel adıdır. Ad aşağıdaki karakterleri içerebilir:

- Büyük harf alfabetik karakterler (A-Z)
- Küçük harf alfabetik karakterler (a-z)
- Sayısal basamaklar (0-9)
- Nokta (.), eğik çizgi (/), altçizgi (_), yüzde (%)

Ad başta ya da gömülü boşluklar içermemelidir, ancak sonda boşluklar içerebilir. Addaki önemli verilerin sonunu göstermek için boş karakter kullanılabilir; boş değer ve ardından gelen karakterler boşluk olarak değerlendirilir. Belirtilen ortamlarda aşağıdaki kısıtlamalar geçerlidir:

- *EBCDIC* Katakana kullanan sistemlerde küçük harfli karakterler kullanılamaz.

- IBM üzerinde, komutlarda belirtildiğinde küçük harf, eğik çizgi ya da yüzde içeren adlar tırnak içine alınmalıdır. Bu tırnak işaretleri, yapılarda alan olarak ya da çağrılarda parametre olarak oluşan adlar için belirlenmemelidir.

Aşağıdaki noktalar, belirtilen nesne tipleri için geçerlidir:

- *ODON* bir model kuyruğunun adıysa, kuyruk yöneticisi model kuyruğunun özniteliklerini içeren dinamik bir kuyruk yaratır ve *ODON* alanında yaratılan kuyruğun adını döndürür. Model kuyruğu yalnızca *MQOPEN* çağrısında belirtilebilir; *MQPUT1* çağrısında model kuyruğu geçerli değil.
- Açılmakta olan nesne bir dağıtım listesiye (yani, *ODREC* var ve sıfırdan büyük), *ODON* boş değerli ya da boş değerli olmalıdır. Bu koşul karşılanmazsa, çağrı *RC2152* neden koduyla başarısız olur.
- *ODOT* *OTQM* ise, özel kurallar geçerlidir; bu durumda ad, ilk boş karakter ya da alanın sonuna kadar tamamen boş olmalıdır.
- *ODON*, *TARGETYPE* (*TOPIC*) ile bir diğer ad kuyruğunun adıysa, diğer ad kuyruklarının kullanımı için olağan olduğu gibi, önce adı belirtilen diğer ad kuyruğunda bir güvenlik denetimi yapılır. Bu güvenlik denetimi başarılı olursa, bu *MQOPEN* çağrısı devam eder ve denetim konusu nesnesiyle ilgili bir güvenlik denetimi de içinde olmak üzere, bir *OTTOP* *MQOPEN* çağrısı gibi davranır.

Bu, *ODON* bir model kuyruğunun adı ve diğer tüm durumlarda yalnızca giriş alanı olduğunda *MQOPEN* çağrısına ilişkin bir giriş/çıkış alanıdır. Bu alanın uzunluğu *LNQN* tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

Tam konu adı iki farklı alandan oluşturulabilir: *ODON* ve *ODOS*. Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.

ODORO (10 basamaklı işaretli tamsayı)

MQOD ' un başlangıcından ilk nesne kaydının görel konumu.

Bu, *MQOD* yapısının başlangıcından itibaren ilk *MQOR* nesne kaydının bayt cinsinden görel konumudur. Görel konum pozitif ya da negatif olabilir. *ODORO* yalnızca bir dağıtım listesi açıldığında kullanılır. *ODREC* sıfırsa alan yoksayılr.

Dağıtım listesi açılırken, dağıtım listesindeki hedef kuyrukların adlarını belirtmek için bir ya da daha fazla *MQOR* nesne kaydı dizisi sağlanmalıdır. Bu iki yoldan biriyle yapılabilir:

- *ODORO* görel konum alanını kullanarak

Bu durumda, uygulama *MQOD* içeren kendi yapısını ve ardından *MQOR* kayıtları dizisini (gerektiği kadar dizi ögesi olan) bildirmeli ve *ODORO* ögesini *MQOD* ' un başlangıcından itibaren dizideki ilk ögenin görel konumuna ayarlamalıdır. Bu kaymanın doğru olduğundan emin olmak için dikkatli olunmalıdır.

- *ODORP* gösterge alanını kullanarak

Bu durumda, uygulama *MQOR* yapılarının dizisini *MQOD* yapısından ayrı olarak bildirebilir ve *ODORP* değerini dizinin adresine ayarlayabilir.

Hangi teknik seçilirse seçilsin, *ODORO* ve *ODORP* yöntemlerinden biri kullanılmalıdır; her ikisi de sıfırsa ya da her ikisi de sıfır değilse, çağrı *RC2155* neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *ODVER* *ODVER2* değerinden küçükse bu alan yoksayılr.

ODORP (gösterge)

İlk nesne kaydının adresi.

Bu, ilk *MQOR* nesne kaydının adresidir. *ODORP* yalnızca bir dağıtım listesi açıldığında kullanılır. *ODREC* sıfırsa alan yoksayılr.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş işaretçidir. Nesne kayıtlarını belirlemek için *ODORP* ya da *ODORO* kullanılabilir, ancak her ikisini birden belirleyemez; ayrıntılar için daha önce *ODORO* alanının tanımına bakın. *ODORP* kullanılmazsa, boş değerli göstergeye ya da boş değerli byte 'lara ayarlanmalıdır. *ODVER* *ODVER2* değerinden küçükse bu alan yoksayılr.

ODOS (MÇCHARV)

ODOS, kullanılacak uzun nesne adını belirtir.

Bu alana yalnızca belirli *ODOT* değerleri için başvurulur. Bu alanın kullanıldığını gösteren değerlerin ayrıntıları için *ODOT* açıklamasına bakın.

ODOS yanlış belirtilirse, *MÇCHARV* yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı RC2441 neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu yapıdaki alanların başlangıç değerleri, *MÇCHARV* yapısındaki değerlerle aynıdır.

Tam konu adı iki farklı alandan oluşturulabilir: *ODON* ve *ODOS*. Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın. *ODVER* ODVER4 değerinden küçükse bu alan yoksayılr.

ODOT (10 basamaklı işaretli tamsayı)

Nesne tipi.

ODON içinde adlandırılmakta olan nesnenin tipi. Olası değerler şunlardır:

OTQ

Sıraya girin. Nesnenin adı *ODON* içinde bulunur.

OTNLST

-Namelist. Nesnenin adı *ODON* içinde bulunur.

OTPRO

Süreç tanımlaması. Nesnenin adı *ODON* içinde bulunur.

OTQM (Kalite Yönetimi)

Kuyruk yöneticisi. Nesnenin adı *ODON* içinde bulunur.

OTTOP

Konu. Tam konu adı iki farklı alandan oluşturulabilir: *ODON* ve *ODOS*.

Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.

ODON alanıyla tanıtilan nesne bulunamazsa, *ODOS* içinde bir dizgi belirtilmiş olsa bile, çağrı RC2425 neden koduyla başarısız olur.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri OTQ 'dur.

ODREC (10 basamaklı işaretli tamsayı)

Var olan nesne kayıtlarının sayısı.

Bu, uygulama tarafından sağlanan MÇOR nesne kayıtlarının sayısıdır. Bu sayı sıfırdan büyükse, bir dağıtım listesinin açılmakta olduğunu ve *ODREC* ' un listedeki hedef kuyrukların sayısı olduğunu gösterir. Bir dağıtım listesinin tek bir hedef içermesi geçerlidir.

ODREC değeri sıfırdan küçük olmamalıdır ve sıfırdan büyükse *ODOT* OTQ olmalıdır; bu koşullar karşılanmazsa çağrı RC2154 neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *ODVER* ODVER2 değerinden küçükse bu alan yoksayılr.

ODRMN (48 baytlık karakter dizgisi)

Çözölmüş kuyruk yöneticisi adı.

Bu, ad çözmeye işlemi yerel kuyruk yöneticisi tarafından gerçekleştirildikten sonra hedef kuyruk yöneticisinin adıdır. Döndürölen ad, *ODRQN* ile tanıtilan kuyruğun iyesi olan kuyruk yöneticisinin adıdır. *ODRMN* , yerel kuyruk yöneticisinin adı olabilir.

ODRQN , yerel kuyruk yöneticisinin ait olduđu kuyruk paylaşım grubunun sahip olduđu paylaşılan bir kuyruksa, *ODRMN* , kuyruk paylaşım grubunun adıdır. Kuyruk başka bir kuyruk paylaşım grubuna aitse, *ODRQN* kuyruk paylaşım grubunun adı ya da kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisinin adı olabilir (döndürölen değerin türü, yerel kuyruk yöneticisinde var olan kuyruk tanımlamalarına göre belirlenir).

Boş olmayan bir değer, nesne göz atma, giriş ya da çıkış (ya da herhangi bir birleşim) için açılmış tek bir kuyruksa döndürülür. Açılan nesne aşağıdakilerden biriye, *ODRMN* boş olarak ayarlanır:

- Kuyruk değil
- Bir kuyruk, ancak göz atma, giriş ya da çıkış için açılmadı
- OOBNDN değeri belirlenmiş bir küme kuyruğu (ya da **DefBind** kuyruk özniteliği BNDNOT değerine sahip olduğunda OOBNDQ ile)
- Dağıtım listesi

Bu bir çıkış alanıdır. Bu alanın uzunluğu LNQN tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir. *ODVER* ODVER3değerinden küçükse bu alan yoksayıdır.

ODRO (MQCHARV)

ODRO, kuyruk yöneticisi *ODON* içinde sağlanan adı çözdükten sonra uzun nesne adıdır.

Bu alan yalnızca, bir konu nesnesine gönderme yapan belirli nesne, konu ve kuyruk diğer adları tipleri için döndürülür.

Uzun nesne adı *ODOS* içinde sağlanırsa ve *ODON* içinde hiçbir şey sağlanmazsa, bu alanda döndürülen değer, *ODOS* içinde sağlananla aynıdır.

Bu alan atlanırsa (*ODRO.VSBufSize* sıfır), *ODRO* döndürülmez, ancak uzunluk *ODRO.VSLength*. Uzunluk tam *ODRO* değerinden kısaysa, kesilir ve sağlanan uzunluğa sığabilecek en sağdaki karakterlerin sayısını döndürür.

ODRO yanlış belirtilirse, *MQCHARV* yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı RC2520neden koduyla başarısız olur. *ODVER* ODVER4değerinden küçükse bu alan yoksayıdır.

ODRQN (48 baytlık karakter dizilimi)

Çözülmüş kuyruk adı.

Ad çözme işlemi yerel kuyruk yöneticisi tarafından gerçekleştirildikten sonra hedef kuyruğun adıdır. Döndürülen ad, *ODRMN* ile tanımlanan kuyruk yöneticisinde bulunan bir kuyruğun adıdır.

Boş olmayan bir değer, nesne göz atma, giriş ya da çıkış (ya da herhangi bir birleşim) için açılmış tek bir kuyruksa döndürülür. Açılan nesne aşağıdakilerden biriye, *ODRQN* boş olarak ayarlanır:

- Kuyruk değil
- Bir kuyruk, ancak göz atma, giriş ya da çıkış için açılmadı
- Dağıtım listesi
- Bir konu nesnesine gönderme yapan bir diğer ad kuyruğu (bunun yerine "ODRO (MQCHARV)" sayfa 1146 konusuna bakın)

Bu bir çıkış alanıdır. Bu alanın uzunluğu LNQN tarafından verilir. Bu alanın ilk değeri, C içindeki boş dizgi ve diğer programlama dillerindeki 48 boş karakterdir. *ODVER* ODVER3değerinden küçükse bu alan yoksayıdır.

ODRRO (10 basamaklı işaretli tamsayı)

MQOD ' nin başlangıcından ilk yanıt kaydının görelî konumu.

Bu, MQOD yapısının başlangıcından itibaren ilk MQRR yanıt kaydının bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir. *ODRRO* yalnızca bir dağıtım listesi açıldığında kullanılır. *ODREC* sıfırsa alan yoksayıdır.

Bir dağıtım listesi açıldığında, açılmayan kuyrukları (MQRR 'dekiRRCC alanı) ve her başarısızlığın nedenini (MQRR 'dekiRRREA alanı) saptamak için bir ya da daha çok MQRR yanıt kaydı dizisi sağlanabilir. Veriler, yanıt kayıtları dizisinde, nesne kayıtları dizisindeki kuyruk adlarıyla aynı sırayla döndürülür. Kuyruk yöneticisi yanıt kayıtlarını yalnızca çağrı sonucu karışık olduğunda (yani, bazı kuyruklar başarılı bir şekilde açıldığında diğerleri başarısız olduğunda ya da tümü farklı nedenlerle başarısız olduğunda) ayarlar; çağrıdaki RC2136 neden kodu bu durumu gösterir. Aynı neden kodu tüm

kuyruklar için de geçerliyse, MÇOPEN ya da MÇPUT1 çağrısının **REASON** değiştirilmesinde bu neden döndürülür ve yanıt kayıtları ayarlanmaz. Yanıt kayıtları isteğe bağlıdır, ancak bunlar sağlanırsa *ODREC* olmalıdır.

Yanıt kayıtları, *ODRRO* içinde bir görelî konum belirtilerek ya da *ODRRP* içinde bir adres belirtilerek nesne kayıtlarıyla aynı şekilde sağlanabilir; Bunun nasıl yapılacağına ilişkin ayrıntılar için önceki *ODORO* açıklamasına bakın. Ancak, en çok *ODRRO* ve *ODRRP* kullanılabilir; her ikisi de sıfır değilse, çağrı RC2156 neden koduyla başarısız olur.

MÇPUT1 çağrısına ilişkin bu yanıt kayıtları, ileti dağıtım listesindeki kuyruklara gönderildiğinde ortaya çıkan hatalarla ilgili bilgileri ve kuyruklar açıldığında oluşan hataları döndürmek için kullanılır. Bir kuyruk için koyma işlemindeki tamamlanma kodu ve neden kodu, yalnızca kuyruktaki tamamlanma kodu CCOK ya da CCWARN ise, o kuyruk için açık işlemdekileri değiştirir.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir. *ODVER* *ODVER2* değerinden küçükse bu alan yoksayılr.

ODRRP (işaretçi)

İlk yanıt kaydının adresi.

Bu, ilk MÇRR yanıt kaydının adresidir. *ODRRP* yalnızca bir dağıtım listesi açıldığında kullanılır. *ODREC* sıfırsa alan yoksayılr.

ODRRP ya da *ODRRO* yanıt kayıtlarını belirtmek için kullanılabilir, ancak her ikisini birden belirtmez; ayrıntılar için *ODRRO* alanının önceki açıklamasına bakın. *ODRRP* kullanılmazsa, boş değerli göstergeye ya da boş değerli byte 'lara ayarlanmalıdır.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş işaretçidir. *ODVER* *ODVER2* değerinden küçükse bu alan yoksayılr.

ODSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

ODSIDV

Nesne tanımlayıcı yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri *ODSIDV* 'dir.

ODSS (MÇCHARV)

ODSS, kuyruktan ileti alınırken kullanılan seçim ölçütlerini sağlamak için kullanılan dizgiyi içerir.

ODSS aşağıdaki durumlarda sağlanmamalıdır:

- *ODOT* OTQ değilse
- Açılmakta olan kuyruk giriş seçeneklerinden biri kullanılarak açılmıyorsa, *OOINP**

Bu durumlarda *ODSS* sağlanırsa, arama RC2516neden koduyla başarısız olur.

ODSS yanlış belirtilirse, *MÇCHARV* yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı RC2519neden koduyla başarısız olur. *ODVER* *ODVER4* değerinden küçükse bu alan yoksayılr.

ODUDC (10 basamaklı işaretli tamsayı)

Başarıyla açılan uzak kuyruk sayısı

Bu, dağıtım listesindeki uzak kuyruklara çözülecek ve başarıyla açılacak kuyrukların sayısıdır. Varsa, dağıtım listesinde olmayan tek bir kuyruk açılırken de bu alan ayarlanır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dir. *ODVER* *ODVER2* değerinden küçükse bu alan yoksayılr.

ODVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer aşağıdakilerden biri olmalıdır:

ODVER1

Version-1 nesne tanımlayıcı yapısı.

ODVER2

Version-2 nesne tanımlayıcı yapısı.

ODVER3

Version-3 nesne tanımlayıcı yapısı.

ODVER4

Version-4 nesne tanımlayıcı yapısı.

Yalnızca yapının daha yeni sürümlerinde bulunan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

ODVERC

Nesne tanımlayıcı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri ODVER1' dir.

Başlangıçtaki değerler

Çizelge 713. MÇOD içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
ODSID	ODSIDV	'OD---
ODVER	ODVER1	1
ODOT	OTQ	1
ODON	Yok	Boşluklar
ODMN	Yok	Boşluklar
ODDN	Yok	'AMQ.*'
ODAU	Yok	Boşluklar
ODREC	Yok	0
ODKDC	Yok	0
ODUDC	Yok	0
ODIDC	Yok	0
ODORO	Yok	0
ODRRO	Yok	0
ODORP	Yok	Boş değerli gösterge ya da boş değerli byte
ODRRP	Yok	Boş değerli gösterge ya da boş değerli byte
ODASI	SINONE.	Boş Değerler
ODRQN	Yok	Boşluklar
ODRMN	Yok	Boşluklar
ODOS	MÇCHARV için tanımlandığı şekilde	MÇCHARV için tanımlandığı şekilde
ODRO	ODOS içinde sağlandığı gibi	ODOS içinde sağlandığı gibi
ODSS	Yok	Boşluklar

Çizelge 713. MÇOD içindeki alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
Notlar:		
1. - simgesi, tek bir boş karakteri temsil eder.		

RPG bildirim

```

D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MÇOD Structure
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 MÇOD
D ODORO          185    188I 0 INZ(0)
D*
** Offset of first response record
D* from start of MÇOD
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                                00000000000000')
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 IBM i üzerinde MQOR (Nesne kaydı)

MQOR yapısı, tek bir hedef kuyruğun kuyruk adını ve kuyruk yöneticisi adını belirtmek için kullanılır.

Genel Bakış

Amaç: MQOR, MQOPEN ve MQPUT1 çağrılarına ilişkin bir giriş yapısıdır.

Karakter kümesi ve kodlama: MQOR içindeki veriler, ENNAT tarafından verilen yerel kuyruk yöneticisinin **CodedCharSetId** kuyruk yöneticisi özneliği ve kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Kullanım: MQOPEN çağrısında bu yapılardan oluşan bir dizi sağlanarak, bir kuyruk listesi açılabilir; bu listeye *dağıtım listesi* adı verilir. Kuyruk başarıyla açılırsa, MQOPEN çağrısının döndürdüğü kuyruk tanıtıcısı kullanılarak konan her ileti, listedeki kuyrukların her birine yerleştirilir.

- [“Alanlar” sayfa 1150](#)
- [“Başlangıçtaki değerler” sayfa 1151](#)
- [“RPG bildirimi” sayfa 1151](#)

Alanlar

MQOR yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırad**a açıklanır:

ORMN (48 baytlık karakter dizgisi)

Nesne kuyruğu yöneticisi adı.

Bu, MÇOD yapısındaki *ODMN* alanıyla aynıdır (ayrıntılar için MÇOD ' ye bakın).

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri 48 boş karakterdir.

ORON (48 baytlık karakter dizgisi)

Nesne adı.

Bu, aşağıdaki durumlar dışında, MÇOD yapısındaki *ODON* alanıyla aynıdır (ayrıntılar için MÇOD ' ye bakın):

- Bir kuyruğun adı olmalıdır.
- Bir model kuyruğunun adı olmamalıdır.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri 48 boş karakterdir.

Başlangıçtaki değerler

Çizelge 714. MÇOR içindeki alanlar		
Alan adı	Değişmez adı	Değişmez değeri
<i>ORON</i>	Yok	Boşluklar
<i>ORMN</i>	Yok	Boşluklar

RPG bildirim

```
D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MÇOR Structure
D*
D* Object name
D  ORON                1      48  INZ
D* Object queue manager name
D  ORMN                49     96  INZ
```

MÇPD-Özellik tanımlayıcısı

MÇPD , bir özelliğin özniteliklerini tanımlamak için kullanılır.

Genel Bakış

Amaç: Yapı, MÇSETMP çağrısında bir giriş/çıkış değiştirgesi ve MÇINÇMP çağrısında bir çıkış değiştirgesidir.

Karakter kümesi ve kodlama: MÇPD ' deki veriler, uygulamanın karakter kümesinde ve kodlamasında (ENNAT) yer almalıdır.

- [“Alanlar” sayfa 1151](#)
- [“Başlangıçtaki değerler” sayfa 1154](#)
- [“RPG bildirim” sayfa 1154](#)

Alanlar

MÇPD yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

PDCT (10 basamaklı işaretli tamsayı)

Bu, özelliğin hangi ileti bağlamına ait olduğunu açıklar.

Bir kuyruk yöneticisi, kuyruk yöneticisinin hatalı olarak algıladığı IBM MQtanımlı bir özellik içeren bir ileti aldığıında. Kuyruk yöneticisi, *PDCT* alanının değerini düzeltir.

Aşağıdaki seçenek belirtilebilir:

PDUSC

Özellik, kullanıcı bağlamıyla ilişkilendirildi.

MQSETMP çağrısını kullanarak kullanıcı bağlamıyla ilişkili bir özellik ayarlayabilmek için özel bir yetkilendirme gerekmez.

Daha önce açıklanan seçenek gerekli değilse, aşağıdaki seçenek kullanılabilir:

PDNOC

Özellik bir ileti bağlamıyla ilişkilendirilmemiş.

Tanınmayan bir değer, *PDREA* kodu RC2482 ile reddedilir.

MQSETMP çağrısı için bir giriş/çıkış alanı ve MQINQMP çağrısından bir çıkış alanıdır. Bu alanın ilk değeri PDNOC ' dur.

PCPYOPT (10 basamaklı işaretli tamsayı)

Bu, özelliğin hangi tipte iletilere kopyalanması gerektiğini açıklar.

Bu yalnızca tanınan IBM MQtanımlı özellikler için bir çıkış alanıdır; IBM MQ uygun değeri ayarlar.

Bir kuyruk yöneticisi, kuyruk yöneticisinin hatalı olarak algıladığı IBM MQtanımlı bir özellik içeren bir ileti aldığıında. Kuyruk yöneticisi, *CopyOptions* alanının değerini düzeltir.

Bu seçeneklerden birini ya da birkaçını belirtebilirsiniz. birden çok seçenek belirtmek için değerleri bir araya ekleyin (aynı sabiti bir kereden fazla eklemeyin) ya da bit bakımından OR işlemini kullanarak değerleri birleştirin (programlama dili bit işlemlerini destekliyse).

KOPYALAMA

Bu özellik, iletilmekte olan bir iletiye kopyalanır.

COPPUB

Bu özellik, bir ileti yayınlanırken abone tarafından alınan iletiye kopyalanır.

COPREP (KOP)

Bu özellik bir yanıt iletisine kopyalanır.

COPRP (ÇEVİRİ)

Bu özellik bir rapor iletisine kopyalanır.

KOPYALA

Bu özellik, sonraki iletilerin tüm tiplerine kopyalanır.

KOPYALAYIN

Bu özellik bir iletiye kopyalanmaz.

Varsayılan seçenek: Varsayılan kopyalama seçenekleri kümesini sağlamak için aşağıdaki seçenek belirtilebilir:

KOPYALA

Bu özellik, iletilen bir iletiye, bir rapor iletisine ya da bir ileti yayınlanırken abone tarafından alınan bir iletiye kopyalanır.

Bu, COPFOR, artı COPRP ve COPPUB seçeneklerinin birleşimini belirtmeye eşdeğerdir.

Daha önce açıklanan seçeneklerden hiçbiri gerekmiyorsa, aşağıdaki seçeneği kullanın:

KOPYALAYIN

Başka bir kopyalama seçeneği belirtilmediğini belirtmek için bu değeri kullanın; programsal olarak bu özellik ile sonraki iletiler arasında ilişki yoktur. Bu, ileti tanımlayıcı özellikleri için her zaman döndürülür.

MQSETMP çağrısı için bir giriş/çıkış alanı ve MQINQMP çağrısından bir çıkış alanıdır. Bu alanın ilk değeri COPDEF 'dir.

PDOPT (10 basamaklı işaretli tamsayı)

Değer şöyle olmalıdır:

PDNONE

Seçenek belirtilmedi

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri PDNONE.

PDSID (10 basamaklı işaretli tamsayı)

Bu yapı tanıttıcısıdır; değer şu olmalıdır:

PSIDV

Özellik tanımlayıcı yapısına ilişkin tanıttıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **PSIDV**.

PDSUP (10 basamaklı işaretli tamsayı)

Bu alanda, bu özelliği içeren iletinin kuyruğa konması için kuyruk yöneticisinin ileti özelliği için hangi destek düzeyinin gerekli olduğu açıklanır. Bu yalnızca IBM MQtanımlı özellikler için geçerlidir; diğer tüm özellikler için destek isteğe bağlıdır.

IBM MQtanımlı özellik kuyruk yöneticisi tarafından bilindiğinde alan otomatik olarak doğru değere ayarlanır. Özellik tanınmazsa, PDSUPO atanır. Bir kuyruk yöneticisi, kuyruk yöneticisinin hatalı olarak algıladığı IBM MQtanımlı bir özellik içeren bir ileti aldığı anda. Kuyruk yöneticisi, *PDSUP* alanının değerini düzeltir.

CMNOVA seçeneğinin ayarlandığı ileti tanıttıcısında MQSETMP çağrısı kullanılarak IBM MQtanımlı bir özellik ayarlanırken *PDSUP* bir giriş alanı olur. Bu, bir uygulamanın, özelliğin bağlı kuyruk yöneticisi tarafından desteklenmediği, ancak iletinin başka bir kuyruk yöneticisinde işlenmek üzere tasarlandığı, doğru değere sahip bir IBM MQtanımlı özellik koymasına olanak sağlar.

PDSUPO değeri her zaman IBM MQtanımlı olmayan özelliklere atanır.

Aşağıdaki değerlerden biri MQINQMP çağrısı tarafından döndürülür ya da CMNOVA seçeneğinin ayarlandığı bir ileti tanıttıcısında MQSETMP çağrısı kullanılırken değerlerden biri belirtilebilir:

PDSUPO (PDSUPO)

Özellik desteklenmese de bir kuyruk yöneticisi tarafından kabul edilir. İletinin ileti özelliklerini desteklemeyen bir kuyruk yöneticisine akması için özellik atılabilir. Bu değer, IBM MQtanımlı olmayan özelliklere de atanır.

PDSUPR

Özellik için destek gereklidir. İleti, IBM MQtanımlı özelliği desteklemeyen bir kuyruk yöneticisi tarafından reddedildi. MQPUT ya da MQPUT1 çağrısı, tamamlanma kodu CCFAIL ve neden kodu RC2490 ile başarısız oluyor.

PDSUPL

İleti yerel bir kuyruğa gönderildiyse, ileti IBM MQtanımlı özelliği desteklemeyen bir kuyruk yöneticisi tarafından reddedilir. MQPUT ya da MQPUT1 çağrısı, tamamlanma kodu CCFAIL ve neden kodu RC2490 ile başarısız oluyor.

İleti uzak bir kuyruk yöneticisine gönderildiyse, MQPUT ya da MQPUT1 çağrısı başarılı olur.

İleti tanıttıcısı CMNOVA seçeneği kümesiyle yaratıldıysa, MQINQMP çağrısında bir çıkış alanı ve MQSETMP çağrısında bir giriş alanıdır. Bu alanın ilk değeri PDSUPO ' dur.

PDVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

PDVER1

Version-1 özellik tanımlayıcı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

PDVERC

Özellik tanımlayıcı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **PDVER1**.

Başlangıçtaki değerler

Çizelge 715. MQPD içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
PDSID	PDSIDV	'PD'
PDVER	PDVER1	1
PDOPT	PDNONE	0
PDSUP	PDSUPO (PDSUPO	0
PDCT	PDNOC	0
PDCPYOPT	KOPYALA	0

RPG bildirim

```
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

IBM i üzerinde MQPMO (Put-message options)

MQPMO yapısı, uygulamanın iletilerin kuyruklara nasıl yerleştirileceğini ya da konulara nasıl yayınlanacağını denetleyen seçenekleri belirtmesini sağlar.

Genel Bakış

Amaç

Yapı, MQPUT ve MQPUT1 çağrılarında bir giriş/çıkış değiştirgesidir.

Sürüm

MQPMO ' nun yürürlükteki sürümü: PMVER2. Yapının yalnızca daha yeni sürümlerinde bulunan alanlar, izleyen açıklamalarda bu şekilde tanımlanır.

Sağlanan COPY dosyası, ortam tarafından desteklenen, ancak *PMVER* alanının ilk değeri *PMVER1* olarak ayarlanmış MQPMO ' nun en son sürümünü içerir. *version-1* yapısında var olmayan alanları kullanmak için uygulamanın *PMVER* alanını gerekli sürümün sürüm numarasına ayarlaması gerekir.

Karakter kümesi ve kodlama

MQPMO içindeki veriler, ENNAT tarafından verilen yerel kuyruk yöneticisinin **CodedCharSetId** kuyruk yöneticisi özneliği ve kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

- “Alanlar” sayfa 1155
- “Başlangıçtaki değerler” sayfa 1167

- “RPG bildirimi” sayfa 1168

Alanlar

MQPMO yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

PMCT (10 basamaklı işaretli tamsayı)

Giriş kuyruğunun nesne tanıtıcısı.

PMPASI ya da PMPASA belirtilirse, bu alan, konmakta olan iletiyle ilişkilendirilecek bağlam bilgilerinin alınacağı giriş kuyruğu tanıtıcısını içermelidir.

PPMPASI ve PMPASA belirtilmezse, bu alan yoksayılr.

Bu bir giriş alanıdır. Bu alanın ilk değeri şudur: 0.

PMIDC (10 basamaklı işaretli tamsayı)

Gönderilemeyen iletilerin sayısı.

Bu, dağıtım listesindeki kuyruklara gönderilemeyen iletilerin sayısıdır. Sayı, açılmayan kuyrukları ve başarıyla açılan, ancak koyma işleminin başarısız olduğu kuyrukları içeriyor. Bu alan, dağıtım listesinde olmayan tek bir kuyruğa ileti yerleştirilirken de ayarlanır.

Not: Bu alan yalnızca MQPUT ya da MQPUT1 çağrısındaki **CMPCOD** değiştirgesi CCOK ya da CCWARN ise ayarlanır; **CMPCOD** değiştirgesi CCFAIL ise ayarlanmaz.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır. *PMVER* , *PMVER2*değerinden küçükse bu alan ayarlanmaz.

PMKDC (10 basamaklı işaretli tamsayı)

Yerel kuyruklara başarıyla gönderilen iletilerin sayısı.

Bu, yürürlükteki MQPUT ya da MQPUT1 çağrısının yerel kuyruklar olan dağıtım listesindeki kuyruklara başarıyla gönderdiği iletilerin sayısıdır. Bu sayı, uzak kuyruklara çözülecek kuyruklara gönderilen iletileri içermez (iletiyi saklamak için başlangıçta yerel bir iletim kuyruğu kullanılsa bile). Bu alan, dağıtım listesinde olmayan tek bir kuyruğa ileti yerleştirilirken de ayarlanır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır. *PMVER* , *PMVER2*değerinden küçükse bu alan ayarlanmaz.

PMOPT (10 basamaklı işaretli tamsayı)

MQPUT ve MQPUT1eylemine denetleyen seçenekler.

Aşağıdakilerden herhangi biri ya da hiçbiri belirtilemez. Birden çok değer gerekiyorsa, değerler eklenebilir (aynı değişmezi bir kereden fazla eklemeyin). Geçerli olmayan birleşimler belirtilmiştir; diğer birleşimler geçerlidir.

Yayınlama seçenekleri: Aşağıdaki seçenekler, iletilerin bir konuya yayınlama şeklini denetler.

PMSRTO

Bu yayındaki MQMD ' nin MDRQ ve MDRM alanlarına doldurulan bilgiler abonelere iletilmez. Bu seçenek, ReplyToQ gerektiren bir rapor seçeneğiyle kullanılırsa, arama RC2027 ile başarısız olur.

PMRET

Gönderilmekte olan yayın, kuyruk yöneticisi tarafından alıkonacak. Bu, bir abonenin MQSUBRQ çağrısıyla yayımlandıktan sonra bu yayının bir kopyasını istemesini sağlar. Ayrıca, SONEWP seçeneği kullanılarak gönderilmemeyi seçmedikleri sürece, bu yayın yapıldıktan sonra aboneliklerini yapan uygulamalara bir yayın gönderilmesine izin verir. Bir uygulamaya alıkonan bir yayın gönderilirse, bu, o yayının mq.IsRetained ileti özelliğiyle gösterilir.

Konu ağacının her düğümünde yalnızca bir yayın korunabilir. Başka bir deyişle, bu konu için başka bir uygulama tarafından yayınlanan bir alıkonan yayın varsa, bu yayın bu yayınlı değiştirilir. Bu nedenle, aynı konuda birden fazla yayıncının ileti tutmasını önlemek daha iyidir.

Alıkonan yayınlar bir abone tarafından istendiğinde, kullanılan abonelik konuda genel arama karakteri içerebilir; bu durumda, alıkonan yayınların sayısı (konu ağacındaki çeşitli düğümlerde) eşleşebilir ve istekte bulunan uygulamaya birkaç yayın gönderilebilir. Daha fazla ayrıntı için [“MQSUBRQ-Abonelik isteği” sayfa 786](#) aramasının açıklamasına bakın.

Bu seçenek kullanılırsa ve yayın alıkonamazsa, ileti yayınlanmaz ve çağrı RC2479 ile başarısız olur.

Eşitleme noktası seçenekleri: Aşağıdaki seçenekler, bir iş birimi içindeki MQPUT ya da MQPUT1 çağrısının katılımıyla ilgilidir:

PMSYP

İletiyi syncpoint denetimine koyun.

İstek, normal iş birimi protokolleri içinde çalışmaktadır. İleti, iş birimi kesinleştirilinceye kadar iş biriminin dışında görünmez. İş birimi geriletildiyse, ileti silinir.

Bu seçenek ve PMNSYP belirtilmezse, koyma isteği bir iş birimi içinde değildir.

PMSYP, PMNSYP ile birlikte belirtilmemelidir.

PMNSYP

İletiyi eşitleme noktası denetimi olmadan koyun.

İstek, normal iş birimi protokollerinin dışında çalışmaktadır. İleti hemen kullanılabilir ve bir iş birimi yedeklenerek silinemez.

Bu seçenek ve PMSYP belirtilmezse, koyma isteği bir iş birimi içinde değildir.

PMNSYP, PMSYP ile birlikte belirtilmemelidir.

İleti tanıtıcısı ve ilinti tanıtıcısı seçenekleri: Aşağıdaki seçenekler, kuyruk yöneticisinden yeni bir ileti tanıtıcısı ya da ilinti tanıtıcısı oluşturmasını ister:

PMNMID

Yeni bir ileti tanıtıcısı oluşturun.

Bu seçenek, kuyruk yöneticisinin MQMD ' deki *MDMID* alanının içeriğini yeni bir ileti tanıtıcısıyla değiştirmesine neden olur. Bu ileti tanıtıcısı iletiyle birlikte gönderilir ve MQPUT ya da MQPUT1 çağrısının çıkışında uygulamaya döndürülür.

Bu seçenek, ileti bir dağıtım listesine konurken de belirtilebilir; ayrıntılar için MQPMR yapısındaki *PRMID* alanının tanımına bakın.

Bu seçeneğin kullanılması, *MDMID* alanını her MQPUT ya da MQPUT1 çağrısının öncesinde MINONE olarak sıfırlama gereksinmesini ortadan kaldırır.

PMNCID

Yeni bir ilinti tanıtıcısı oluşturun.

Bu seçenek, kuyruk yöneticisinin MQMD ' deki *MDCID* alanının içeriğini yeni bir ilinti tanıtıcısıyla değiştirmesine neden olur. Bu ilinti tanıtıcısı iletiyle birlikte gönderilir ve MQPUT ya da MQPUT1 çağrısının çıkışında uygulamaya döndürülür.

Bu seçenek, ileti bir dağıtım listesine konurken de belirtilebilir; ayrıntılar için MQPMR yapısındaki *PRCID* alanının tanımına bakın.

PMNCID, uygulamanın benzersiz bir ilinti tanıtıcısı gerektirdiği durumlarda kullanışlıdır.

Grup ve bölüm seçenekleri: Aşağıdaki seçenek, iletilerin gruplar ve mantıksal ileti kesimleri halinde işlenmesiyle ilgilidir. Bu tanımlar, aşağıdaki seçeneğin anlaşılmasına yardımcı olabilir:

Fiziksel ileti

Bu, bir kuyruğa yerleştirilebilecek ya da kuyruktan kaldırılabilir en küçük bilgi birimidir; genellikle tek bir MQPUT, MQPUT1 ya da MQGET çağrısında belirtilen ya da alınan bilgilere karşılık gelir. Her fiziksel iletinin kendi ileti tanımlayıcısı (MQMD) vardır. Genellikle, fiziksel iletiler ileti tanıtıcısı (MQMD ' deki *MDMID* alanı) için farklı değerlerle ayırt edilir, ancak bu kuyruk yöneticisi tarafından zorlanmaz.

Mantıksal ileti

Bu, tek bir uygulama bilgisi birimidir. Sistem kısıtlamaları olmadığında, mantıksal bir ileti fiziksel bir iletiyle aynı olur. Ancak, mantıksal iletilerin büyük olduğu durumlarda, sistem kısıtlamaları mantıksal bir iletinin *segment* adı verilen iki ya da daha fazla fiziksel iletiye bölünmesini önerebilir ya da gerekli kılabilir.

Bölümlenmiş bir mantıksal ileti, boş olmayan grup tanıtıcısı (MQMD 'deMDGID alanı) aynı olan iki ya da daha çok fiziksel iletiden ve aynı ileti sıra numarasından (MQMD' deMDSEQ alanı) oluşur. Kesimler, mantıksal iletidaki verilerin başlangıcından fiziksel iletidaki verilerin görelî konumunu veren kesim görelî konumu (MQMD ' deMDOFF alanı) için farklı değerlerle ayırt edilir. Her bir bölüm fiziksel bir ileti olduğu için, mantıksal bir iletidaki bölümlerin genellikle farklı ileti tanıtıcıları vardır.

Bölümlenmemiş, ancak bölümlenmeye gönderen uygulama tarafından izin verilen mantıksal bir ileti, boş olmayan bir grup tanıtıcısına da sahiptir; ancak bu durumda, mantıksal ileti bir ileti grubuna ait değilse, bu grup tanıtıcısına sahip yalnızca bir fiziksel ileti vardır. Mantıksal ileti bir ileti grubuna ait değilse, bölümlenme gönderen uygulama tarafından engellenen mantıksal iletilerin boş değerli bir grup tanıtıcısı (GINONE) vardır.

İleti grubu

Bu, aynı boş olmayan grup tanıtıcısına sahip bir ya da daha fazla mantıksal iletiden oluşan bir kümedir. Gruptaki mantıksal iletiler, 1-n aralığında bir tamsayı olan ileti sıra numarasına ilişkin farklı değerlerle ayırt edilir; burada n, gruptaki mantıksal ileti sayısıdır. Mantıksal iletilerden biri ya da daha fazlası bölümlenmişse, grupta n 'den fazla fiziksel ileti vardır.

PMLOGO

Mantıksal ileti grupları ve kesimleri içindeki iletiler mantıksal sıraya konmuştur.

Bu seçenek, kuyruk yöneticisine uygulamanın iletileri gruplar ve mantıksal ileti kesimleri halinde nasıl yerleştirdiğini bildirir. Yalnızca MQPUT çağrısında belirtilebilir; MQPUT1 çağrısında geçerli değildir.

PMLOGO belirtilirse, uygulamanın aşağıdaki öğeler için ardışık MQPUT çağrıları kullandığını gösterir:

- Her mantıksal iletidaki kesimleri, 0 'dan başlayarak, aralıksız olarak, kesim görelî konumunun artma sırasına göre koyun.
- Bölümleri sonraki mantıksal iletiye yerleştirmeden önce tüm bölümleri tek bir mantıksal iletiye koyun.
- Her ileti grubundaki mantıksal iletileri, 1 'den başlayarak, aralıksız artan ileti sıra numarası sırasına göre koyun.
- Mantıksal iletileri sonraki ileti grubuna koymadan önce tüm mantıksal iletileri bir ileti grubuna koyun.

Bu sıraya "mantıksal düzen" denir.

Uygulama kuyruk yöneticisine iletileri mantıksal ileti gruplarına ve bölümlerine nasıl yerleştirdiğini belirttiği için, kuyruk yöneticisi bunu yaptığı için uygulamanın her MQPUT çağrısına ilişkin grup ve bölüm bilgilerini koruması ve güncellemesi gerekmez. Özellikle, kuyruk yöneticisi bunları uygun değerlere ayarladığı için, uygulamanın MQMD 'deki MDGID, MDSEQ ve MDOFF alanlarını ayarlamasına gerek olmadığı anlamına gelir. Uygulamanın yalnızca MQMD 'deki MDMFL alanını ayarlayıp iletilerin ne zaman gruplara ait olduğunu ya da mantıksal ileti kesimleri olduğunu göstermesi ve mantıksal iletinin bir grubundaki ya da son bölümündeki son iletiyi belirtmesi gerekir.

Bir ileti grubu ya da mantıksal ileti başlatıldıktan sonra, sonraki MQPUT çağrıları MQMD 'deki MDMFL içinde uygun MF* işaretlerini belirtmelidir. Uygulama, sonlandırılmamış bir ileti grubu olduğunda bir iletiyi gruba yerleştirmeyi denerse ya da sonlandırılmamış bir mantıksal ileti olduğunda kesim olmayan bir ileti yerleştirirse, çağrı RC2241 ya da RC2242 neden koduyla başarısız olur. Ancak, kuyruk yöneticisi yürürlükteki ileti grubuna ya da yürürlükteki mantıksal iletiye ilişkin bilgileri saklar ve uygulama, MFLMIG ya da MFLSEG komutlarını uygun şekilde belirterek (büyük olasılıkla uygulama ileti verisi olmadan), MQPUT çağrısıyla grup içinde olmayan ya da bölüm olmayan iletiyi yeniden yayınlamadan önce bunları sonlandırabilir.

Çizelge 716 sayfa 1158 , geçerli seçenek ve işaret birleşimlerini ve kuyruk yöneticisinin her durumda kullandığı *MDGID*, *MDSEQ*ve *MDOFF* alanlarının değerlerini gösterir. Tabloda gösterilmeyen seçenek ve işaret birleşimleri geçerli değil. Çizelgedeki kolonlar aşağıdaki anlamlara sahiptir:

Günlük ORDı

Çağrıda PMLOGO seçeneğinin belirtilip belirtilmediğini belirtir.

MIG

Çağrıda MFMIG ya da MFLMIG seçeneğinin belirlenip belirlenmediğini belirtir.

SİZ

Çağrıda MFSEG ya da MFLSEG seçeneğinin belirlenip belirlenmediğini belirtir.

SEG OK (TAMAM)

Çağrıda MFSEGA seçeneğinin belirtilip belirtilmediğini belirtir.

Cur grp

Aramadan önce geçerli bir ileti grubunun var olup olmadığını belirtir.

Cur günlük iletisi

Çağrıdan önce geçerli bir mantıksal iletinin bulunup bulunmadığını belirtir.

Diğer sütunlar

Kuyruk yöneticisinin kullandığı değerleri gösterir. "Önceki", kuyruk tanıtıcısı için önceki iletide alan için kullanılan değeri belirtir.

PMRLOC

MQPPO yapısındaki PMRQN ' nin, iletinin gerçekten konacağı yerel kuyruğun adıyla tamamlanması gerektiğini belirtir. ResolvedQMGrAdı, yerel kuyruğu barındıran yerel kuyruk yöneticisinin adıyla benzer şekilde tamamlanır. Bunun ne anlama geldiğini görmek için OORLOQ ' ya bakın. Bir kullanıcının kuyruğa koyma işlemi için yetkisi varsa, kullanıcının MQPUT çağrısında bu işareti belirtmek için gereken yetkisi vardır. Özel bir yetkiye gerek yoktur.

Çizelge 716. Mantıksal ileti grupları ve bölümlerindeki iletilerle ilgili MQPUT seçenekleri								
Belirlediğiniz seçenekler				Çağrıdan önce grup ve günlük-ileti durumu		Kuyruk yöneticisinin kullandığı değerler		
Günlük ORDA Sı	MIG	SANI YE	TAMA M	Cur grp	Cur günlük iletisi	MDGID	MDSEQ	MDOFF
Evet	Hayır	Hayır	Hayır	Hayır	Hayır	GINONE	1	0
Evet	Hayır	Hayır	Evet	Hayır	Hayır	Yeni grup tanıtıcısı	1	0
Evet	Hayır	Evet	Evet ya da Hayır	Hayır	Hayır	Yeni grup tanıtıcısı	1	0
Evet	Hayır	Evet	Evet ya da Hayır	Hayır	Evet	Önceki grup tanıtıcısı	1	Önceki görel konum + önceki kesim uzunluğu
Evet	Evet	Evet ya da Hayır	Evet ya da Hayır	Hayır	Hayır	Yeni grup tanıtıcısı	1	0
Evet	Evet	Evet ya da Hayır	Evet ya da Hayır	Evet	Hayır	Önceki grup tanıtıcısı	Önceki sıra numarası + 1	0

Çizelge 716. Mantıksal ileti grupları ve bölümlerindeki iletilerle ilgili MQPUT seçenekleri (devamı var)

Belirlediğiniz seçenekler				Çağrıdan önce grup ve günlük-ileti durumu		Kuyruk yöneticisinin kullandığı değerler		
Evet	Evet	Evet	Evet ya da Hayır	Evet	Evet	Önceki grup tanıtıcısı	Önceki sıra numarası	Önceki görel konum + önceki kesim uzunluğu
Hayır	Hayır	Hayır	Hayır	Evet ya da Hayır	Evet ya da Hayır	GINONE	1	0
Hayır	Hayır	Hayır	Evet	Evet ya da Hayır	Evet ya da Hayır	GINONE ise yeni grup tanıtıcısı, alanda else değeri	1	0
Hayır	Hayır	Evet	Evet ya da Hayır	Evet ya da Hayır	Evet ya da Hayır	GINONE ise yeni grup tanıtıcısı, alanda else değeri	1	Alandaki değer
Hayır	Evet	Hayır	Evet ya da Hayır	Evet ya da Hayır	Evet ya da Hayır	GINONE ise yeni grup tanıtıcısı, alanda else değeri	Alandaki değer	0
Hayır	Evet	Evet	Evet ya da Hayır	Evet ya da Hayır	Evet ya da Hayır	GINONE ise yeni grup tanıtıcısı, alanda else değeri	Alandaki değer	Alandaki değer

Not:

- PMLOGO, MQPUT1 çağrısında geçerli değil.
- *MDMID* alanı için, *PMNMID* ya da *MINONE* belirtildiyse kuyruk yöneticisi yeni bir ileti tanıtıcısı oluşturur; tersi durumda, alandaki değeri kullanır.
- *MDCID* alanı için, *PMNCID* belirtilirse kuyruk yöneticisi yeni bir ilinti tanıtıcısı oluşturur; tersi durumda, alandaki değeri kullanır.

PMLOGO belirtildiğinde, kuyruk yöneticisi, bir gruptaki ve bir mantıksal iletideki bölümlerdeki tüm iletilerin MQMD 'deki *MDPER* alanına aynı değerle yerleştirilmesini, yani tümünün kalıcı olmasını ya da tümünün kalıcı olmamasını gerektirir. Bu koşul karşılanmazsa, MQPUT çağrısı RC2185 neden koduyla başarısız olur.

PMLOGO seçeneği iş birimlerini aşağıdaki gibi etkiler:

- Bir gruptaki ya da mantıksal iletideki ilk fiziksel ileti bir iş birimine yerleştirilirse, aynı kuyruk tanıtıcısı kullanılıyorsa, gruptaki ya da mantıksal iletideki diğer tüm fiziksel iletilerin bir iş birimine yerleştirilmesi gerekir. Ancak, aynı iş birimine konmaları gerekmez. Bu, birçok fiziksel iletiden oluşan bir ileti grubunun ya da mantıksal iletinin, kuyruk tanıtıcısı için art arda iki ya da daha fazla iş birimine bölünmesini sağlar.
- Bir grup ya da mantıksal iletideki ilk fiziksel ileti bir iş birimine yerleştirilmezse, aynı kuyruk tanıtıcısı kullanılırsa, gruptaki ya da mantıksal iletideki diğer fiziksel iletilerin hiçbiri bir iş birimine yerleştirilemez.

Bu koşullar karşılanmazsa, MQPUT çağrısı RC2245 neden koduyla başarısız olur.

PMLOGO belirtildiğinde, MQPUT çağrısında sağlanan MQMD, MDVER2değerinden küçük olmamalıdır. Bu koşul karşılanmazsa, arama RC2257 neden koduyla başarısız olur.

PMLOGO belirtilmezse, gruplar ve mantıksal ileti bölümlerindeki iletiler herhangi bir sıraya konabilir ve tam ileti grupları ya da tam mantıksal iletiler konması gerekmez. *MDGID*, *MDSEQ*,

MDOFF ve *MDMFL* alanlarının uygun değerlere sahip olduğundan emin olmak uygulamanın sorumluluğundadır.

Bu, bir sistem arızası oluşuktan sonra ortada bir ileti grubunu ya da mantıksal iletiyi yeniden başlatmak için kullanılabilir bir tekniktir. Sistem yeniden başlatıldığında, uygulama *MDGID*, *MDSEQ*, *MDOFF*, *MDMFL* ve *MDPER* alanlarını uygun değerlere ayarlayabilir ve daha sonra, *PMSYP* ya da *PMNSYP* için gerekli olarak ayarlanmış *MQPUT* çağrısına *PMLOGO* belirtmeden devam edebilir. Bu çağrı başarılı olursa, kuyruk yöneticisi grup ve bölüm bilgilerini korur ve bu kuyruk tanıtıcısını kullanan sonraki *MQPUT* çağrıları *PMLOGO* ' yı normal olarak belirtebilir.

Kuyruk yöneticisinin *MQPUT* çağrısı için sakladığı grup ve bölüm bilgileri, *MQGET* çağrısı için sakladığı grup ve bölüm bilgilerinden ayrıdır.

Belirli bir kuyruk tanıtıcısı için, uygulama *PMLOGO* ' u belirtmeyen *MQPUT* çağrılarıyla *MQPUT* çağrılarını karışık olarak kullanmakta serbesttir, ancak aşağıdaki noktalar dikkate alınmalıdır:

- *PMLOGO* belirtilmezse, her başarılı *MQPUT* çağrısı kuyruk yöneticisinin kuyruk tanıtıcısı için grup ve bölüm bilgilerini uygulama tarafından belirlenen değerlere ayarlamasına neden olur; bu, kuyruk tanıtıcısı için kuyruk yöneticisi tarafından tutulan var olan grup ve bölüm bilgilerinin yerini alır.
- *PMLOGO* belirtilmezse, yürürlükteki ileti grubu ya da mantıksal ileti varsa çağrı başarısız olmaz; ancak, çağrı *CCWARN* tamamlama koduyla başarılı olabilir. Çizelge 717 sayfa 1160 , ortaya çıkabilecek çeşitli durumları gösterir. Bu durumlarda, tamamlanma kodu *CCOK* değilse, neden kodu aşağıdakilerden biridir (uygun olduğu şekilde):
 - RC2241
 - RC2242
 - RC2185
 - RC2245

Not: Kuyruk yöneticisi, *MQPUT1* çağrıyla ilgili grup ve bölüm bilgilerini denetlemez.

Çizelge 717. <i>MQPUT</i> ya da <i>MQCLOSE</i> çağrısı grup ve bölüm bilgileriyle tutarlı olmadığına sonuç		
Geçerli arama:	Önceki arama <i>PMLOGO</i> ile <i>MQPUT</i> idi	Önceki arama <i>PMLOGO</i> olmadan <i>MQPUT</i> idi
<i>PMLOGO</i> ile <i>MQPUT</i>	<i>CCFAIL</i>	<i>CCFAIL</i>
<i>PMLOGO</i> olmadan <i>MQPUT</i>	<i>CCWARN</i>	<i>CCOK</i>
Sonlandırılmamış grup ya da mantıksal ileti ile <i>MQCLOSE</i>	<i>CCWARN</i>	<i>CCOK</i>

Yalnızca iletileri ve bölümleri mantıksal sıraya koymak isteyen uygulamalar, *PMLOGO* seçeneğini belirlemek için önerilir; bu en basit seçenektir. Kuyruk yöneticisi bu bilgileri yönettiği için, bu seçenek grup ve bölüm bilgilerini yönetme gereksinimini ortadan kaldırır. Ancak, özelleştirilmiş uygulamalar *PMLOGO* seçeneği tarafından sağlanandan daha fazla denetime ihtiyaç duyabilirler ve bu, bu seçenek belirlenmeden gerçekleştirilebilir. Bu yapıldıysa, her *MQPUT* ya da *MQPUT1* çağrılarında önce, uygulama *MQMD* ' deki *MDGID*, *MDSEQ*, *MDOFF* ve *MDMFL* alanlarının doğru ayarlandığından emin olmalıdır.

Örneğin, aldığı fiziksel iletileri iletmek isteyen bir uygulama, bu iletilerin gruplar halinde mi, yoksa mantıksal ileti bölümlerinde mi olduğu dikkate alınmaksızın, *PMLOGO* belirtmemelidir. Bunun iki nedeni vardır:

- İletiler alınıp sıraya konursa, *PMLOGO* değerinin belirlenmesi iletilere yeni bir grup tanıtıcısının atanmasına neden olur ve bu, iletilerin yaratıcısının ileti grubundan kaynaklanan yanıt ya da rapor iletileri ilintilendirmesini zorlaştırabilir ya da olanaksız kılabilir.
- Gönderme ve alma kuyruk yöneticileri arasında birden çok yolu olan karmaşık bir ağda, fiziksel iletiler sıradışı gelebilir. *MQGET* çağrısında *PMLOGO* ve karşılık gelen *GMLOGO* belirtilmediğinde,

iletme uygulaması her fiziksel iletiyi gelir gelmez alabilir ve iletebilir, ancak mantıksal sırayla bir sonraki iletiyi beklemesine gerek yoktur.

Mantıksal iletilerin gruplarındaki ya da bölümlerindeki iletiler için rapor iletileri oluşturan uygulamalar, rapor iletisini koyarken PMLOGO belirtmemelidir.

PMLOGO, diğer PM* seçeneklerinden herhangi biriyle belirtilebilir.

Bağlam seçenekleri: Aşağıdaki seçenekler ileti bağlamının işlenmesini denetler:

PMNOC

İletiyile ilişkilendirilecek bir bağlam yok.

Hem kimlik hem de kaynak bağlamı, bağlam olmadığını gösterecek şekilde ayarlanır. Bu, MQMD 'deki bağlam alanlarının şu değere ayarlandığı anlamına gelir:

- Karakter alanları için boşluklar
- Bayt alanları için boş değerler
- Sayısal alanlar için sıfırlar

PMDEFC

Varsayılan bağlamı kullan.

İleti, hem kimlik hem de kaynak için, kendisiyle ilişkilendirilmiş varsayılan bağlam bilgisine sahip olmaktadır. Kuyruk yöneticisi, ileti tanımlayıcısındaki bağlam alanlarını aşağıdaki gibi ayarlar:

Çizelge 718. MQMD alanları için varsayılan bağlam bilgisi değerleri

MQMD 'deki Alan	Kullanılan değer
<i>MDUID</i>	Olanaklıysa ortamdaki belirlenir; tersi durumda boşluk değerine ayarlanır.
<i>MDACC</i>	Olanaklıysa ortamdaki belirlenir; tersi durumda ACNONE olarak ayarlanır.
<i>MDAID</i>	Boşluk olarak ayarlayın.
<i>MDPAT</i>	Çevreden belirlenmiştir.
<i>MDPAN</i>	Olanaklıysa ortamdaki belirlenir; tersi durumda boşluk değerine ayarlanır.
<i>MDPD</i>	İleti konduğunda tarihe ayarlayın.
<i>MDPT</i>	İleti konduğunda saat olarak ayarlayın.
<i>MDAOD</i>	Boşluk olarak ayarlayın.

İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

Bağlam seçeneği belirtilmezse, bu varsayılan işlemidir.

PMPASI

Bir giriş kuyruğu tanıtıcısından kimlik bağlamını geçirin.

İleti, kendisiyle ilişkilendirilmiş bağlam bilgisine sahip olacak. Kimlik bağlamı, *PMCT* alanında belirtilen kuyruk tanıtıcısından alınır. Kaynak bağlam bilgileri, kuyruk yöneticisi tarafından PMDEFC ile aynı şekilde oluşturulur (değerler için önceki çizelgeye bakın). İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

MQPOT çağrısı için, kuyruk OOPASI seçeneğiyle (ya da bunu gösteren bir seçenikle) açılmış olmalıdır. MQPUT1 çağrısı için, OOPASI seçeneğiyle MQOPEN çağrısıyla aynı yetki denetimi gerçekleştirilir.

PMPASA

Giriş kuyruğu tanıtıcısından tüm bağlamı geçirin.

İleti, kendisiyle ilişkilendirilmiş bağlam bilgisine sahip olacak. Kimlik ve kaynak bağlamı, *PMCT* alanında belirtilen kuyruk tanıtıcısından alınır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

MQPUT çağrısıyla ilgili olarak, kuyruk OOPASA seçeneğiyle (ya da bunu gösteren bir seçenekle) açılmış olmalıdır. MQPUT1 çağrısı için, OOPASA seçeneğiyle MQOPEN çağrısıyla aynı yetki denetimi gerçekleştirilir.

PMSETI

Uygulamadan kimlik bağlamını ayarlayın.

İleti, kendisiyle ilişkilendirilmiş bağlam bilgisine sahip olacak. Uygulama, MQMD yapısında kimlik bağlamını belirtir. Kaynak bağlam bilgileri, kuyruk yöneticisi tarafından PMDEFC ile aynı şekilde oluşturulur (değerler için önceki çizelgeye bakın). İleti bağlamıyla ilgili daha fazla bilgi için İleti bağlamı ve Bağlam bilgilerini denetleme başlıklı konuya bakın.

MQPUT çağrısı için, kuyruk OOSSETI seçeneğiyle (ya da bunu gösteren bir seçenekle) açılmış olmalıdır. MQPUT1 çağrısı için, OOSSETI seçeneğiyle MQOPEN çağrısıyla aynı yetki denetimi gerçekleştirilir.

PMSETA

Uygulamadan tüm bağlamı ayarlayın.

İleti, kendisiyle ilişkilendirilmiş bağlam bilgisine sahip olacak. Uygulama, MQMD yapısında kimlik ve kaynak bağlamını belirtir. İleti bağlamıyla ilgili daha fazla bilgi için İleti bağlamı ve Bağlam bilgilerini denetleme başlıklı konuya bakın.

MQPUT çağrısı için, kuyruk OOSSETA seçeneğiyle açılmış olmalıdır. MQPUT1 çağrısı için, OOSSETA seçeneğiyle MQOPEN çağrısıyla aynı yetki denetimi gerçekleştirilir.

PM* bağlam seçeneklerinden yalnızca biri belirtilebilir. Bu seçeneklerden hiçbiri belirtilmezse, PMDEFC varsayılır.

Yanıt tiplerini koyun. Aşağıdaki seçenekler, bir MQPUT ya da MQPUT1 çağrısına döndürülen yanıtı denetler. Bu seçeneklerden yalnızca birini belirleyebilirsiniz. PMARES ve PMSRES belirtilmezse, PMRASQ ya da PMRAST varsayılır.

MARLAR

PMARES seçeneği, uygulama kuyruk yöneticisinin çağrısı tamamlamasını beklemeden bir MQPUT ya da MQPUT1 işleminin tamamlanmasını ister. Bu seçeneğin kullanılması, özellikle istemci bağ tanımlarını kullanan uygulamalar için ileti sistemi başarımını artırabilir. Bir uygulama, MQSTAT komutunu kullanarak, önceki zamanuyumsuz çağrılar sırasında bir hata oluşup oluşmadığını düzenli aralıklarla denetleyebilir.

Bu seçenekle, MQMD ' de yalnızca aşağıdaki alanların doldurulacağı garanti edilir;

- MDAID
- MDPAT
- MDPAN
- MDAOD

Ayrıca, seçenekler olarak PMNMID ya da PMNCID ' lerden biri ya da her ikisi belirtilirse, döndürülen MDMID ve MDCID de tamamlanır. (PMNMID, boş bir MDMID alanı belirtilerek örtük olarak belirtilebilir).

Yalnızca önceden belirlenen alanlar doldurulmuştur. Olağan durumda MQMD ya da MQPMO yapısında döndürülecek diğer bilgiler tanımsız.

MQPUT ya da MQPUT1 için zamanuyumsuz koyma yanıtı istenirken, CMPCOD ve REASON cCOK ve RCNONE, iletinin başarıyla kuyruğa konduğunu göstermez. Zamanuyumsuz koyma yanıtı kullanan ve iletilerin kuyruğa konduğunu doğrulamasını gerektiren bir MQI uygulaması geliştirirken, koyma işlemlerinden hem CMPCOD hem de REASON kodlarını denetlemeli ve zamanuyumsuz hata bilgilerini sorgulamak için MQSTAT kullanmalısınız.

Her bir MQPUT/MQPUT1 çağrısının başarısı ya da başarısızlığı hemen döndürülme de, zamanuyumsuz bir çağrı altında oluşan ilk hata, MQSTAT çağrısıyla daha sonraki bir bağlantıda saptanabilir.

Syncpoint altındaki kalıcı bir ileti zamanuyumsuz koyma yanıtı kullanılarak teslim edilemezse ve işlemi kesinleştirme girişiminde bulunursanız, kesinleştirme başarısız olur ve işlem CCFAIL tamamlanma kodu ve RC2003 nedeni ile geri çekilir. Uygulama, önceki bir MQPUT ya da MQPUT1 hatasının nedenini saptamak için MQSTAT çağrısında bulunabilir.

PMSRES

MQPMO yapısında bir koyma seçeneği için bu değer belirtilmesi, MQPUT ya da MQPUT1 işleminin her zaman zamanuyumlu olarak yayınlanmasını sağlar. İşlem başarılı olursa, MQMD ve MQPMO 'daki tüm alanlar tamamlanır. Kuyrukta ya da konu nesnesinde tanımlanan varsayılan koyma yanıtı değerinden bağımsız olarak zamanuyumlu bir yanıt sağlamak için sağlanır.

PMRASQ

Bir MQPUT çağrısı için bu değer belirtilirse, kullanılan koyma yanıtı tipi, uygulama tarafından açıldığında kuyrukta belirtilen DEFPRESP değerinden alınır. Bir istemci uygulaması IBM WebSphere MQ 7.0 düzeyinden önceki bir düzeyde bir kuyruk yöneticisine bağlıysa, PMSRES belirtilmiş gibi davranır.

Bir MQPUT1 çağrısı için bu seçenek belirtilirse, kuyruk tanımlamasındaki DEFPRESP değeri kullanılmaz. MQPUT1 çağrısı PMSYP kullanıyorsa, PMARES için olduğu gibi davranır ve PMNSYP kullanıyorsa, PMSRES için olduğu gibi davranır.

PMRAST

Bu, konu nesneleriyle kullanılmak üzere PMRASQ ile eşanlamlıdır.

Diğer seçenekler: Aşağıdaki seçenekler yetki denetimini denetler ve kuyruk yöneticisi susturulduğunda ne olur:

PMALTU

Belirtilen kullanıcı kimliğiyle doğrulayın.

Bu, MQPUT1 çağrısının **OBJDSC** değiştirgesindeki *ODAU* alanının, iletileri kuyruğa koymak için yetkinin geçerliliğini denetlemek için kullanılacak bir kullanıcı kimliği içerdiğini gösterir. Çağrı, yalnızca *ODAU* 'in belirtilen seçeneklerle kuyruğu açma yetkisi varsa, uygulamanın altında çalıştığı kullanıcı kimliğinin bunu yapma yetkisi olup olmadığına bakılmaksızın başarılı olabilir. (Bu, belirtilen bağlam seçenekleri için geçerli değildir; ancak, her zaman uygulamanın çalıştığı kullanıcı kimliğiyle karşılaştırılarak denetlenir.)

Bu seçenek yalnızca MQPUT1 çağrısıyla geçerlidir.

PMFIQ

Kuyruk yöneticisi susturulursa başarısız olur.

Bu seçenek, kuyruk yöneticisi susturma durumundaysa, MQPUT ya da MQPUT1 çağrısına başarısız olmasına neden olur.

Çağrı, RC2161 neden koduyla CCFAIL tamamlanma kodunu döndürür.

Varsayılan seçenek: Önceden açıklanan seçeneklerin hiçbiri gerekmiyorsa, aşağıdaki seçenek kullanılabilir:

PMNONE

Seçenek belirtilmedi.

Bu değer, başka seçenek belirtilmediğini belirtmek için kullanılabilir; tüm seçenekler varsayılan değerlerini varsayar. PMNONE, program belgelerine yardımcı olmak üzere tanımlanır; bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmaz, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu bir giriş alanıdır. *PMOPT* alanının ilk değeri PMNONE değeridir.

PMPRF (10 basamaklı işaretli tamsayı)

Hangi MQPMR alanlarının var olduğunu gösteren işaretler.

Bu alan, uygulama tarafından sağlanan koyma iletileri kayıtlarında hangi MQPMR alanlarının bulunduğunu belirtmek için ayarlanması gereken işaretleri içerir. *PMPRF* yalnızca ileti bir dağıtım listesine konurken kullanılır. *PMREC* sıfır ya da *PMPRO* ve *PMPRP* her ikisi de sıfır alan yoksa.

Varolan alanlar için, kuyruk yöneticisi her hedef için karşılık gelen koyma iletisi kaydındaki alanlardaki değerleri kullanır. Var olmayan alanlar için, kuyruk yöneticisi MQMD yapısındaki değerleri kullanır.

Koyma iletisi kayıtlarında hangi alanların bulunduğunu belirtmek için aşağıdaki işaretlerden biri ya da birkaçı belirtilebilir:

PFMID

İleti tanıtıcısı alanı var.

PFCID

İlinti tanıtıcısı alanı var.

PFGTnt

Grup tanıtıcısı alanı var.

PFFB.

Geribildirim alanı var.

PFACC

Muhasebe simgesi alanı var.

Bu işaret belirtilirse, *PMOPT* alanında *PMSETI* ya da *PMSETA* belirtilmelidir; bu koşul karşılanmazsa, çağrı RC2158 neden koduyla başarısız olur.

MQPMR alanı yoksa, aşağıdaki belirtilebilir:

PFNONE

Bir koyma iletisi kayıt alanı yok.

Bu değer belirtilirse, *PMREC* sıfır ya da her ikisi de *PMPRO* ve *PMPRP* sıfır olmalıdır.

PFNONE, program belgelerine yardımcı olmak için tanımlanır. Bu değişimin başka bir değişimle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğundan bu tür bir kullanım algılanamaz.

PMPRF geçerli olmayan işaretler içeriyorsa ya da konan ileti kayıtları sağlandıysa, ancak *PMPRF* PFNONE değerine sahipse, çağrı RC2158 neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri PFNONE. *PMVER* *PMVER2* değerinden küçükse bu alan yoksayılr.

PMPRO (10 basamaklı işaretli tamsayı)

MQPMO ' nun başlangıcından ilk koyma iletisi kaydının görelî konumu.

Bu, MQPMO yapısının başlangıcından itibaren ilk MQPMR koyma iletisi kaydının bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir. *PMPRO* yalnızca ileti bir dağıtım listesine konurken kullanılır. *PMREC* sıfırsa alan yoksayılr.

İleti bir dağıtım listesine konduğunda, her hedef için iletinin belirli özelliklerini tek tek belirtmek üzere bir ya da daha fazla MQPMR koyma iletisi kaydı dizisi sağlanabilir; bu özellikler şunlardır:

- İleti Tanıtıcısı
- ilinti tanıtıcısı
- grup tanıtıcısı
- geribildirim değeri
- muhasebe simgesi

Bu özelliklerin tümünün belirtilmesi gerekmez, ancak hangi altküme seçilirse seçilsin, alanların doğru sırayla belirtilmesi gerekir. Daha fazla ayrıntı için MQPMR yapısının tanımına bakın.

Genellikle, dağıtım listesi açıldığında MQOD tarafından belirtilen nesne kayıtları kadar çok koyma iletisi kaydı olmalıdır; her bir koyma iletisi kaydı, ilgili nesne kaydı tarafından tanımlanan kuyruk için ileti özelliklerini sağlar. Dağıtım listesindeki açılmayan kuyruklar, bu durumda ileti özellikleri yoksayılsa da, dizideki uygun konumlara kendilerine ayrılmış ileti kayıtlarını yerleştirmeye devam etmelidir.

Koyma iletisi kaydı sayısı, nesne kaydı sayısından farklı olabilir. Nesne kayıtlarından daha az koyma iletisi kaydı varsa, koyma iletisi kayıtları olmayan hedeflere ilişkin ileti özellikleri, ileti tanımlayıcısı

MQMD 'deki ilgili alanlardan alınır. Nesne kayıtlarından daha fazla koyma iletisi kaydı varsa, fazlalık kullanılmaz (bunlara erişmek yine de mümkün olmalıdır). Koyma iletisi kayıtları isteğe bağlıdır, ancak bunlar sağlanırsa *PMREC* olmalıdır.

Koyma iletisi kayıtları, *PMPRO* içinde bir görelî konum belirtilerek ya da *PMPRP* içinde bir adres belirtilerek MQOD 'daki nesne kayıtlarına benzer şekilde sağlanabilir; Bunun nasıl yapılacağını öğrenmek için "IBM i üzerinde MQOD (Nesne tanımlayıcı)" sayfa 1141 içinde açıklanan *ODORO* alanına bakın.

PMPRO ve *PMPRP* seçeneklerinden en fazla biri kullanılabilir; her ikisi de sıfır değilse, çağrı RC2159 neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *PMVER* *PMVER2* değerinden küçükse bu alan yoksayılr.

PMPRP (gösterge)

İlk koyma iletisi kaydının adresi.

Bu, ilk MQPMR koyma iletisi kaydının adresidir. *PMPRP* yalnızca ileti bir dağıtım listesine konurken kullanılır. *PMREC* sıfırsa alan yoksayılr.

Koyma iletisi kayıtlarını belirtmek için *PMPRP* ya da *PMPRO* kullanılabilir, ancak her ikisini birden belirleyemez; ayrıntılar için *PMRRO* alanının açıklamasına bakın. *PMPRP* kullanılmazsa, boş değeri göstergeye ya da boş değeri byte 'lara ayarlanmalıdır.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş işaretçidir. *PMVER* *PMVER2* değerinden küçükse bu alan yoksayılr.

PMREC (10 basamaklı işaretli tamsayı)

Var olan koyma iletisi kayıtlarının ya da yanıt kayıtlarının sayısı.

Bu, uygulama tarafından sağlanan MQPMR koyma iletisi kayıtlarının ya da MQRR yanıt kayıtlarının sayısıdır. Bu sayı, ileti bir dağıtım listesine konuluyorsa sıfırdan büyük olabilir. Konan ileti kayıtları ve yanıt kayıtları isteğe bağlıdır; uygulamanın herhangi bir kayıt sağlamasına gerek yoktur ya da yalnızca bir tip kayıt sağlamayı seçebilir. Ancak, uygulama her iki tipten de kayıt sağlarsa, her tipten *PMREC* kayıt sağlamalıdır.

PMREC değeri, dağıtım listesindeki hedef sayısı ile aynı olmalıdır. Çok fazla kayıt sağlanırsa, fazla kayıt kullanılmaz; çok az kayıt sağlanırsa, ileti kaydı olmayan hedeflere ilişkin ileti özellikleri için varsayılan değerler kullanılır (bu konuda daha sonra bkz. *PMPRO*).

PMREC sıfırdan küçükse ya da sıfırdan büyükse, ancak ileti bir dağıtım listesine konmuyorsa, çağrı RC2154 neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *PMVER* *PMVER2* değerinden küçükse bu alan yoksayılr.

PMRMN (48 baytlık karakter dizgisi)

Hedef kuyruk yöneticisinin çözülmüş adı.

Bu, ad çözme işlemi yerel kuyruk yöneticisi tarafından gerçekleştirildikten sonra hedef kuyruk yöneticisinin adıdır. Döndürülen ad, *PMRQN* ile tanımlanan kuyruğa sahip olan kuyruk yöneticisinin adıdır ve yerel kuyruk yöneticisinin adı olabilir.

PMRQN , yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun sahip olduğu paylaşılan bir kuyruksa, *PMRMN* , kuyruk paylaşım grubunun adıdır. Kuyruk başka bir kuyruk paylaşım grubuna aitse, *PMRQN* kuyruk paylaşım grubunun adı ya da kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisinin adı olabilir (döndürülen değerin türü, yerel kuyruk yöneticisinde var olan kuyruk tanımlamalarına göre belirlenir).

Boş olmayan bir değer yalnızca nesne tek bir kuyruksa döndürülür; nesne bir dağıtım listesi ya da konu ise, döndürülen değer tanımsız olur.

Bu bir çıkış alanıdır. Bu alanın uzunluğu LNQMN tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

PMRQN (48 baytlık karakter dizgisi)

Hedef kuyruğun çözülmüş adı.

Ad çözme işlemi yerel kuyruk yöneticisi tarafından gerçekleştirildikten sonra hedef kuyruğun adıdır. Döndürülen ad, *PMRMN* ile tanımlanan kuyruk yöneticisinde bulunan bir kuyruğun adıdır.

Boş olmayan bir değer yalnızca nesne tek bir kuyruksa döndürülür; nesne bir dağıtım listesi ya da konu ise, döndürülen değer tanımsız olur.

Bu bir çıkış alanıdır. Bu alanın uzunluğu LNQN tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

PMRRO (10 basamaklı işaretli tamsayı)

MQPMO ' nun başlangıcındaki ilk yanıt kaydının görel konumu.

Bu, MQPMO yapısının başlangıcından itibaren ilk MQRR yanıt kaydının bayt cinsinden görel konumudur. Görel konum pozitif ya da negatif olabilir. *PMRRO* yalnızca ileti bir dağıtım listesine konurken kullanılır. *PMREC* sıfırsa alan yoksayıdır.

İleti bir dağıtım listesine konduğunda, iletinin başarıyla gönderilmediği kuyrukları (MQRR 'deki *RRCC* alanı) ve her başarısızlığın nedenini (MQRR' deki *RRREA* alanı) saptamak için bir ya da daha çok MQRR yanıt kaydı dizisi sağlanabilir. Kuyruk açılmadığı için ya da koyma işlemi başarısız olduğu için ileti gönderilmemiş olabilir. Kuyruk yöneticisi, yanıt kayıtlarını yalnızca çağrı sonucu karışık olduğunda (diğer bir deyişle, bazı iletiler başarısız olduğunda ya da tümü farklı nedenlerle başarısız olduğunda) ayarlar; çağrıdaki RC2136 neden kodu bu durumu gösterir. Aynı neden kodu tüm kuyruklar için de geçerliyse, MQPUT ya da MQPUT1 çağrısının **REASON** değiştirilmesinde bu neden döndürülür ve yanıt kayıtları ayarlanmaz.

Genellikle, dağıtım listesi açıldığında MQOD tarafından belirtilen nesne kayıtları kadar yanıt kaydı olmalıdır; gerektiğinde, her yanıt kaydı, ilgili nesne kaydı tarafından belirlenen kuyruğa konmaya ilişkin tamamlanma koduna ve neden koduna ayarlanır. Dağıtım listesindeki açılmayan kuyruklar, koyma işlemi yerine, açık işlemden kaynaklanan tamamlanma koduna ve neden koduna ayarlanmış olsalar da, dizideki uygun konumlarda bunlar için ayrılmış yanıt kayıtları olmalıdır.

Yanıt kaydı sayısı, nesne kaydı sayısından farklı olabilir. Nesne kayıtlarından daha az yanıt kaydı varsa, uygulamanın koyma işleminin başarısız olduğu tüm hedefleri ya da başarısızlıkların nedenlerini tanımlaması mümkün olmayabilir. Nesne kayıtlarından daha fazla yanıt kaydı varsa, fazlalık kullanılmaz (bunlara erişmek yine de mümkün olmalıdır). Yanıt kayıtları isteğe bağlıdır, ancak bunlar sağlanırsa *PMREC* olmalıdır.

Yanıt kayıtları, *PMRRO* içinde bir görel konum belirtilerek ya da *PMRRP* içinde bir adres belirtilerek MQOD ' daki nesne kayıtlarına benzer bir şekilde sağlanabilir; Bunun nasıl yapılacağını öğrenmek için "IBM i üzerinde MQOD (Nesne tanımlayıcı)" sayfa 1141 içinde açıklanan *ODORO* alanına bakın. Ancak, en fazla *PMRRO* ve *PMRRP* kullanılabilir; her ikisi de sıfır değilse, çağrı RC2156 neden koduyla başarısız olur.

MQPUT1 çağrısı için bu alan sıfır olmalıdır. Bunun nedeni, yanıt bilgilerinin (istenirse) nesne tanımlayıcısı MQOD tarafından belirtilen yanıt kayıtlarında döndürülmesi olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır. *PMVER* *PMVER2* değerinden küçükse bu alan yoksayıdır.

PMRRP (işaretçi)

İlk yanıt kaydının adresi.

Bu, ilk MQRR yanıt kaydının adresidir. *PMRRP* yalnızca ileti bir dağıtım listesine konurken kullanılır. *PMREC* sıfırsa alan yoksayıdır.

Yanıt kayıtlarını belirtmek için *PMRRP* ya da *PMRRO* kullanılabilir, ancak her ikisini birden belirleyemez; ayrıntılar için *PMRRO* alanının açıklamasına bakın. *PMRRP* kullanılmazsa, boş değerli göstergeye ya da boş değerli byte 'lara ayarlanmalıdır.

MQPUT1 çağrısı için bu alan boş değerli gösterge ya da boş değerli byte olmalıdır. Bunun nedeni, yanıt bilgilerinin (istenirse) nesne tanımlayıcısı MQOD tarafından belirtilen yanıt kayıtlarında döndürülmesi olabilir.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş işaretçidir. *PMVER* *PMVER2*değerinden küçükse bu alan yoksayılır.

PMSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

PMSIDV

Koyma iletisi seçenekleri yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri PMSIDV 'dir.

PMSL (MQLONG)

Bu yayın tarafından hedeflenen abonelik düzeyi.

Bu yayını yalnızca bu değerden küçük ya da bu değere eşit *PMSL* en yüksek aboneliklere sahip olan abonelikler alır. Bu değer sıfır ile 9 aralığında olmalıdır; sıfır en düşük düzeydir.

Bu alanın ilk değeri 9 'dur.

PMTO (10 basamaklı işaretli tamsayı)

Ayrılmış.

Bu ayrılmış bir alandır; değeri önemli değildir. Bu alanın ilk değeri şudur: -1.

PMUDC (10 basamaklı işaretli tamsayı)

Uzak kuyruklara başarıyla gönderilen iletilerin sayısı.

Bu, yürürlükteki MQPUT ya da MQPUT1 çağrısının dağıtım listesindeki uzak kuyruklara çözülecek kuyruklara başarıyla gönderdiği iletilerin sayısıdır. Kuyruk yöneticisinin dağıtım listesi formunda geçici olarak tuttuğu iletiler, bu dağıtım listelerinin içerdiği tek tek hedeflerin sayısı olarak sayılır. Bu alan, dağıtım listesinde olmayan tek bir kuyruğa ileti yerleştirilirken de ayarlanır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır. *PMVER* , *PMVER2*değerinden küçükse bu alan ayarlanmaz.

PMVER (10 basamaklı imzalı tamsayı)

Yapı sürümü numarası.

Değer aşağıdakilerden biri olmalıdır:

PMVER1

Version-1 koyma iletisi seçenekleri yapısı.

PMVER2

Version-2 koyma iletisi seçenekleri yapısı.

Yalnızca yapının daha yeni sürümünde var olan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

PMVERC

Koyma iletisi seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri PMVER1' dir.

Başlangıçtaki değerler

Çizelge 719. MQPMO içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
<i>PMSID</i>	PMSIDV	' PMO→ '
<i>PMVER</i>	PMVER1	1

Çizelge 719. MQPMO içindeki alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
PMOPT	PMNONE	0
PMTO	Yok	-1
PMCT	Yok	0
PMKDC	Yok	0
PMUDC	Yok	0
PMIDC	Yok	0
PMRQN	Yok	Boşluklar
PMRMN	Yok	Boşluklar
PMREC	Yok	0
PMPRF	PFNONE	0
PMPRO	Yok	0
PMRRO	Yok	0
PMPRP	Yok	Boş değerli gösterge ya da boş değerli byte
PMRRP	Yok	Boş değerli gösterge ya da boş değerli byte

Not:

1. - simgesi, tek bir boş karakteri temsil eder.

RPG bildirimi

```

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	PMRRP	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

IBM i üzerinde MQPMR (Put-message record)

MQPMR yapısı, bir ileti dağıtım listesine konurken tek bir hedefe ilişkin çeşitli ileti özelliklerini belirtmek için kullanılır.

Genel Bakış

Amaç: MQPMR, MQPUT ve MQPUT1 çağrılarına ilişkin bir giriş/çıkış yapısıdır.

Karakter kümesi ve kodlama: MQPMR 'daki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Kullanım: MQPUT ya da MQPUT1 çağrısında bu yapılardan oluşan bir dizi sağlanarak, dağıtım listesindeki her hedef kuyruk için farklı değerler belirtilebilir. Alanlardan bazıları yalnızca giriş, diğerleri giriş/çıkış.

Not: Bu yapı, sabit bir düzene sahip olmadığı için alışılmadık bir yapıdır. Bu yapıdaki alanlar isteğe bağlıdır ve her alanın varlığı ya da yokluğu, MQPMO 'daki *PMPRF* alanındaki işaretlerle gösterilir. **olan alanlar aşağıdaki sırayla oluşmalıdır :**

- *PRMID*
- *PRCID*
- *PRGID*
- *PRFB*
- *PRACC*

Eksik olan alanlar, kayıta yer kaplamıyor.

MQPMR 'nin değişmez bir yerleşim düzeni olmadığından, COPY dosyasında bu yerleşim düzeninin tanımlaması sağlanmaz. Uygulama programcısı, uygulamanın gerektirdiği alanları içeren bir bildirim yaratmalı ve var olan alanları belirtmek için *PMPRF* içindeki işaretleri ayarlamalıdır.

- [“Alanlar” sayfa 1169](#)
- [“Başlangıçtaki değerler” sayfa 1171](#)
- [“RPG bildirim” sayfa 1171](#)

Alanlar

MQPMR yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

PRACC (32 baytlık bit dizisi)

Hesap simgesi.

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen bir adla kuyruğa gönderilen ileti için kullanılacak hesap simgesidir. Tek bir kuyruğa koyma değeri için MQMD 'deki *MDACC* alanıyla aynı şekilde işlenir. Bu alanın içeriğine ilişkin bilgi için [“IBM i üzerinde MQMD \(İleti tanımlayıcı\)” sayfa 1093](#) içindeki *MDACC* açıklamasına bakın.

Bu alan yoksa, MQMD 'deki değer kullanılır.

Bu bir giriş alanıdır.

PRCID (24 baytlık bit dizgisi)

İlinti tanıtıcısı.

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen adla kuyruğa gönderilen ileti için kullanılacak ilinti tanıtıcısıdır. Tek bir kuyruğa koyma değeri için MQMD ' deki *MDCID* alanıyla aynı şekilde işlenir.

Bu alan MQPMR kaydında yoksa ya da hedeflerden daha az MQPMR kaydı varsa, *PRCID* alanı içeren bir MQPMR kaydı olmayan hedefler için MQMD ' deki değer kullanılır.

PMNCID belirtilirse, dağıtım listesindeki tüm hedefler için, MQPMR kayıtları olup olmadığına bakılmaksızın *tek* yeni bir ilinti tanıtıcısı oluşturulur ve kullanılır. Bu, PMNMID ' nin işleniş biçiminden farklıdır (bkz. *PRMID* alanı).

Bu bir giriş/çıkış alanıdır.

PRFB (10 basamaklı işaretli tamsayı)

Geribildirim ya da neden kodu.

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen adla kuyruğa gönderilen ileti için kullanılacak geribildirim kodudur. Tek bir kuyruğa koyma değeri için MQMD ' deki *MDFB* alanıyla aynı şekilde işlenir.

Bu alan yoksa, MQMD ' deki değer kullanılır.

Bu bir giriş alanıdır.

PRGID (24 baytlık bit dizgisi)

Grup tanıtıcısı.

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen adla kuyruğa gönderilen ileti için kullanılacak grup tanıtıcısıdır. Tek bir kuyruğa koyma değeri için MQMD ' deki *MDGID* alanıyla aynı şekilde işlenir.

Bu alan MQPMR kaydında yoksa ya da hedeflerden daha az MQPMR kaydı varsa, *PRGID* alanı içeren bir MQPMR kaydı olmayan hedefler için MQMD ' deki değer kullanılır. Değer, Çizelge 716 sayfa 1158 içinde belgelendiği şekilde, ancak aşağıdaki farklarla işlenir:

- Yeni bir grup tanıtıcısının kullanılacağı durumlarda, kuyruk yöneticisi her hedef için farklı bir grup tanıtıcısı oluşturur (yani, iki hedef aynı grup tanıtıcısına sahip değildir).
- Alandaki değer kullanılacağı durumlarda, çağrı RC2258neden koduyla başarısız olur.

Bu bir giriş/çıkış alanıdır.

PRMID (24 baytlık bit dizgisi)

İleti tanıtıcısı.

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili öge tarafından belirtilen adla kuyruğa gönderilen ileti için kullanılacak ileti tanıtıcısıdır. Tek bir kuyruğa koyma değeri için MQMD ' deki *MDMID* alanıyla aynı şekilde işlenir.

Bu alan MQPMR kaydında yoksa ya da hedeflerden daha az MQPMR kaydı varsa, *PRMID* alanı içeren bir MQPMR kaydı olmayan hedefler için MQMD ' deki değer kullanılır. Bu değer MINONE ise, bu hedeflerin *her biri* için yeni bir ileti tanıtıcısı oluşturulur (yani, bu hedeflerin ikisi aynı ileti tanıtıcısına sahip değildir).

PMNMID belirtilirse, dağıtım listesindeki tüm hedefler için, MQPMR kayıtları olup olmadığına bakılmaksızın yeni ileti tanıtıcıları oluşturulur. Bu, PMNCID ' in işleniş biçiminden farklıdır (bkz. *PRCID* alanı).

Bu bir giriş/çıkış alanıdır.

Başlangıçtaki değerler

Yapı bildirimini sağlanmadığı için bu yapı için tanımlanmış başlangıç değeri yok. Aşağıdaki örnek bildirim, tüm alanlar gerekliyse, yapının uygulama programcısı tarafından nasıl bildirilmesi gerektiğini göstermektedir.

RPG bildirim

```
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
```

IBM i

IBM i üzerinde MQRFH (Kurallar ve biçimlendirme üstbilgisi)

MQRFH yapısı, kuralların ve biçimlendirme üstbilgisinin düzenini tanımlar.

Genel Bakış

Amaç: Bu üstbilgi, dizilim verilerini ad-değer çiftleri biçiminde göndermek için kullanılabilir.

Biçim adı: FMRFH.

Karakter kümesi ve kodlama: MQRFH yapısındaki alanlar (*RFNVS*dahil), MQRFH ' den önce gelen üstbilgi yapısında *MDCSI* ve *MDENC* alanları tarafından verilen karakter kümesi ve kodlamada ya da MQRFH uygulama ileti verilerinin başındadır.

Karakter takımı, kuyruk adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır.

- [“Alanlar” sayfa 1171](#)
- [“Başlangıçtaki değerler” sayfa 1173](#)
- [“RPG bildirim” sayfa 1174](#)

Alanlar

MQRFH yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

RFCSI (10 basamaklı işaretli tamsayı)

RFNVS' i izleyen verilerin karakter kümesi tanıtıcısı.

Bu, *RFNVS* ; ' den sonra gelen verilerin karakter kümesi tanıtıcısını belirtir. MQRFH yapısının kendisindeki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değer kullanılabilir:

CSINHT

Bu yapının karakter kümesi tanıtıcısını edin.

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı CSINHT değerini döndürmez.

MQMD ' deki *MDPAT* alanının değeri ATBRKR ise, CSINHT kullanılamaz.

Bu alanın ilk değeri CSUNDF 'dir.

RFNVS' i izleyen verilerin sayısal kodlaması.

Bu, *RFNVS* ; ' yi izleyen verilerin sayısal kodlamasını belirtir. MQRFH yapısının kendisindeki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri ENNAT ' tır.

RFFLG (10 basamaklı işaretli tamsayı)

Bayraklar.

Aşağıdaki belirtilebilir:

RFNONE

Bayrak yok.

Bu alanın ilk değeri RFNONE.

RFFMT (8 baytlık karakter dizgisi)

RFNVS' i izleyen verilerin biçim adı.

Bu, *RFNVS*' i izleyen verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *MDFMT* alanıyla aynıdır.

Bu alanın ilk değeri FMNONE.

RFLEN (10 basamaklı işaretli tamsayı)

*RFNVS*da dahil olmak üzere MQRFH ' nin toplam uzunluğu.

Bu, yapının sonundaki *RFNVS* alanı da içinde olmak üzere MQRFH yapısının bayt cinsinden uzunluğudur. Uzunluk, *RFNVS* alanını izleyen kullanıcı verilerini içermez.

Bazı ortamlardaki kullanıcı verilerinin dönüştürülmesinde sorun çıkmasını önlemek için, *RFLEN* ' i dörtlü bir veri katı olarak kullanmayı düşünün.

Aşağıdaki değişmez, *RFNVS* alanı dışında, yapının *fixed* kısmının uzunluğunu gösterir:

RFLENV

MQRFH yapısının sabit kısmının uzunluğu.

Bu alanın ilk değeri RFLENV 'dir.

RFNVS (n baytlık karakter dizgisi)

Ad-değer çiftlerini içeren dizgi.

Bu, formda ad-değer çiftlerini içeren değişken uzunluklu bir karakter dizgisidir:

```
name1 value1 name2 value2 name3 value3 ...
```

Her ad ya da değer, bitişik addan ya da değerden bir ya da daha çok boşluk karakteriyle ayrılmalıdır; bu boşluklar önemli değildir. Ad ya da değer, adın ya da değer başına ve sonuna tırnak işareti karakteri konarak anlamlı boşluklar içerebilir; açılış tırnak işareti ile eşleşen tırnak işareti arasındaki tüm karakterler anlamlı olarak kabul edilir. Aşağıdaki örnekte ad FAMOUS_WORDS ve değer Hello World' dir:

```
FAMOUS_WORDS "Hello World"
```

Bir ad ya da değer, boş karakter dışında herhangi bir karakter içerebilir (*RFNVS* için sınırlayıcı görevi görür). Ancak, bir uygulamanın birlikte çalışabilirliğini desteklemek için adları aşağıdaki karakterlerle sınırlamayı tercih edebilir:

- İlk karakter: büyük ya da küçük harf alfabetik (A-Z ya da a-z) ya da alt çizgi.
- Sonraki karakterler: büyük ya da küçük harf alfabetik, ondalık basamak (0-9), alt çizgi, tire ya da nokta.

Bir ad ya da değer bir ya da daha çok tırnak işareti içeriyorsa, ad ya da değer tırnak işareti içine alınmalı ve dizgi içindeki her tırnak işareti iki katına alınmalıdır:

```
Famous_Words "The program displayed ""Hello World"""
```

Adlar ve değerler büyük ve küçük harfe duyarlıdır; diğer bir deyişle, küçük harfler büyük harflerle aynı olarak kabul edilmez. Örneğin, FAMOUS_WORDS ve Famous_Words iki farklıdır.

RFNVS bayt cinsinden uzunluk *RFLEN* eksi *RFLENV* değerine eşittir. Bazı ortamlardaki kullanıcı verilerinin veri dönüştürülmesiyle ilgili sorunları önlemek için, bu uzunluğun dörtten bir katı olması önerilir. *RFNVS*, bu uzunluğa kadar boşluklarla doldurulmalı ya da dizedeki son anlamlı karakterin ardından boş bir karakter konarak daha önce sonlandırılmalıdır. Belirtilen *RFNVS* uzunluğuna kadar olan boş karakter ve ardından gelen bayt dikkate alınmaz.

Not: Bu alanın uzunluğu değişmez olmadığı için, alan, desteklenen programlama dilleri için sağlanan yapının bildirimlerinden çıkarılır.

RFSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

RFSIDV

Kurallar ve biçimlendirme üstbilgi yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri RFSIDV'dir.

RFVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

RFVER1

Version-1 kuralları ve üstbilgi yapısını biçimlendirme.

Bu alanın ilk değeri RFVER1'dir.

Başlangıçtaki değerler

Çizelge 720. MQRFH 'deki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
<i>RFSID</i>	RFSIDV	'RFH↵'
<i>RFVER</i>	RFVER1	1
<i>RFLEN</i>	RFLENV	32
<i>RFENC</i>	GIRIş	Ortama bağlıdır
<i>RFCSI</i>	CSUNDF	0
<i>RFFMT</i>	FMNONE	Boşluklar
<i>RFFLG</i>	RFNONE	0
Notlar:		
1. ↵ simgesi, tek bir boş karakteri temsil eder.		

RPG bildirimi

```
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)
```

IBM i

MQRFH2 (Kural ve biçimlendirme üstbilgisi 2) IBM i

MQRFH2 yapısı, version-2 kurallarının ve biçimlendirme üstbilgisinin biçimini tanımlar.

Genel Bakış

Amaç: Bu üstbilgi, XML benzeri bir sözdizimi kullanılarak kodlanmış verileri göndermek için kullanılabilir. Bir ileti, seri olarak iki ya da daha fazla MQRFH2 yapısı içerebilir; kullanıcı verileri isteğe bağlı olarak serideki son MQRFH2 yapısını takip edebilir.

Biçim adı: FMRFH2.

Karakter kümesi ve kodlama: Özel kurallar, MQRFH2 yapısı için kullanılan karakter kümesi ve kodlama için geçerlidir:

- *RF2NVD* dışındaki alanlar, MQRFH2' den önce gelen üstbilgi yapısındaki *MDCSI* ve *MDENC* alanları tarafından verilen karakter kümesinde ve kodlamasında ya da MQRFH2 uygulama iletisi verilerinin başlangıcında ise MQMD yapısındaki bu alanlar tarafından bulunur.

Karakter takımı, kuyruk adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır.

MQGET çağrısında GMCONV belirtildiğinde, kuyruk yöneticisi bu alanları istenen karakter kümesine ve kodlamaya dönüştürür.

- *RF2NVD* , *RF2NVC* alanı tarafından verilen karakter kümesinde bulunur. *RF2NVC* için yalnızca belirli Unicode karakter kümeleri geçerlidir (ayrıntılar için *RF2NVC* açıklamasına bakın).

Bazı karakter kümelerinin, kodlamaya bağlı bir gösterimi vardır. *RF2NVC* bu karakter kümelerinden biriye, *RF2NVD* , MQRFH2 içindeki diğer alanlarla aynı kodlamada olmalıdır.

MQGET çağrısında GMCONV belirtildiğinde, kuyruk yöneticisi *RF2NVD* ' i istenen kodlamaya dönüştürür, ancak karakter kümesini değiştirmez.

- [“Alanlar” sayfa 1174](#)
- [“Başlangıçtaki değerler” sayfa 1179](#)
- [“RPG bildirimi” sayfa 1179](#)

Alanlar

MQRFH2 yapısı aşağıdaki alanları içerir; alanlar alfabetik sırada açıklanır:

RF2CSI (10 basamaklı işaretli tamsayı)

Son *RF2NVD* alanını izleyen verilerin karakter kümesi tanıtıcısı.

Bu, son *RF2NVD* alanını izleyen verilerin karakter kümesi tanıtıcısını belirtir. MQRFH2 yapısındaki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değer kullanılabilir:

CSINHT

Bu yapının karakter kümesi tanıtıcısını edin.

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı CSINHT değerini döndürmez.

MQMD ' deki *MDPAT* alanının değeri ATBRKR ise, CSINHT kullanılamaz.

Bu alanın ilk değeri CSINHT ' dir.

RF2ENC (10 basamaklı işaretli tamsayı)

Son *RF2NVD* alanını izleyen verilerin sayısal kodlaması.

Bu, son *RF2NVD* alanını izleyen verilerin sayısal kodlamasını belirtir; MQRFH2 yapısındaki sayısal veriler için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri ENNAT ' tır.

RF2FLG (10 basamaklı işaretli tamsayı)

Bayraklar.

Aşağıdaki değer belirtilmelidir:

RFNONE

Bayrak yok.

Bu alanın ilk değeri RFNONE.

RF2FMT (8 baytlık karakter dizgisi)

Son *RF2NVD* alanını izleyen verilerin biçim adı.

Bu, son *RF2NVD* alanını izleyen verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *MDFMT* alanıyla aynıdır.

Bu alanın ilk değeri FMNONE.

RF2LEN (10 basamaklı işaretli tamsayı)

Tüm *RF2NVL* ve *RF2NVD* alanlarını içeren toplam MQRFH2 uzunluğu.

Bu, yapının sonundaki *RF2NVL* ve *RF2NVD* alanları da içinde olmak üzere MQRFH2 yapısının bayt cinsinden uzunluğudur. Yapının sonunda aşağıdaki sırayla birden çok *RF2NVL* ve *RF2NVD* alanı çifti olması geçerlidir:

```
length1, data1, length2, data2, ...
```

RF2LEN , yapının sonundaki son *RF2NVD* alanını izleyebilecek kullanıcı verilerini içermez.

Bazı ortamlardaki kullanıcı verilerinin dönüştürülmesinde sorun çıkmasını önlemek için, *RF2LEN* ' i dörtlü bir veri katı olarak kullanmayı düşünün.

Aşağıdaki değişmez, yapının *fixed* kısmının uzunluğunu, yani *RF2NVL* ve *RF2NVD* alanlarını dışlamayan uzunluğunu verir:

RFLEN2

MQRFH2 yapısının sabit kısmının uzunluğu.

Bu alanın ilk değeri RFLEN2' dir.

RF2NVC (10 basamaklı işaretli tamsayı)

RF2NVD karakter takımı tanıtıcısı.

Bu, *RF2NVD* alanındaki verilerin kodlanmış karakter takımı tanıtıcısını belirtir. Bu, MQRFH2 yapısındaki diğer dizgilerin karakter kümesinden farklıdır ve yapının sonundaki son *RF2NVD* alanını izleyen verilerin karakter kümesinden (varsa) farklı olabilir.

RF2NVC aşağıdaki CCSID değerlerinden birine sahip olmalıdır:

1200

UTF-16, desteklenen en son Unicode sürümü

13488

UTF-16, Unicode sürümü 2.0 altkümesi

17584

UTF-16, Unicode sürümü 3.0 altkümesi (Euro simgesini içerir)

1208

UTF-8, desteklenen en son Unicode sürümü

UTF-16 karakter kümeleri için *RF2NVD* kodlaması (byte sırası), MQRFH2 yapısındaki diğer alanların kodlamasıyla aynı olmalıdır. Vekil karakterler (X'D800'-X'DFFF') desteklenmez.

Not: *RF2NVC* daha önce listelenen değerlerden birine sahip değilse ve MQRFH2 yapısı MQGET çağrısında dönüştürme gerektiriyorsa, çağrı RC2111 neden koduyla tamamlanır ve ileti dönüştürülmeden döndürülür.

Bu alanın ilk değeri 1208 'dir.

RF2NVD (n baytlık karakter dizgisi)

Ad/değer verileri.

Bu, XML benzeri bir sözdizimi kullanılarak kodlanmış verileri içeren değişken uzunluklu bir karakter dizgisidir. Bu dizginin bayt cinsinden uzunluğu, *RF2NVD* alanından önce gelen *RF2NVL* alanı tarafından sağlanır; bu uzunluk dörtlü olmalıdır.

RF2NVL ve *RF2NVD* alanları isteğe bağlıdır, ancak varsa çift olarak oluşmaları ve bitişik olmaları gerekir. Alan çifti gerektiği kadar yinelenebilir, örneğin:

```
length1 data1 length2 data2 length3 data3
```

Bu alanlar isteğe bağlı olduğundan, desteklenen çeşitli programlama dilleri için sağlanan yapının bildirimlerinden çıkarılır.

RF2NVD , ileti GMCONV seçeneğiyle alındığında MQGET çağrısında belirtilen karakter kümesine dönüştürülmediğinden olağandışı bir durumdur; *RF2NVD* özgün karakter kümesinde kalır. Ancak *RF2NVD* , MQGET çağrısında belirtilen kodlamaya dönüştürülür.

Ad/değer verilerinin sözdizimi: Dizgi, sıfır ya da daha fazla özellik içeren tek bir "klasör" den oluşur. Klasör, klasörle aynı ada sahip XML başlangıç ve bitiş etiketleriyle sınırlanmıştır:

```
<folder> property1 property2 ... </folder>
```

RF2NVL ile tanımlanan uzunluğa kadar klasör bitiş etiketini izleyen karakterler boş olmalıdır. Klasör içinde, her özellik bir ad ve değerden ve isteğe bağlı olarak bir veri tipinden oluşur:

```
<name dt="datatype">value</name>
```

Bu örneklerde:

- Sınırlayıcı karakterler (<, =, ",/ve >) tam olarak gösterildiği gibi belirtilmelidir.

- name , özelliğin kullanıcı tarafından belirtilen adıdır; adlarla ilgili ek bilgi için aşağıdaki örneğe bakın.
- datatype , özelliğin isteğe bağlı bir kullanıcı tanımlı veri tipidir; geçerli veri tipleri için aşağıdaki örneğe bakın.
- value , özelliğin kullanıcı tarafından belirtilen değeridir; değerlerle ilgili daha fazla bilgi için aşağıdaki paragraflara bakın.
- Bir değerden önce gelen > karakteri ile değerden sonra gelen < karakteri ve en az bir boşluk dt=arasında anlamlı boşluk olmalıdır. Başka bir yerde, etiketler arasında ya da etiketlerden önce ya da sonra serbestçe kodlanabilir (örneğin, okunabilirliği artırmak için); bu boşluklar önemli değildir.

Özellikler birbiriyle ilişkiliyse, grupla aynı ada sahip XML başlangıç ve bitiş etiketleri içine alarak birlikte gruplanabilirler:

```
<folder> <group> property1 property2 ... </group> </folder>
```

Gruplar, sınır olmadan başka gruplar içinde içiçe yerleştirilebilir ve bir grup bir klasör içinde birden çok kez oluşabilir. Bir klasörün gruplarda ve gruplarda olmayan diğer özelliklerde bazı özellikleri içermesi de geçerlidir.

Özellik, grup ve klasör adları: Özellik, grup ve klasör adları, özellik, grup ya da klasör adında izin verilmeyen iki nokta üst üste karakteri dışında, geçerli XML etiket adları olmalıdır. Özellikle:

- Adlar bir harf ya da alt çizgiyle başlamalıdır. Geçerli harfler W3C XML belirtiminde tanımlanır ve esasen Unicode kategorileri Ll, Lu, Lo, Lt ve Nl 'den oluşur.
- Bir addaki diğer karakterler harfler, ondalık basamaklar, alt çizgiler, kısa çizgiler ya da noktalar olabilir. Bu, Unicode kategorileri Ll, Lu, Lo, Lt, Nl, Mc, Mn, Lm ve Nd kategorilerine karşılık gelir.
- Bir adın hiçbir bölümünde Unicode uyumluluk karakterlerine (X'F900' ve üstü) izin verilmez.
- Adlar, büyük ya da küçük harf karışımında XML dizisiyle başlamamalıdır.

Ek olarak:

- Adlar büyük ve küçük harfe duyarlıdır. Örneğin, ABC, abcve Abc üç farklı addır.
- Her klasörün ayrı bir ad alanı vardır. Sonuç olarak, bir klasördeki bir grup ya da özellik, başka bir klasörde aynı adı taşıyan bir grupla ya da özellikle çakışmaz.
- Gruplar ve özellikler, bir klasör içinde aynı ad alanını kaplar. Sonuç olarak, bir özellik, o özelliği içeren klasör içindeki bir grupla aynı ada sahip olamaz.

Genel olarak, RF2NVD alanını çözümlen programlar, bu özelliklerin ya da grupların doğru biçimde oluşturulmuş olması koşuluyla, programın tanımadığı özelliklere ya da gruplara sahip olan özellikleri ya da grupları yoksaymalıdır.

Özelliklerin veri tipleri: Her özelliğin isteğe bağlı bir veri tipi olabilir. Belirtirirse, veri tipi büyük, küçük ya da büyük harf karışık olarak aşağıdaki değerlerden biri olmalıdır:

<i>Çizelge 721. Veri tipleri ve bunların kullanımı</i>	
Veri tipi	Kullanılma Amacı
string	Herhangi bir karakter sırası. Çıkış dizileri kullanılarak belirli karakterler belirtilmelidir.
boolean	0 ya da 1 karakteri (1, TRUEdeğerini belirtir).
bin.hex	Sekizliyi gösteren onaltılı basamaklar.
i1	-128 ile +127 aralığında, yalnızca ondalık basamaklar ve isteğe bağlı işaret kullanılarak ifade edilen tamsayı.
i2	-32 768 ile +32 767 aralığında, yalnızca ondalık basamaklar ve isteğe bağlı işaret kullanılarak ifade edilen tamsayı.

Çizelge 721. Veri tipleri ve bunların kullanımı (devamı var)	
Veri tipi	Kullanılma Amacı
i4	-2 147 483 648+ 2 147 483 647 aralığında, yalnızca ondalık basamaklar ve isteğe bağlı işaret kullanılarak ifade edilen tamsayı.
i8	-9 223 372 036 854 775 808+ 9 223 372 036 854 775 807 aralığındaki tamsayı, yalnızca ondalık basamaklar ve isteğe bağlı işaret kullanılarak ifade edilir.
int	-9 223 372 036 854 775 808+ 9 223 372 036 854 775 807 aralığındaki tamsayı, yalnızca ondalık basamaklar ve isteğe bağlı işaret kullanılarak ifade edilir. Bu, i1, i2, i4ya da gönderenin belirli bir duyarlılığı belirtmesini istememesi durumunda i8 yerine kullanılabilir.
r4	1.175E-37 - 3.402 823 47E+38 aralığında kayan noktalı sayı; ondalık basamaklar, isteğe bağlı işaret, isteğe bağlı kesirli basamaklar ve isteğe bağlı üstel basamaklar kullanılarak ifade edilir.
r8	2.225E-307 - 1.797 693 134 862 3E+308 ondalık basamaklar, isteğe bağlı işaret, isteğe bağlı kesirli basamaklar ve isteğe bağlı üstel basamaklar kullanılarak ifade edilen kayan noktalı sayı.

Özellik değerleri: Bir özelliğin değeri, zorunlu bir ilişkili çıkış sırası olan özel karakterler dışında, herhangi bir karakterden oluşabilir. Aşağıdaki çizelgede "zorunlu" olarak işaretlenmiş bir karakterin değerindeki her bir oluşum, karşılık gelen kaçış sırasıyla değiştirilmelidir. Çizelge, isteğe bağlı bir kaçış dizisi olan karakterleri de gösterir. "İsteğe bağlı" olarak işaretli bir karakterin değerindeki her bir geçiş, karşılık gelen kaçış sırasıyla değiştirilebilir, ancak bu gerekli değildir.

Çizelge 722. Çıkış karakteri eklenmiş karakterler ve bunların kullanımı		
Karakter	Çıkış sırası	Kullanım
&	& ;	Zorunlu
<	<	Zorunlu
>	> ;	İsteğe Bağlı
"	" ;	İsteğe Bağlı
'	' ;	İsteğe Bağlı

Not: Bir çıkış dizisinin başındaki & karakteri & ; ile değiştirilmemelidir.

Aşağıdaki örnekte, değerdeki boşluklar anlamlıdır; ancak, çıkış sırası gerekmez:

```
<Famous_Words>The program displayed "Hello World"</Famous_Words>
```

RF2NVL (10 basamaklı işaretli tamsayı)

RF2NVDuzunluğu.

Bu, RF2NVD alanındaki verilerin bayt cinsinden uzunluğunu belirtir. RF2NVD alanını izleyen verilerin (varsa) veri dönüştürme işlemiyle ilgili sorunları önlemek için RF2NVL , dört verinin katı olmalıdır.

Not: RF2NVL ve RF2NVD alanları isteğe bağlıdır, ancak varsa çift olarak oluşmaları ve bitişik olmaları gerekir. Alan çifti gerektiği kadar yinelenebilir, örneğin:

```
length1 data1 length2 data2 length3 data3
```

Bu alanlar isteğe bağlı olduğundan, desteklenen çeşitli programlama dilleri için sağlanan yapının bildirimlerinden çıkarılır.

RF2SID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

RFSIDV

Kurallar ve biçimlendirme üstbilgi yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri RFSIDV 'dir.

RF2VER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

RFVER2

Version-2 kuralları ve üstbilgi yapısını biçimlendirme.

Bu alanın ilk değeri RFVER2' dir.

Başlangıçtaki değerler

Çizelge 723. MQRFH2 içindeki alanlar		
Alan adı	Değişmez adı	Değişmez değeri
RF2SID	RFSIDV	'RFH↵'
RF2VER	RFVER2	2
RF2LEN	RFLLEN2	36
RF2ENC	GİRİŞ	Ortama bağlıdır
RF2CSI	CSINHT	-2
RF2FMT	FMNONE	Boşluklar
RF2FLG	RFNONE	0
RF2NVC	Yok	1208

Notlar:

1. ↵ simgesi, tek bir boş karakteri temsil eder.

RPG bildirim

```
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 follows last NameValueData field
D RF2ENC          13     16I 0 INZ(273)
D* Character set identifier of data that follows last NameValueData field
D RF2CSI          17     20I 0 INZ(-2)
D* Format name of data that follows last NameValueData field
D RF2FMT          21     28   INZ(' ')
D* Flags
D RF2FLG          29     32I 0 INZ(0)
D* Character set identifier of NameValueData
D RF2NVC          33     36I 0 INZ(1208)

```

IBM i

IBM i üzerinde MQRMH (Başvuru iletisi üstbilgisi)

MQRMH yapısı, başvuru iletisi üstbilgisinin biçimini tanımlar.

Genel Bakış

Amaç: Bu üstbilgi, büyük hacimli verileri ("toplu veri" olarak adlandırılır) göndermek için kullanıcı tarafından yazılan ileti kanalı çıkışlarıyla birlikte kullanılır. bir kuyruk yöneticisinden diğerine. Normal ileti sistemiyle karşılaştırıldığında, toplu veriler bir kuyrukta saklanmaz; bunun yerine, kuyrukta yalnızca toplu verilere ilişkin bir *başvuru* saklanır. Bu, IBM MQ kaynaklarının birkaç büyük ileti tarafından tükenme olasılığını azaltır.

Biçim adı: FMRMH.

Karakter kümesi ve kodlama: MQRMH ' deki karakter verileri ve görelî konum alanları tarafından adreslenen dizgiler, yerel kuyruk yöneticisinin karakter kümesinde olmalıdır; bu, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından verilir. MQRMH ' deki sayısal veriler yerel makine kodlamasında olmalıdır; bu, C programlama dili için ENNAT değeriyle verilir.

MQRMH ' nin karakter takımı ve kodlaması aşağıdaki alanlardaki *MDCSI* ve *MDENC* alanlarına ayarlanmalıdır:

- MQMD (MQRMH yapısı ileti verilerinin başındadır) ya da
- MQRMH yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Kullanım: Bir uygulama, MQRMH ' den oluşan bir iletiyi koyar, ancak toplu verileri atlar. İleti, bir ileti kanalı aracı (MCA) tarafından iletim kuyruğundan okunduğunda, başvuru iletisi üstbilgisini işlemek için kullanıcı tarafından sağlanan bir ileti çıkışı çağrılır. Çıkış, MCA iletiyi kanal üzerinden sonraki kuyruk yöneticisine göndermeden önce, MQRMH yapısıyla tanıtılan toplu verilerin başvuru iletisinin sonuna eklenebilir.

Alma sonunda, başvuru iletilerini bekleyen bir ileti çıkışı olmalıdır. Bir başvuru iletisi alındığında, çıkış nesneyi iletideki MQRMH ' yi izleyen toplu verilerden yaratmalı ve daha sonra, toplu veriler olmadan başvuru iletisini geçirmelidir. Başvuru iletisi, daha sonra bir kuyruktan başvuru iletisini (toplu veri olmadan) okuyan bir uygulama tarafından alınabilir.

Normalde, MQRMH yapısı iletideki tek yapıdır. Ancak, ileti bir iletim kuyruğuyrsa, MQRMH yapısından önce bir ya da daha çok ek üstbilgi gelir.

Dağıtım listesine bir başvuru iletisi de gönderilebilir. Bu durumda, ileti bir iletim kuyruğunda olduğunda MQDH yapısı ve ilgili kayıtları MQRMH yapısından önce gelir.

Not: Bir başvuru iletisi bölümlenmiş ileti olarak gönderilmemelidir; ileti çıkışı iletiyi doğru olarak işleyemez.

- [“Veri dönüştürme” sayfa 1181](#)
- [“Alanlar” sayfa 1181](#)
- [“Başlangıçtaki değerler” sayfa 1185](#)
- [“RPG bildirim” sayfa 1186](#)

Veri dönüştürme

Veri dönüştürme amacıyla, MQRMH yapısının dönüştürülmesi, kaynak ortam verilerinin, kaynak nesne adının, hedef ortam verilerinin ve hedef nesne adının dönüştürülmesini içerir. Yapının başlangıcını izleyen *RMLen* bayt içindeki diğer tüm baytlar atılır ya da veri dönüştürmeden sonra tanımlanmamış değerlere sahip olur. Aşağıdaki deyimlerin tümünün doğru olması koşuluyla toplu veriler dönüştürülür:

- Veri dönüştürme işlemi gerçekleştirildiğinde iletide toplu veri bulunur.
- MQRMH ' deki *RMFMT* alanında *FMNONE* dışında bir değer var.
- Belirtilen biçim adına sahip, kullanıcı tarafından yazılan bir veri dönüştürme çıkışı var.

Ancak, ileti kuyruktayken genellikle toplu verilerin iletide bulunmadığını ve bunun sonucunda toplu verilerin *GMCONV* seçeneği tarafından dönüştürülmeyeceğini unutmayın.

Alanlar

MQRMH yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

RMCSI (10 basamaklı işaretli tamsayı)

Toplu verilerin karakter kümesi tanıtıcısı.

Bu, toplu verilerin karakter kümesi tanıtıcısını belirler; MQRMH yapısındaki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değer kullanılabilir:

CSINHT

Bu yapının karakter kümesi tanıtıcısını edin.

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı CSINHT değerini döndürmez.

MQMD ' deki *MDPAT* alanının değeri *ATBRKR* ise, CSINHT kullanılamaz.

Bu alanın ilk değeri *CSUNDF* 'dir.

RMDEL (10 basamaklı işaretli tamsayı)

Hedef ortam verilerinin uzunluğu.

Bu alan sıfırsa, hedef ortam verileri yoktur ve *RMDEO* yoksayıdır.

RMDEO (10 basamaklı işaretli tamsayı)

Hedef ortam verilerinin görel konumu.

Bu alan, MQRMH yapısının başlangıcından itibaren hedef ortam verilerinin görel konumunu belirler. Hedef ortam verileri, yaratan tarafından biliniyorsa, başvuru iletisini oluşturan tarafından belirtilebilir. Örneğin, hedef ortam verileri, toplu verilerin saklanacağı nesnenin izin yolu olabilir. Ancak, yaratan hedef ortam verilerini bilmiyorsa, gerekli ortam bilgilerini belirlemek kullanıcı tarafından sağlanan ileti çıkışının sorumluluğundadır.

Hedef ortam verilerinin uzunluğu *RMDEL* ; Bu uzunluk sıfır ise, hedef ortam verileri yoktur ve *RMDEO* yoksayıdır. Varsa, hedef ortam verileri, yapının başlangıcından itibaren *RMLen* bayt içinde tam olarak bulunmalıdır.

Uygulamalar, hedef ortam verilerinin *RMSEO*, *RMSNOve* *RMDNO* alanları tarafından adreslenen herhangi bir veriyle bitişik olduğunu varsaymamalıdır.

Bu alanın ilk değeri 0 'dir.

RMDL (10 basamaklı işaretli tamsayı)

Toplu verilerin uzunluğu.

RMDL alanı, MQRMH yapısının gönderme yaptığı toplu verilerin uzunluğunu belirtir.

İletide toplu veri varsa, veriler MQRMH yapısının başlangıcından itibaren *RMLen* baytlık bir görelî konumla başlar. Tüm iletinin uzunluğu eksi *RMLen* , var olan toplu verilerin uzunluğunu verir.

İletide veri varsa, *RMDL* ilgili verilerin miktarını belirtir. Normal durum, *RMDL* ' in iletide bulunan verilerin uzunluğuyla aynı değere sahip olması içindir.

MQRMH yapısı, nesnedeki geri kalan verileri (belirlenen mantıksal görelî konumdan başlayarak) gösteriyorsa, iletide toplu veri yoksa, *RMDL* için sıfır değeri kullanılabilir.

Veri yoksa, MQRMH ' nin sonu iletinin sonuna denk gelir.

Bu alanın ilk değeri 0 'dır.

RMDNL (10 basamaklı işaretli tamsayı)

Hedef nesne adının uzunluğu.

Bu alan sıfırda, hedef nesne adı yoktur ve *RMDNO* yoksayılr.

RMDNO (10 basamaklı işaretli tamsayı)

Hedef nesne adının görelî konumu.

Bu alan, hedef nesne adının MQRMH yapısının başlangıcından başlayarak görelî konumunu belirler. Hedef nesne adı, bu verileri yaratan kişi biliyorsa, başvuru iletisini oluşturan kişi tarafından belirtilebilir. Ancak, yaratan hedef nesne adını bilmiyorsa, yaratılacak ya da değiştirilecek nesneyi tanımlamak kullanıcı tarafından sağlanan ileti çıkışının sorumluluğundadır.

Hedef nesne adının uzunluğu *RMDNL* ; Bu uzunluk sıfır ise, hedef nesne adı yoktur ve *RMDNO* yoksayılr. Varsa, hedef nesne adı, yapının başlangıcından itibaren *RMLen* bayt içinde tam olarak bulunmalıdır.

Uygulamalar, hedef nesne adının *RMSEO*, *RMSNO* ve *RMDEO* alanları tarafından adreslenen herhangi bir veriyle bitişik olduğunu varsaymamalıdır.

Bu alanın ilk değeri 0 'dır.

RMDO (10 basamaklı işaretli tamsayı)

Toplu verilerin düşük görelî konumu.

Bu alan, toplu veri formunun parçası olduğu nesnenin başlangıcından itibaren toplu verilerin düşük görelî konumunu belirtir. Toplu verilerin nesnenin başlangıcındaki görelî konumu *mantıksal görelî konum* olarak adlandırılır. Bu, *RMLen* tarafından verilen MQRMH yapısının başlangıcındaki toplu verilerin fiziksel görelî konumu değildir.

Büyük nesnelerin başvuru iletileri kullanılarak gönderilmesini sağlamak için, mantıksal görelî konum iki alana bölünür ve gerçek mantıksal görelî konum şu iki alanın toplamıyla verilir:

- *RMDO* , mantıksal görelî konum 1 000 000 000 000 'e bölündüğünde elde edilen kalanı temsil eder. Bu nedenle, 0-999 999 999 999 aralığında bir değerdir.
- *RMDO2* , mantıksal görelî konum 1 000 000 000 000 'e bölündüğünde elde edilen sonucu gösterir. Bu nedenle, mantıksal görelî konumda var olan 1 000 000 000 000 'in tam katlarının sayısıdır. Çarpı sayısı, 0-999 999 999 999 aralığındadır.

Bu alanın ilk değeri 0 'dır.

RMDO2 (10 basamaklı işaretli tamsayı)

Toplu verilerin yüksek görelî konumu.

Bu alan, toplu veri parçasının oluşturulduğu nesnenin başlangıcından itibaren toplu verilerin yüksek görelî konumunu belirtir. Bu değer, 0-999 999 999 999 aralığında bir değerdir. Ayrıntılar için bkz. *RMDO* .

Bu alanın ilk değeri 0 'dır.

RMENC (10 basamaklı işaretli tamsayı)

Toplu verilerin sayısal kodlaması.

Bu, toplu verilerin sayısal kodlamasını belirtir; MQRMH yapısındaki sayısal veriler için geçerli değildir. MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir.

Bu alanın ilk değeri ENNAT ' tır.

RMFLG (10 basamaklı işaretli tamsayı)

Başvuru iletisi işaretleri.

Şu işaretler tanımlandı:

RMLAST

Başvuru iletisi nesnenin son kısmını içerir ya da gösterir.

Bu işaret, başvuru iletisinin başvurulmuş nesnenin son kısmını gösterdiğini ya da içerdiğini gösterir.

RMNLST

Başvuru iletisi, nesnenin son bölümünü içermiyor ya da temsil etmiyor.

RMNLST, program belgelerine yardımcı olmak için tanımlanır. Bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bu alanın ilk değeri RMNLST ' dir.

RMFMT (8 baytlık karakter dizgisi)

Toplu verilerin biçim adı.

Bu, toplu verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Bu alanı kodlamaya ilişkin kurallar, MQMD ' deki *MDFMT* alanıyla aynıdır.

Bu alanın ilk değeri FMNONE.

RMLEN (10 basamaklı işaretli tamsayı)

Değişmez alanların sonundaki dizgiler de içinde olmak üzere toplam MQRMH uzunluğu, toplu veriler değil.

Bu alanın ilk değeri sıfırdır.

RMOII (24 baytlık bit dizgisi)

Nesne eşgörünümü tanıtıcısı.

Bu alan, bir nesnenin belirli bir örneğini tanımlamak için kullanılabilir. Gerekli değilse, aşağıdaki değere ayarlanmalıdır:

OIINON

Nesne eşgörünümü tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

Bu alanın uzunluğu LNOIID tarafından verilir. Bu alanın ilk değeri OIINON.

RMOT (8 baytlık karakter dizgisi)

Nesne tipi.

Bu, desteklediği başvuru iletisi tiplerini tanımak için ileti çıkışı tarafından kullanılacak bir addır. Adı, *RMFMT* alanıyla aynı kurallara uygun hale getirmeyi düşünün.

Bu alanın ilk değeri 8 boşluktur.

RMSEL (10 basamaklı işaretli tamsayı)

Kaynak ortam verilerinin uzunluğu.

Bu alan sıfırsa, kaynak ortam verileri yoktur ve *RMSEO* yoksayılr.

Bu alanın ilk değeri 0 'dır.

RMSEO (10 basamaklı işaretli tamsayı)

Kaynak ortam verilerinin görelı konumu.

Bu alan, MQRMH yapısının başlangıcından itibaren kaynak ortam verilerinin görelı konumunu belirler. Kaynak ortam verileri, yaratan tarafından biliniyorsa, başvuru iletisini oluşturan tarafından belirtilebilir. Örneğın, kaynak ortam verileri, toplu verileri içeren nesnenin dizin yolu olabilir. Ancak, yaratan, kaynak ortam verilerini bilmiyorsa, gerekli ortam bilgilerini belirlemek kullanıcı tarafından sağlanan ileti çıkışının sorumluluğundadır.

Kaynak ortam verilerinin uzunluđu *RMSEL* ; Bu uzunluk sıfır ise, kaynak ortam verileri yoktur ve *RMSEO* yoksayılr. Varsa, kaynak ortam verileri, yapının başlangıcından itibaren *RMLLEN* bayt içinde tam olarak bulunmalıdır.

Uygulamalar, ortam verilerinin yapıdaki son sabit alandan hemen sonra başladığını ya da *RMSNO*, *RMDEOve* *RMDNO* alanları tarafından adreslenen herhangi bir veriyle bitişik olduğunu varsaymamalıdır.

Bu alanın ilk değeri 0 'dır.

RMSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

RMSIDV

Başvuru iletisi üstbilgi yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri RMSIDV 'dir.

RMSNL (10 basamaklı işaretli tamsayı)

Kaynak nesne adının uzunluđu.

Bu alan sıfırsa, kaynak nesne adı yoktur ve *RMSNO* yoksayılr.

Bu alanın ilk değeri 0 'dır.

RMSNO (10 basamaklı işaretli tamsayı)

Kaynak nesne adının görelı konumu.

Bu alan, kaynak nesne adının MQRMH yapısının başlangıcından başlayarak görelı konumunu belirler. Kaynak nesne adı, verileri yaratan tarafından biliniyorsa, başvuru iletisini oluşturan tarafından belirtilebilir. Ancak, yaratan kaynak nesne adını bilmiyorsa, erişilecek nesneyi tanımlamak kullanıcı tarafından sağlanan ileti çıkışının sorumluluğundadır.

Kaynak nesne adının uzunluđu *RMSNL* ; Bu uzunluk sıfır ise, kaynak nesne adı yoktur ve *RMSNO* yoksayılr. Varsa, kaynak nesne adı, yapının başlangıcından itibaren *RMLLEN* bayt içinde tam olarak bulunmalıdır.

Uygulamalar, kaynak nesne adının *RMSEO*, *RMDEOve* *RMDNO* alanları tarafından adreslenen herhangi bir veriyle bitişik olduğunu varsaymamalıdır.

Bu alanın ilk değeri 0 'dır.

RMVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

RMVER1

Version-1 başvuru iletisi üstbilgi yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

RMVERC

Başvuru iletisi üstbilgi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri RMVER1' dir.

Başlangıçtaki değerler

Çizelge 724. MQRMH ' deki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
RMSID	RMSIDV	' RMH↵ '
RMVER	RMVER1	1
RMLN	Yok	0
RMENC	GİRİŞ	Ortama bağlıdır
RMCSI	CSUNDF	0
RMFMT	FMNONE	Boşluklar
RMFLG	RMNLST	0
RMOT	Yok	Boşluklar
RMOII	OIINON	Boş Değerler
RMSEL	Yok	0
RMSEO	Yok	0
RMSNL	Yok	0
RMSNO	Yok	0
RMDEL	Yok	0
RMDEO	Yok	0
RMDNL	Yok	0
RMDNO	Yok	0
RMDL	Yok	0
RMDO	Yok	0
RMDO2	Yok	0

Notlar:

- ↵ simgesi, tek bir boş karakteri temsil eder.

```
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, includingstrings at end of fixed fields,but not
D* the bulk data
D RMLN 9 12I 0 INZ(0)
D* Numeric encoding of bulk data
D RMENC 13 16I 0 INZ(273)
D* Character set identifier of bulkdata
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                                     000000000000000000-
D                                     000000000000')
D* Length of source environmentdata
D RMSEL          65      68I 0 INZ(0)
D* Offset of source environmentdata
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 environmentdata
D RMDL           81      84I 0 INZ(0)
D* Offset of destination environmentdata
D RMDEO          85      88I 0 INZ(0)
D* Length of destination objectname
D RMDNL          89      92I 0 INZ(0)
D* Offset of destination objectname
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)

```

RPG bildirimi

IBM i IBM i üzerinde MQRR (Yanıt kaydı)

MQRR yapısı, hedef bir dağıtım listesi olduğunda, tek bir hedef kuyruğa ilişkin açma ya da yerleştirme işleminden kaynaklanan tamamlanma kodunu ve neden kodunu almak için kullanılır.

Genel Bakış

Amaç: MQRR, MQOPEN, MQPUT ve MQPUT1 çağrılarına ilişkin bir çıkış yapısıdır.

Karakter kümesi ve kodlama: MQRR ' deki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

Kullanım: MQOPEN ve MQPUT çağrılarında ya da MQPUT1 çağrısında bu yapılardan oluşan bir dizi sağlanarak, bir dağıtım listesindeki tüm kuyrukların tamamlanma kodları ve neden kodları, çağrı sonucu karışık olduğunda (yani, çağrı listedeki bazı kuyruklar için başarılı olduğunda ancak diğerleri için başarısız olduğunda) saptanabilir. Çağrıdaki RC2136 neden kodu, yanıt kayıtlarının (uygulama tarafından sağlandıysa) kuyruk yöneticisi tarafından ayarlandığını gösterir.

- [“Alanlar” sayfa 1186](#)
- [“Başlangıçtaki değerler” sayfa 1187](#)
- [“RPG bildirimi” sayfa 1187](#)

Alanlar

MQRR yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırad**a açıklanır:

RRCC (10 basamaklı işaretli tamsayı)

Kuyruk için tamamlanma kodu.

Bu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki ilgili ögenin belirttiği adı taşıyan kuyruğa ilişkin açma ya da koyma işleminden kaynaklanan tamamlanma kodudur.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri CCOK ' dur.

RRREA (10 basamaklı işaretli tamsayı)

Kuyruğa ilişkin neden kodu.

Bu neden kodu, MQOPEN ya da MQPUT1 çağrısında sağlanan MQOR yapıları dizisindeki karşılık gelen ögenin belirttiği adı taşıyan kuyruğa ilişkin açma ya da koyma işleminin sonucudur.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri RCNONE.

Başlangıçtaki değerler

Çizelge 725. MQRR ' deki Alanlar		
Alan adı	Değişmez adı	Değişmez değeri
RRCC	CCOK	0
RRREA	RCNONE	0

RPG bildirim

```
D*..1.....2.....3.....4.....5.....6.....7..
D*
D* MQRR 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)
```

IBM i üzerinde MQSCO (TLS yapılandırma seçenekleri)

MQSCO yapısı (MQCD yapısında TLS alanları olan), IBM MQ MQI client olarak çalışan bir uygulamanın kanal protokolü TCP/IP olduğunda istemci bağlantısı için TLS kullanımını denetleyen yapılandırma seçeneklerini belirtmesine olanak sağlar.

Genel Bakış

Amaç: Yapı, MQCONNX çağrısında bir giriş değiştirgesidir.

İstemci kanalına ilişkin kanal protokolü TCP/IP değilse, MQSCO yapısı yoksayılır.

Karakter kümesi ve kodlama: MQSCO ' daki veriler, **CodedCharSetId** kuyruk yöneticisi özniteliği ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır.

- [“Alanlar” sayfa 1187](#)
- [“Başlangıçtaki değerler” sayfa 1192](#)
- [“RPG bildirim” sayfa 1192](#)

Alanlar

MQSCO yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

SCAIC (10 basamaklı işaretli tamsayı)

Bu, SCAIP ya da SCAIO alanları tarafından adreslenen kimlik doğrulama bilgileri (MQAIR) kayıtlarının sayısıdır. Daha fazla bilgi için, bkz. [“IBM i üzerinde MQAIR \(Kimlik doğrulama bilgileri kaydı\)” sayfa](#)

999. Değer sıfır ya da daha büyük olmalıdır. Değer geçerli değilse, çağrı RC2383neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

SCAIO (10 basamaklı işaretli tamsayı)

Bu, MQSCO yapısının başlangıcından itibaren ilk kimlik doğrulama bilgileri kaydının bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir. SCAIC sıfırda alan yoksayılr.

MQAIR kayıtlarını belirtmek için SCAIO ya da SCAIP kullanabilirsiniz, ancak her ikisini birden kullanamazsınız; ayrıntılar için SCAIP alanının tanımına bakın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dır.

SCAIP (10 basamaklı işaretli tamsayı)

Bu, ilk kimlik doğrulama bilgileri kaydının adresidir. SCAIC sıfırda alan yoksayılr.

MQAIR kayıtları dizisini aşağıdaki iki yoldan biriyle sağlayabilirsiniz:

- SCAIP gösterge alanını kullanarak

Bu durumda, uygulama MQSCO yapısından ayrı bir MQAIR kayıtları dizisi bildirebilir ve SCAIP dizisini dizinin adresine ayarlayabilir.

İşaretçi veri tipini destekleyen programlama dilleri için farklı ortamlarda (örneğin, C programlama dili) taşınabilir bir şekilde SCAIP ' i kullanmayı düşünün.

- SCAIO görelî konum alanını kullanarak

Bu durumda, uygulama MQSCO içeren bir bileşik yapıyı ve ardından MQAIR kayıtları dizisini bildirmeli ve SCAIO ögesini MQSCO yapısının başlangıcından itibaren dizideki ilk kaydın görelî konumuna ayarlamalıdır. Bu değerîn doğru olduğundan ve MQLONG içinde tutulabilecek bir değere sahip olduğundan emin olun (en kısıtlayıcı programlama dili COBOL 'dur; geçerli aralık -999 999 999 ile 999 999 999 999 arasındadır).

İşaretçi veri tipini desteklemeyen ya da gösterge veri tipini farklı ortamlara (örneğin, COBOL programlama dili) taşınmayacak şekilde uygulayan programlama dilleri için SCAIO komutunu kullanmayı düşünün.

Hangi tekniği seçerseniz seçin, SCAIP ve SCAIO yöntemlerinden yalnızca biri kullanılabilir; her ikisi de sıfır değilse, çağrı RC2384 neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri, göstergeleri destekleyen programlama dillerindeki boş değeri gösterge ve tersi durumda tümü boş değeri bir bayt dizgisidir.

Not: Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir.

SCCERLBL (10 basamaklı işaretli tamsayı)

Bu alan, kullanılmakta olan sertifika etiketinin ayrıntılarını verir.

IBM MQ , SCCERLBL alanının değerini boşluk olarak başlatır. Gerekli değeri girin ya da varsayılan değeri kabul edin.

ibmwebsphereuser_id , bu alan için ürünün tüm sürümleri için geçerli bir değerdir ve 5.0 ' dan küçük MQSCO sürümleri için tek geçerli değerdir. Bu nedenle, bu alanın değeri yürütme sırasında yorumlanır ve gerekirse değiştirilir. 5.0' dan küçük bir MQSCO sürümü belirtirseniz ya da SCCERLBL alanı için varsayılan değer olan boşlukları kabul ederseniz, sistem ibmwebsphereuser_id değerini kullanır.

Bu bir giriş alanıdır.

SCCERTVPOL (10 basamaklı işaretli tamsayı)

Bu alan, hangi sertifika doğrulama ilkesinin kullanıldığını belirtir. Alan aşağıdaki değerlerden birine ayarlanabilir:

MQ_CERT_VAL_POLICY_ANY

Güvenli yuva kitaplığı tarafından desteklenen sertifika geçerlilik denetimi ilkelerinin her birini uygulayın. İlkelerden herhangi biri sertifika zincirini geçerli sayarsa sertifika zincirini kabul edin.

MQ_CERT_VAL_POLICY_RFC5280

Yalnızca RFC5280 uyumlu sertifika doğrulama ilkesini uygulayın. Bu ayar, HERHANGİ BİRİ ayarından daha sıkı doğrulama sağlar, ancak bazı eski dijital sertifikaları reddeder.

Bu alanın ilk değeri: MQ_CERT_VAL_POLICY_ANY

SCCH (10 basamaklı işaretli tamsayı)

Bu alanda, istemci sistemine bağlı şifreleme donanımına ilişkin yapılandırma ayrıntıları verilir.

Alanı aşağıdaki biçimde bir dizgiye ayarlayın ya da boş bırakın:

```
GSK_PKCS11=the PKCS #11 driver path and file name;the PKCS #11  
token label;the PKCS #11 token password;symmetric cipher setting>;
```

PKCS11 arabirimine uyan şifreleme donanımını kullanmak için, örneğin, IBM 4960 ya da IBM 4963, PKCS11 sürücü yolunu, PKCS11 simge etiketini ve PKCS11 simge parolası dizgilerini belirtin; bunların her biri noktalı virgülle sonlandırılmış olarak.

PKCS #11 sürücü yolu, PKCS #11 kartı için destek sağlayan paylaşılan kitaplığın mutlak yoludur. PKCS #11 sürücü dosyası adı, paylaşılan kitaplığın adıdır. PKCS #11 yolu ve dosya adı için gereken değerler bir örneği:

```
/usr/lib/pkcs11/PKCS11_API.so
```

PKCS #11 simge etiketi tamamen küçük harfli olmalıdır. Donanımınızı büyük ya da küçük harf karışık bir simge etiketiyle yapılandırdıysanız, bu küçük harfli etiketle yeniden yapılandırın.

Şifreleme donanımı yapılandırması gerekmiyorsa, alanı boş ya da boş olarak ayarlayın.

Değer alanın uzunluğundan kısaysa, değeri boş bir karakterle sonlandırın ya da alanın uzunluğuna kadar boşluklarla doldurun. Değer geçerli değilse ya da şifreleme donanımını yapılandırmak için kullanıldığında bir hataya neden olursa, çağrı RC2382neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNSSCH tarafından verilir. Bu alanın ilk değeri boş karakterdir.

SCEPSUITEB (10 basamaklı işaretli tamsayı)

Bu alan, Suite B uyumlu şifrelemenin kullanılıp kullanılmayacağını ve hangi güç düzeyinin kullanıldığını belirtir. Değer aşağıdakilerden biri ya da daha fazlası olabilir:

- SCEPSUITEB0

Suite B uyumlu şifreleme kullanılmaz.

- SCEPSUITEB1

Suite B 128-bit güvenlik kullanılır.

- SCEPSUITEB2

Suite B 192-bit güvenlik kullanılır.

Not: SCEPSUITEB0 değerinin bu alanda başka bir değerle kullanılması geçersizdir.

SCFR (10 basamaklı işaretli tamsayı)

IBM MQ , kullanılan şifreleme modüllerinin donanım ürünü tarafından sağlanacak şekilde şifreleme donanımıyla yapılandırılabilir; bunlar, kullanılmakta olan şifreleme donanım ürününe bağlı olarak belirli bir düzeyde FIPS onaylı olabilir.

Şifreleme IBM MQ tarafından sağlanan yazılımda sağlanırsa yalnızca FIPS onaylı algoritmaların kullanılacağını belirtmek için bu alanı kullanın.

IBM MQ kurulduğunda, FIPS onaylı bazı modüller sağlayan bir TLS şifreleme uygulaması da kurulur.

Değerler şunlar olabilir:

MQSSL_FIPS_NO

Bu varsayılan değerdir. Bu değere ayarlandığında:

- Belirli bir platformda desteklenen herhangi bir CipherSpec kullanılabilir.
- Şifreleme donanımı kullanılmadan çalıştırılırsa, IBM MQ platformlarında FIPS 140-2 sertifikalı şifreleme kullanılarak aşağıdaki CipherSpecs çalıştırılır:
 - TLS_RSA_WITH_3DES_EDE_CBC_SHA
 - TLS_RSA_WITH_AES_128_CBC_SHA
 - TLS_RSA_WITH_AES_256_CBC_SHA

MQSSL_FIPS_YES

Bu değere ayarlandığında, şifrelemeyi gerçekleştirmek için şifreleme donanımı kullanmıyorsanız,

- Bu istemci bağlantısına uygulanan CipherSpec içinde yalnızca FIPS onaylı şifreleme algoritmaları kullanılabilir.
- Gelen ve giden TLS kanal bağlantıları, yalnızca aşağıdaki Şifre Belirtilimlerinden biri kullanıldığında başarılı olur:
 - TLS_RSA_WITH_3DES_EDE_CBC_SHA
 - TLS_RSA_WITH_AES_128_CBC_SHA
 - TLS_RSA_WITH_AES_256_CBC_SHA

Notlar:

1. **Deprecated** CipherSpec TLS_RSA_WITH_3DES_EDE_CBC_SHA kullanımdan kaldırılmıştır.
2. Mümkünse, yalnızca FIPS CipherSpecs yapılandırılırsa, MQI istemcisi FIPS CipherSpec withRC2393 olmayan bağlantıları reddeder. IBM MQ , bu tür bağlantıların tümünü reddetmeyi garanti etmez ve IBM MQ yapılandırmanızın FIPS uyumlu olup olmadığını belirlemek sizin sorumluluğunuzdadır.

V 9.3.0

V 9.3.0

SCKEYPWL (10 basamaklı işaretli tamsayı)

Bu, TLS anahtar havuzu parolası uzunluğudur.

Anahtar havuzu parolası uzunluğu üst sınırı 128 karakterdir. Anahtar havuzu parolası izin verilen uzunluk üst sınırından fazlaysa, bağlantı RC2381 ile başarısız olur.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir.

V 9.3.0

V 9.3.0

SCKEYPWO (10 basamaklı işaretli tamsayı)

Bu, TLS anahtar havuzu geçiş tümcesinin bayt cinsinden görelî konumudur. Görelî konum pozitif ya da negatif olabilir.

Anahtar havuzu parolasını belirtmek için SCKEYPWO ya da SCKEYPWP kullanabilirsiniz, ancak ikisini birden kullanamazsınız. Daha fazla bilgi için SCKEYPWP alanının açıklamasına bakın.

Bu bir giriş alanıdır. Bu alanın ilk değeri 0 'dir.

V 9.3.0

V 9.3.0

SCKEYPWP (işaretçi)

Bu, TLS anahtar havuzu parolası adresidir.

Bu bir giriş alanıdır. Bu alanın ilk değeri boş işaretçidir.

Anahtar havuzu parolası, **runmqicred** yardımcı programı kullanılarak şifrelenmiş bir parola ya da düz metin dizgisi olarak belirtilebilir.

Bu alan kullanılarak belirtilen anahtar havuzu parolası, **MQKEYRPWD** ortam değişkeni ya da istemci yapılandırma dosyasının SSL kısmına ilişkin **SSLKeyRepositoryPassword** özelliği kullanılarak belirtilen anahtar havuzu parolasını geçersiz kılar.

Anahtar havuzu parolasını belirtmek için SCKEYPW0 ya da SCKEYPWP kullanabilirsiniz, ancak ikisini birden kullanamazsınız.



SCKR (10 basamaklı işaretli tamsayı)

Bu alan, anahtarların ve sertifikaların saklandığı anahtar veritabanı dosyasının yerini belirtir.

  Dosya soneki belirtilmezse, otomatik olarak .kdb soneki eklenir.

Her anahtar veritabanı dosyasının ilişkili bir *parola saklama kütüğü* olabilir. Bu, anahtar veritabanına programlı erişime izin vermek için kullanılan şifrelenmiş parolaları tutar. Parola saklama kütüğü aynı dizinde bulunmalı ve anahtar veritabanıyla aynı kütük sistemine sahip olmalı ve .sthsonekiyle bitmelidir.

Örneğin, anahtar veritabanı dosyası /xxx/yyy/key.kdbise, parola saklama dosyası /xxx/yyy/key.sth olmalıdır; burada xxx ve yyy dizin adlarını gösterir.

  Anahtar veritabanı parolası SCKEYPWP ya da SCKEYPW0 alanları kullanılarak da belirtilebilir.

Değer alanın uzunluğundan kısaysa, değeri boş bir karakterle sonlandırın ya da alanın uzunluğuna kadar boşluklarla doldurun. Değer denetlenmez; anahtar havuzuna erişilirken hata oluşursa, çağrı RC2381neden koduyla başarısız olur.

Bir IBM MQ MQI client'inden TLS bağlantısı çalıştırmak için, SCKR değerini geçerli bir anahtar veritabanı dosyası adına ayarlayın.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNSSKR tarafından verilir. Bu alanın ilk değeri boş bir karakterdir.

SCSID (10 basamaklı imzalı tamsayı)

Bu yapı tanıttırıcıdır; değer şu olmalıdır:

SCSIDV

TLS yapılandırma seçenekleri yapısına ilişkin tanıttırıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri SCSIDV'dir.

SCVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

SCVER1

Version-1 TLS yapılandırma seçenekleri yapısı.

SCVER2

Version-2 TLS yapılandırma seçenekleri yapısı.

SCVER3

Version-3 TLS yapılandırma seçenekleri yapısı.

SCVER4

Version-4 TLS yapılandırma seçenekleri yapısı.

SCVER5

Version-5 TLS yapılandırma seçenekleri yapısı.

  **SCVER6**

Version-6 TLS yapılandırma seçenekleri yapısı.







Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

SCVERC

TLS yapılandırma seçenekleri yapısının geçerli sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri SCVER1'dir.

Başlangıçtaki değerler

Çizelge 726. MQSCO 'daki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
SCSID	SCSIDV	'SCO~'
SCVER	SCVER1	1
SCKR	Yok	Boş değerli dizilim ya da boşluklar
SCCH	Yok	Boş değerli dizilim ya da boşluklar
SCAIC	Yok	0
SCAIO	Yok	0
SCAIP	Yok	Boş değerli gösterge ya da boş değerli byte
SCKRC	Yok	Boş değerli gösterge ya da boş değerli byte
SCFR	Yok	Boş değerli gösterge ya da boş değerli byte
SCEPSUITEB	Yok	Boş değerli gösterge ya da boş değerli byte
SCCERTVPOL	Yok	Boş değerli gösterge ya da boş değerli byte
SCCERLBL	Yok	Boş değerli gösterge ya da boş değerli byte
  SCKEYPWP	Yok	Boş değerli gösterge ya da boş değerli byte
  SCKEYPWO	Yok	0
  SCKEYPWL	Yok	0
Notlar:		
1. ~ simgesi tek bir boş karakteri temsil eder.		
2. SCEPSUITEB seçenekleri için bkz. "RPG bildirim" sayfa 1192 .		

RPG bildirim

```
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 IBM i üzerinde MQSD (Abonelik tanımlayıcısı)

MQSD yapısı, yapılmakta olan abonelik ile ilgili ayrıntıları belirtmek için kullanılır.

Genel Bakış

Amaç

Yapı, MQSUB çağrısında bir giriş/çıkış değiştirgesidir.

Yönetilen abonelikler

Bir uygulamanın belirli bir kuyruğu aboneliğiyle eşleşen yayınlar için hedef olarak kullanması gerekmiyorsa, yönetilen abonelik özelliğini kullanabilir. Bir uygulama yönetilen abonelik kullanmayı seçerse, kuyruk yöneticisi, MQSUB çağrısından çıkış olarak bir nesne tanıtcısı sağlayarak aboneyi yayınlanan iletilerin gönderildiği hedef hakkında bilgilendirir. Daha fazla bilgi için bkz. [HOBj \(10 basamaklı işaretli tamsayı\)-giriş/çıkış](#).

Abonelik kaldırıldığında, kuyruk yöneticisi aşağıdaki durumlarda yönetilen hedeften alınmayan iletileri temizlemeyi de taahhüt eder:

- Abonelik kaldırıldığında-CORMSB ile MQCLOSE kullanımı ile-ve yönetilen Hobj kapatılır.
- Sürekli olmayan bir abonelik (SONDUR) kullanılarak bir uygulamayla bağlantı kesildiğinde örtük olarak
- Süresi dolduğu ve yönetilen Hobj kapatıldığı için bir abonelik kaldırıldığında süre bitimine göre.

Kalıcı olmayan aboneliklerle yönetilen abonelikleri kullanmalısınız; böylece temizleme işlemi gerçekleştirilebilir ve böylece, kapalı kalıcı olmayan aboneliklere ilişkin iletiler kuyruk yöneticinizde yer kaplamaz. Sürekli abonelikler, yönetilen hedefleri de kullanabilir.

Karakter kümesi ve kodlama

MQSD 'deki veriler, ENNAT tarafından verilen yerel kuyruk yöneticisinin **CodedCharSetId** kuyruk yöneticisi özniteliği ve kodlaması tarafından belirtilen karakter kümesinde olmalıdır. Ancak, uygulama bir IBM MQ istemcisi olarak çalışıyorsa, yapı istemcinin karakter kümesinde ve kodlamasında olmalıdır.

- [“Alanlar” sayfa 1194](#)
- [“Başlangıçtaki değerler” sayfa 1205](#)
- [“RPG bildirim” sayfa 1206](#)

Alanlar

MQSD yapısı aşağıdaki alanları içerir; alanlar alfabetik sırayla açıklanır:

SDAID (32 baytlık karakter dizilimi)

Bu değer, bu abonelikte eşleşen tüm yayın iletilerinin İleti Tanımlayıcı (MQMD) *MDAID* alanında bulunur. *SDAID*, iletinin kimlik bağlamının bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

MDAID hakkında daha fazla bilgi için bkz. [MDAID](#).

SOSETI seçeneği belirtilmezse, varsayılan bağlam bilgileri olarak, bu abonelik için yayınlanan her iletide ayarlanan *MDAID* boş olur.

SOSETI seçeneği belirtilirse, *SDAID* kullanıcı tarafından oluşturulur ve bu alan, bu abonelik için her yayında ayarlanacak *MDAID* değerini içeren bir giriş alanıdır.

Bu alanın uzunluğu LNAIDD tarafından verilir. Bu alanın ilk değeri 32 boş karakterdir.

Var olan bir abonelik SOALT seçeneğini kullanarak değiştiriliyorsa, gelecekteki yayın iletilerinin *SDAID* değeri değiştirilebilir.

SORES kullanılarak MQSUB çağrısından dönüş yapılırsa, bu alan abonelik için kullanılmakta olan *MDAID* olarak ayarlanır.

SDACC (32 baytlık karakter dizgisi)

Bu değer, bu abonelikte eşleşen tüm yayın iletilerinin İleti Tanımlayıcı (MQMD) *MDACC* alanında bulunur. *MDACC*, iletinin kimlik bağlamının bir parçasıdır. İleti bağlamıyla ilgili daha fazla bilgi için bkz. [İleti bağlamı](#).

MDACC hakkında daha fazla bilgi için bkz. [MDACC](#).

SDACC alanı için aşağıdaki özel değeri kullanabilirsiniz:

YOK

Hesap simgesi belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

SOSETI seçeneği belirtilmezse, hesap simgesi kuyruk yöneticisi tarafından varsayılan bağlam bilgileri olarak oluşturulur ve bu alan, bu abonelik için yayınlanan her iletide ayarlanan *MDACC* ögesini içeren bir çıkış alanıdır.

SOSETI seçeneği belirtilirse, muhasebe simgesi kullanıcı tarafından oluşturulur ve bu alan, bu abonelik için her yayında ayarlanacak *MDACC* değerini içeren bir giriş alanıdır.

Bu alanın uzunluğu LMACCT tarafından verilir. Bu alanın ilk değeri ACNONE' dir.

SOALT seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, ilerideki yayın iletilerinde *MDACC* değeri değiştirilebilir.

SORES kullanılarak bir MQSUB çağrısından dönüş yapılırsa, bu alan abonelik için kullanılmakta olan *MDACC* değerine ayarlanır.

SDASI (40 baytlık bit dizgisi)

Bu, uygun yetki denetimlerinin gerçekleştirilmesine izin vermek için *SDAU* ile yetkilendirme hizmetine iletilen bir güvenlik tanıtıcısıdır.

SDASI yalnızca SOALTU belirtilirse ve *SDAU* alanı ilk boş karakter ya da alanın sonuna kadar tamamen boş değilse kullanılır.

SORES kullanılarak MQSUB çağrısından dönüş yapılıyorsa, bu alan değişmez.

Daha fazla bilgi için MQOD veri tipinde [ODASI](#) açıklamasına bakın.

SDAU (12 baytlık karakter dizilimi)

SOALTU belirtilirseniz, bu alan, uygulamanın çalışmakta olduğu kullanıcı kimliği yerine, aboneliğe ve hedef kuyruğa (MQSUB çağrısının **Hobj** değiştirilmesinde belirtilir) ilişkin yetkiyi denetlemek için kullanılan diğer bir kullanıcı kimliği içerir.

Başarılı olursa, bu alanda belirtilen kullanıcı kimliği, uygulamanın çalışmakta olduğu kullanıcı kimliği yerine, abonelik sahibi kullanıcı kimliği olarak kaydedilir.

SOALTU belirtilirse ve bu alan ilk boş karaktere ya da alanın sonuna kadar tamamen boşsa, abonelik ancak belirlenen seçeneklerle ya da çıkış için hedef kuyruk ile bu konuya abone olmak için kullanıcı yetkisi gerekmediğinde başarılı olabilir.

SOALTU belirtilmezse, bu alan yoksayıdır.

SORES kullanılarak MQSUB çağrısından dönüş yapılıyorsa, bu alan değişmez.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNUID tarafından verilir. Bu alanın ilk değeri 12 boş karakterdir.

SDCID (24 baytlık bit dizgisi)

Bu abonelik eşleşmesi için gönderilen tüm yayınlar, ileti tanımlayıcısında bu ilinti tanıtıcısını içerir. Birden çok abonelik yayınlarını almak için aynı kuyruğu kullanıyorsa, MQGET ilinti tanıtıcısıyla kullanılması yalnızca belirli bir aboneliğe ilişkin yayınların elde edilmesine izin verir. Bu ilinti tanıtıcısı kuyruk yöneticisi ya da kullanıcı tarafından oluşturulabilir.

SO SCID seçeneği belirtilmezse, ilinti tanıtıcısı kuyruk yöneticisi tarafından üretilir ve bu alan, bu abonelik için yayınlanan her iletide ayarlanan ilinti tanıtıcısını içeren bir çıkış alanıdır.

SO SCID seçeneği belirtilirse, ilinti tanıtıcısı kullanıcı tarafından üretilir ve bu alan, bu abonelik için her yayında ayarlanacak ilinti tanıtıcısını içeren bir giriş alanıdır. Bu durumda, alan CINONE içeriyorsa, bu abonelik için yayınlanan her iletide ayarlanan ilinti tanıtıcısı, iletinin özgün konmasıyla yaratılan ilinti tanıtıcısıdır.

SOGRP seçeneği belirtilirse ve belirtilen ilinti tanıtıcısı, aynı kuyruk ve çakışan bir konu dizgisi kullanılarak var olan bir gruplanmış abonelik aynıysa, yayının bir kopyasıyla yalnızca gruptaki en önemli abonelik sağlanır.

Bu alanın uzunluğu LNCID tarafından verilir. Bu alanın ilk değeri CINONE' dur.

SOALT seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa ve bu alan bir giriş alanıysa, abonelik ilinti tanıtıcısı SOGRP seçeneği kullanılarak yaratılmadıysa değiştirilebilir.

SORES kullanılarak bir MQSUB çağrısından döndüğünüzde, bu alan abonelik için yürürlükteki ilinti tanıtıcısına ayarlanır.

SDEXP (10 basamaklı işaretli tamsayı)

Bu, abonelik süresinin sona ermesinden sonra saniyenin onda biri olarak ifade edilen süredir. Bu aralık geçtikten sonra başka hiçbir yayın bu abonelikle eşleşmeyecek. Bu, bu aboneye gönderilen yayınların MQMD ' deki MDEXP alanında değer olarak da kullanılır.

Şu özel değer tanınır:

EIULIM

Aboneliğin sınırsız bir süre sonu vardır.

Var olan bir abonelik SOALT seçeneği kullanılarak değiştiriliyorsa, aboneliğin süresi değiştirilebilir.

SORES seçeneği kullanılarak bir MQSUB çağrısından dönüşte bu alan, aboneliğin geri kalan süre bitimine değil, özgün süre bitimine ayarlanır.

SDON (48 baytlık karakter dizgisi)

Bu, yerel kuyruk yöneticisinde tanımlandığı şekilde, konu nesnesinin adıdır.

Ad aşağıdaki karakterleri içerebilir:

- Büyük harf alfabetik karakterler (A-Z)
- Küçük harf alfabetik karakterler (a-z)
- Sayısal basamaklar (0-9)
- Nokta (.), eğik çizgi (/), altçizgi (_), yüzde (%)

Ad başta ya da gömülü boşluklar içermemelidir, ancak sonda boşluklar içerebilir. Addaki önemli verilerin sonunu göstermek için boş bir karakter kullanın; bu karakteri izleyen karakterler ve boş değer boşluk olarak değerlendirilir. Aşağıdaki kısıtlamalar söz konusudur:

- EBCDIC Katakana kullanan sistemlerde küçük harfli karakterler kullanılamaz.
- Komutlarda belirtildiğinde, küçük harf, eğik çizgi ya da yüzde içeren adlar tırnak içine alınmalıdır. Bu tırnak işaretleri, yapılarda alan olarak ya da çağrılarda parametre olarak oluşan adlar için belirlenmemelidir.

SDON , Tam konu adını oluşturmak için kullanılır.

Tam konu adı iki farklı alandan oluşturulabilir: *SDON* ve *SDOS*. Bu iki alanın nasıl kullanıldığına ilişkin ayrıntılar için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.

SORES seçeneği kullanılarak bir *MQSUB* çağrısından dönüş yapılıyorsa, bu alan değişmez.

Bu alanın uzunluğu *LNTOPN* tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

SDALT seçeneğini kullanarak var olan bir abonelik değiştiriliyorsa, abone olunan konu nesnesinin adı değiştirilemez. Bu alan ve *SDOS* atlanabilir. Sağlanırsa, aynı tam konu adına çözümleri gerekir ya da arama *RC2510* ile başarısız olur.

SDOPT (10 basamaklı işaretli tamsayı)

Aşağıdaki seçeneklerden en az birini belirtmelisiniz:

- SOALT
- S1RALAR
- SOCRT

Değerler eklenebilir. Aynı sabiti bir kereden fazla eklemeyin. Tablo, bu seçenekleri nasıl birleştirebileceğinizi gösterir: Geçerli olmayan birleşimler not edildi; diğer birleşimler geçerlidir.

Erişim ya da yaratma seçenekleri

Erişim ve yaratma seçenekleri, aboneliğin oluşturulup oluşturulmayacağını ya da var olan bir aboneliğin döndürülüp döndürülmeyeceğini ya da değiştirilip değiştirilmeyeceğini denetler. Bu seçeneklerden en az birini belirtmelisiniz. Çizelge, geçerli erişim ya da yaratma seçenekleri birleşimlerini görüntüler.

<i>Çizelge 727. Geçerli erişim ve yaratma seçenekleri birleşimleri</i>	
Seçenek birleşimi	Notlar
SOCRT	Yoksa bir abonelik oluşturur; abonelik varsa başarısız olur.
S1RALAR	Var olan bir aboneliği sürdürür, abonelik yoksa başarısız olur.
SOCRT + S1RALAMA	Yoksa bir abonelik oluşturur ve varsa, eşleşen bir aboneliği sürdürür. Birkaç kez çalıştırılacak bir uygulamada kullanılırsa kullanışlı bir birleşim.
SORES + SOALT (nota bakın)	Var olan bir aboneliği sürdürür ve <i>MQSD</i> ' de belirtilenlerle eşleşecek şekilde alanları değiştirir, abonelik yoksa başarısız olur.

Çizelge 727. Geçerli erişim ve yaratma seçenekleri birleşimleri (devamı var)	
Seçenek birleşimi	Notlar
SOCRT + SOALT (nota bakın)	Yoksa bir abonelik yaratır ve varsa, eşleşen bir aboneliği sürdürerek, MQSD ' de belirtilen alanlarla eşleşecek şekilde alanların değiştirilmesini sağlar. Devam etmeden önce aboneliğinin belirli bir durumda olduğundan emin olmak isteyen bir uygulamada kullanışlı birleşim.

Not:

SOALT seçeneğini belirleyen seçenekler SORESdeğerini de belirleyebilir, ancak bu birleşim tek başına SOALT belirtiminin belirlenmesinde ek bir etki yaratmaz. SOALT , bir aboneliği değiştirmek için MQSUB 'nin çağrılması aboneliklerin de sürdürüldüğü anlamına geldiğinden SORES' i belirtir. Bunun tersi doğru değildir, ancak: bir aboneliğin sürdürülmesi değiştirileceği anlamına gelmez.

SOCRT

Belirtilen konu için bir abonelik oluşturun. Aynı SDSN kullanan bir abonelik varsa, çağrı RC2432 ile başarısız olur. Bu hata, SOCRT seçeneği SORESile birleştirilerek önlenir. SDSN her zaman gerekli değildir. Daha fazla ayrıntı için bu alanın açıklamasına bakın.

SOCRT 'nin SORES ile birleştirilmesi, önce belirtilen SDSNiçin var olan bir abonelik olup olmadığını denetler ve önceden var olan aboneliğe bir tanıtıcı döndürürse; ancak var olan bir abonelik yoksa, MQSD' de sağlanan tüm alanlar kullanılarak yeni bir abonelik yaratılır.

SOCRT , benzer bir etki için SOALT ile de birleştirilebilir (bu konuda daha sonra SOALT ile ilgili ayrıntılara bakın).

SIRALAR

SDSNtarafından belirtilenlerle eşleşen önceden var olan bir aboneliğe bir tanıtıcı döndürür. Eşleşen abonelik özniteliklerinde değişiklik yapılmadı ve bunlar MQSD yapısındaki çıkışta döndürülür. MQSD içeriğinin çoğu kullanılmaz: Kullanılan alanlar şunlardır: *SDSID, SDVER, SDOPT, SDAID , SDASIVe SDSN.*

Tam abonelik adıyla eşleşen bir abonelik yoksa, çağrı RC2428 neden koduyla başarısız olur. Bu hata, SOCRT seçeneği SORESile birleştirilerek önlenir. SOCRTile ilgili ayrıntılar için bkz. [SOCRT](#).

Aboneliğin kullanıcı kimliği, aboneliği oluşturan kullanıcı kimliğidir ya da daha sonra farklı bir kullanıcı kimliği tarafından değiştirildiyse, bu, en son, başarılı değişikliğin kullanıcı kimliğidir. Bir *SDAID* kullanılırsa ve o kullanıcı için diğer kullanıcı kimliklerinin kullanılmasına izin verilirse, *SDAID* aboneliğin yapıldığı kullanıcı kimliği yerine aboneliği oluşturan kullanıcı kimliği olarak kaydedilir.

Aboneliği oluşturan kullanıcı kimliği, bu alan kullanılıyorsa *SDAU* olarak kaydedilir ve o kullanıcı için diğer kullanıcı kimliklerinin kullanılmasına izin verilir.

SOAUID seçeneği olmadan oluşturulan eşleşen bir abonelik varsa ve aboneliğin kullanıcı kimliği, abonelik için tanıtıcı isteyen uygulamadan farklıysa, çağrı RC2434 neden koduyla başarısız olur.

Eşleşen bir abonelik varsa ve şu anda başka bir uygulama tarafından kullanılıyorsa, çağrı RC2429 neden koduyla başarısız olur. Şu anda aynı bağlantı tarafından kullanılıyorsa, çağrı başarısız olmaz ve abonelik için bir tanıtıcı döndürülür.

SubName içinde adı belirtilen abonelik, bir uygulamadan sürdürmek ya da değiştirmek için geçerli bir abonelik değilse, çağrı RC2523 ile başarısız olur.

SORES , SOALT tarafından örtük olarak belirtilir ve bu seçenekle birleştirilmesi gerekmez; ancak, bu iki seçeneğin birleştirilmesi bir hata değildir.

SOALT

SDSNiçinde belirtilenlerle eşleşen tam abonelik adıyla önceden var olan bir aboneliğe bir tanıtıcı döndürür. Aboneliğin MQSD ' de belirtilenlerden farklı olan öznitelikleri, o öznitelik

için deęişikliğe izin verilmedięi sürece abonelikte deęiştirilir. Ayrıntılar her özniteliğın tanımında belirtilir ve ařağıdaki tabloda özetlenir. Deęiştirilemeyen bir öznitelięi deęiştirmeye çalıřsanız, çağrı ařağıdaki çizelgede gösterilen neden koduyla başarısız olur.

Tam abonelik adıyla eřleşen bir abonelik yoksa, çağrı RC2428 neden koduyla başarısız olur. Bu hata, SOCRT seçeneęi SOALTile birleřtirilerek önlenir.

SOVRT 'nin SOALT ile birleřtirilmesi, önce belirtilen tam abonelik adı için var olan bir abonelik olup olmadıęını ve önceden ayrıntılı olarak yapılan deęişikliklerle önceden var olan aboneliğın bir tanıtıcısı olup olmadıęını denetler; ancak var olan abonelik yoksa, MQSD' de saęlanan tüm alanlar kullanılarak yeni bir abonelik yaratılır.

Aboneliğın kullanıcı kimlięi, abonelięi oluřturan kullanıcı kimlięidir ya da daha sonra farklı bir kullanıcı kimlięi tarafından deęiştirildiyse, en son başarılı deęişikliğin kullanıcı kimlięidir. SDAU kullanılırsa (ve bu kullanıcı için dięer kullanıcı kimliklerinin kullanılmasına izin verilirse), dięer kullanıcı kimlięi, aboneliğın yapıldıęı kullanıcı kimlięi yerine, abonelięi yaratan kullanıcı kimlięi olarak kaydedilir.

SOAUID seçeneęi olmadan oluřturulan eřleşen bir abonelik varsa ve aboneliğın kullanıcı kimlięi, abonelik için tanıtıcı isteyen uygulamadan farklıysa, çağrı başarısız olur. Neden kodu RC2434 .

Eřleşen bir abonelik varsa ve řu anda başka bir uygulama tarafından kullanılıyorsa, arama RC2429 ile başarısız olur. řu anda aynı baęlantı tarafından kullanılıyorsa çağrı başarısız olmaz ve abonelik için bir tanıtıcı döndürülür.

SubName içinde adı belirtilen abonelik, bir uygulamadan sürdürmek ya da deęiştirmek için geçerli bir abonelik deęilse, çağrı RC2523 ile başarısız olur.

Ařağıdaki çizelgelerde, SOALTile deęiştirilebilecek abonelik öznitelikleri gösterilir.

<i>Çizelge 728. MQSD ve MQSUB içindeki deęiştirilebilecek öznitelikler</i>			
Veri tipi tanımlayıcısı ya da iřlev çağırısı	Alan adı	Bu öznitelik SOALT kullanılarak deęiştirilebilir mi?	Neden Kodu
MQSD	Dayanıklılık seçenekleri	Hayır	RC2509
MQSD	Hedef Seçenekleri	Evet	Yok
MQSD	Kayıt seçenekleri	Evet (bkz. not 1)	RC2515 SOGRP ' yi deęiştirmeye çalıřsanız
MQSD	Yayınlama seçenekleri	Evet (bkz. not 2)	Yok
MQSD	Genel arama karakteri seçenekleri	Hayır	RC2510
MQSD	Dięer seçenekler	Hayır (bkz. not 3)	Yok
MQSD	ObjectName	Hayır	RC2510
MQSD	SDAU	Hayır (bkz. not 4)	Yok
MQSD	SDASI	Hayır (bkz. not 4)	Yok
MQSD	SDEXP	Evet	Yok
MQSD	SDOS	Hayır	RC2510
MQSD	SDSN	Hayır (bkz. not 5)	Yok
MQSD	SDSUD	Evet	Yok
MQSD	SDCID	Evet (bkz. not 6)	RC2515 gruplanmış abonelikteyken
MQSD	SDPRI	Evet	Yok

Çizelge 728. MÇSD ve MÇSUB içindeki değiştirilebilecek öznitelikler (devamı var)			
Veri tipi tanımlayıcısı ya da işlev çağrısı	Alan adı	Bu öznitelik SOALT kullanılarak değiştirilebilir mi?	Neden Kodu
MÇSD	SDACC	Evet	Yok
MÇSD	SDAID	Evet	Yok
MÇSD	SDSL	Hayır	RC2512
MÇSUB	Hobj.	Evet (bkz. not 6)	RC2515 gruplanmış abonelikteyken

Notlar:

1. SOGRP değiştirilemiyor.
2. SONEWP , aboneliğin bir parçası olmadığı için değiştirilemiyor
3. Bu seçenekler aboneliğin bir parçası değildir
4. Bu öznitelik, aboneliğin bir parçası değil
5. Bu öznitelik, değiştirilmekte olan aboneliğin tanıtıcısıdır
6. Gruplanmış alt (SOGRP) ögesinin bir parçası dışında değiştirilebilen

Dayanıklılık seçenekleri: Aşağıdaki seçenekler, aboneliğin ne kadar dayanıklı olduğunu denetler. Bu seçeneklerden yalnızca birini belirleyebilirsiniz. Var olan bir aboneliği SOALT seçeneğini kullanarak değiştiriyorsanız, aboneliğin dayanıklılığını değiştiremezsiniz. SORESkullanılarak bir MÇSUB çağrısından döndüğünüzde, uygun dayanıklılık seçeneği ayarlanır.

SODUR.

Bu konuya ilişkin aboneliğin, CORMSB seçeneğiyle MÇCLOSE kullanılarak belirtik olarak kaldırılincaya kadar kalmasını isteyin. Bu abonelik belirtik olarak kaldırılmazsa, bu uygulama kuyruk yöneticisine bağlandıktan sonra da kalır.

Sürekli aboneliğe izin vermeyen bir konuya sürekli abonelik istenirse, çağrı RC2436 ile başarısız olur.

SONDUR.

Önceden belirtik olarak kaldırılmadıysa, kuyruk yöneticisine yönelik uygulama bağlantısı kapatıldığında bu konuya ilişkin aboneliğin kaldırılmasını isteyin. SONDUR , SODUR seçeneğinin tersidir ve program belgelerine yardımcı olmak üzere tanımlanır. İkisi de belirtilmezse, varsayılan değer budur.

Hedef seçenekleri: Aşağıdaki seçenekler, abone olunan bir konuya ilişkin yayınların gönderileceği hedefi denetler. SOALT seçeneğini kullanarak var olan bir abonelik değiştiriliyorsa, abonelik için yayınlar için kullanılan hedef değiştirilebilir. SORESkullanılarak bir MÇSUB çağrısından dönüş yapılırsa, bu seçenek uygunsuz ayarlanır.

SOMAN.

Yayınların gönderildiği hedefin kuyruk yöneticisi tarafından yönetilmesini isteyin.

HOBJ içinde döndürülen nesne tanıtıcısı, kuyruk yöneticisi tarafından yönetilen bir kuyruğu temsil eder ve sonraki MÇGET, MÇCB, MÇINQ ya da MÇCLOSE çağrılarıyla birlikte kullanılır.

Önceki bir MÇSUB çağrısından döndürülen bir nesne tanıtıcısı, SOMAN belirtilmediğinde **Hobj** değiştirilmesinde sağlanamaz.

Kayıt seçenekleri: Aşağıdaki seçenekler, bu abonelik için kuyruk yöneticisine yapılan kayıt ayrıntılarını denetler. SOALT seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, bu kayıt seçenekleri değiştirilebilir. SORES kullanılarak bir MÇSUB çağrısından döndüğünüzde, uygun kayıt seçenekleri ayarlanır.

SOGRP.

Bu abonelik, aynı kuyruk kullanılarak aynı *SDSL* abonelikleriyle gruplanır ve aynı ilinti tanıtıcısı belirtilerek, birden çok yayın iletilsinin abonelik grubuna sağlanmasına neden olacak yayınların, üst üste gelen konu dizeleri kümesi nedeniyle kuyruğa tek bir iletinin gönderilmesine neden olur. Bu seçenek kullanılmazsa, eşleşen her benzersiz abonelik (*SDSN* ile tanıtılır), yayının bir kopyasıyla birlikte sağlanır. Bu, yayının birden çok kopyasının, birden çok abonelik tarafından paylaşılan kuyruğa yerleştirileceği anlamına gelebilir.

Yalnızca gruptaki en önemli abonelik, yayının bir kopyasıyla birlikte sağlanır. En önemli abonelik, genel arama karakterinin bulunduğu noktaya kadar Tam konu adını temel alır. Grup içinde genel arama karakteri şemalarının bir karışımı kullanılırsa, yalnızca genel arama karakterinin konumu önemlidir. Aynı kuyruğu paylaşan abonelikler grubunda farklı genel arama karakteri şemalarını birleştirmemeniz önerilir.

Yeni bir gruplanmış abonelik oluştururken yine de benzersiz bir *SDSN* olmalıdır, ancak gruptaki var olan bir aboneliğin tam konu adıyla eşleşirse, arama RC2514 ile başarısız olur.

Gruptaki en önemli abonelik *SONLC* belirtiyorsa ve bu aynı uygulamadan bir yayınsa, kuyruğa herhangi bir yayın teslim edilmez.

Bu seçenekle yapılan bir abonelik değiştirilirken, *MQSUB* çağrısında *Hobj* (kuyruk ve kuyruk yöneticisi adını gösterir) gruplamayı belirten alanlar ve *SDCID* değiştirilemez. Bunları değiştirme girişimi, RC2515 ile çağrı başarısız olmasına neden olur.

Bu seçenek, *CINONE* olarak ayarlanmamış bir *SDCID* ile *SOSCID* ile birleştirilmeli ve *SOMAN* ile birleştirilmemelidir.

SOAUID

SOAUID belirtildiğinde, abonenin kimliği tek bir kullanıcı kimliğiyle sınırlı değildir. Bu, herhangi bir kullanıcının uygun yetkiye sahip olduğunda aboneliği değiştirmesini ya da sürdürmesini sağlar. Aynı anda yalnızca tek bir kullanıcı aboneliğe sahip olabilir. Başka bir uygulama tarafından kullanılmakta olan bir aboneliği kullanmaya devam etme girişimi, RC2429 ile yapılan aramanın başarısız olmasına neden olur.

Bu seçeneği var olan bir aboneliğe eklemek için, *SOALT* kullanan *MQSUB* çağrısı, özgün aboneliğin kendisiyle aynı kullanıcı kimliğinden gelmelidir.

Bir *MQSUB* çağrısı *SOAUID* ile var olan bir aboneliğe başvurursa ve kullanıcı kimliği özgün abonelikten farklıysa, çağrı yalnızca yeni kullanıcı kimliğinin konuya abone olma yetkisi varsa başarılı olur. İşlem hatasız olarak tamamlandığında, bu abonenin gelecekteki yayınları, yayın iletilsinde yeni kullanıcı kimliği ayarlanmış olarak abonenin kuyruğuna yerleştirilir.

Hem *SOAUID* , hem de *SOFUID* belirtmeyin. Hiçbiri belirtilmezse, varsayılan değer *SOFUID* ' dir.

SOFUID

SOFUID belirtildiğinde, abonelik yalnızca aboneliği değiştiren son kullanıcı kimliği tarafından değiştirilebilir ya da sürdürülebilir. Abonelik değiştirilmediyse, aboneliği yaratan kullanıcı kimliği olur.

Bir *MQSUB* komutu *SOAUID* ile var olan bir aboneliğe gönderme yaparsa ve *SOFUID* ' yi kullanmak için *SOALT* kullanarak aboneliği değiştirirse, aboneliğin kullanıcı kimliği artık bu yeni kullanıcı kimliğinde düzeltilmiştir. Arama, yalnızca yeni kullanıcı kimliğinin konuya abone olma yetkisi varsa başarılı olur.

Aboneliğin sahibi olarak kaydedilenden başka bir kullanıcı kimliği *SOFUID* aboneliğini sürdürmeye ya da değiştirmeye çalışırsa, arama RC2434 ile başarısız olur. Bir aboneliğin sahip olan kullanıcı kimliği, **DISPLAY SBSTATUS** komutu kullanılarak görüntülenebilir.

Hem *SOAUID* , hem de *SOFUID* belirtmeyin. Hiçbiri belirtilmezse, varsayılan değer *SOFUID* ' dir.

Yayın seçenekleri: Aşağıdaki seçenekler, yayınların bu aboneye gönderilme şeklini denetler. Var olan bir abonelik *SOALT* seçeneği kullanılarak değiştiriliyorsa, bu yayın seçenekleri değiştirilebilir.

SONOLC

Aracıya, uygulamanın kendi yayınlarından hiçbirini görmek istemediğini bildirir. Bağlantı tanıtıcıları aynıysa, yayınların aynı uygulamadan kaynaklandığı kabul edilir. SORES kullanılarak bir MQSUB çağrısından dönüşte, bu seçenek uygunsuz ayarlanır.

SONEWP

Bu abonelik oluşturulduğunda, şu anda alıkonan hiçbir yayın gönderilmez, yalnızca yeni yayınlar gönderilir. Bu seçenek yalnızca SOCRE belirtildiğinde geçerlidir. Abonelikte yapılan sonraki değişiklikler, yayınların akışını değiştirmez ve bu nedenle bir konuda tutulan yayınlar, aboneye yeni yayınlar olarak gönderilmiştir.

Bu seçenek SOCRE olmadan belirtilirse, çağrı RC2046 ile başarısız olur. SORES kullanılarak bir MQSUB çağrısından dönüş yapılsa, abonelik bu seçenek kullanılarak yaratılmış olsa da bu seçenek ayarlanmaz.

Bu seçenek kullanılmazsa, önceden alıkonan iletiler sağlanan hedef kuyruğa gönderilir. RC2525 ya da RC2526 hatası nedeniyle bu işlem başarısız olursa, abonelik oluşturma işlemi başarısız olur.

Bu seçenek SOPUBR ile birlikte geçerli değildir.

SOPUBR

Bu seçeneğin ayarlanması, abonenin özellikle gerektiğinde bilgi isteyeceğini gösterir. Kuyruk yöneticisi aboneye istenmeyen iletiler göndermez. Önceki bir MQSUB çağrısından Hsub tanıtıcısı kullanılarak bir MQSUBRQ çağrısı her yapıldığında, alıkonan yayın (ya da bir joker karakter belirtilirse birden çok yayın) aboneye gönderilir. Bu seçenek kullanılarak MQSUB çağrısının sonucu olarak hiçbir yayın gönderilmez. SORES kullanılarak bir MQSUB çağrısından dönüşte, bu seçenek uygunsuz ayarlanır.

Bu seçenek, SONEWP ile birlikte geçerli değildir.

Genel arama karakteri seçenekleri: Aşağıdaki seçenekler, MQSD ' nin SDOS alanında sağlanan dizgide genel arama karakterlerinin nasıl yorumlanacağını denetler. Bu seçeneklerden yalnızca birini belirleyebilirsiniz. Var olan bir abonelik SOALT seçeneği kullanılarak değiştiriliyorsa, bu genel arama karakteri seçenekleri değiştirilemez. SORES kullanılarak bir MQSUB çağrısından döndüğünüzde, uygun genel arama karakteri seçeneği ayarlanır.

SOWCHR

Genel arama karakterleri yalnızca konu dizisi içindeki karakterlerde çalışır. SOWCHR alanı, eğik çizgiyi (/) özel bir anlamı olmayan başka bir karakter olarak işler.

SOWCHR tarafından tanımlanan davranış aşağıdaki çizelgede gösterilir:

Çizelge 729. Joker karakterlerin yorumlanması	
Özel Karakter	Davranış
*	Joker karakter, sıfır ya da daha fazla karakter
?	Genel arama karakteri, bir karakter
%	'*', '?' ya da '%' karakterlerinin bir dizgide kullanılmasına ve özel karakter olarak yorumlanmamasına izin vermek için çıkış karakteri kullanın; örneğin, '% *', '%?' ya da '%% '.

Örneğin, şu konuda yayınlama:

```
/level0/level1/level2/level3/level4
```

aboneleri aşağıdaki konuları kullanarak eşleştirir:

```
*  
/*  
/ level0/level1/level2/level3/*
```

```
/ level0/level1/*/level3/level4  
/ level0/level1/le?e12/level3/level4
```

Not: Bu joker karakter kullanımı, Yayınlama/Abone Olma için MQRFH1 biçimlendirilmiş iletileri kullanırken, IBM MQ V6 ve WebSphere MB V6 ' da sağlanan anlamı tam olarak sağlar. Bunun yeni yazılan uygulamalar için kullanılmaması ve yalnızca daha önce o sürüm için çalışan uygulamalar için kullanılması ve SOWTOP içinde açıklandığı gibi varsayılan genel arama karakteri davranışını kullanacak şekilde değiştirilmemiş olması önerilir.

SOWTOP

Joker karakterler yalnızca konu dizesi içindeki konu öğelerinde çalışır. Hiçbiri seçilmezse, bu varsayılan davranıştır.

SOWTOP için gereken davranış aşağıdaki çizelgede gösterilmiştir:

Özel Karakter	Davranış
/	Konu düzeyi ayırıcısı
#	Genel arama karakteri: birden çok konu düzeyi
+	Genel arama karakteri: tek konu düzeyi

Not:

'+' ve '#', bir konu düzeyinde diğer karakterlerle (kendileri dahil) karıştırıldıysa genel arama karakteri olarak kabul edilmez. Aşağıdaki dizgide, '#' ve '+' karakterleri olağan karakterler olarak işlenir.

```
level0/level1/#+/level3/level#
```

Örneğin, şu konuda yayınlama:

```
/level0/level1/level2/level3/level4
```

aboneleri aşağıdaki konuları kullanarak eşleştirir:

```
#  
/#  
/ level0/level1/level2/level3/#  
/ level0/level1+/level3/level4
```

Not: Bu joker karakter kullanımı, Yayınlama/Abone Olma için MQRFH2 biçimlendirilmiş iletileri kullanırken WebSphere Message Broker 6 içinde sağlanan anlamı sağlar.

Diğer seçenekler: Aşağıdaki seçenekler, abonelik yerine API çağrısını nasıl yayınlanacağına denetler. SORES kullanılarak bir MQSUB çağrısından dönüşte bu seçenekler değişmez.

SOALTU

SDAU alanı, bu MQSUB çağrısının geçerliliğini denetlemek için kullanılacak bir kullanıcı kimliği içerir. Arama, yalnızca bu SDAU ' nun belirtilen erişim seçenekleriyle nesneyi açma yetkisi varsa, uygulamanın altında çalıştığı kullanıcı kimliğinin bunu yapma yetkisi olup olmadığına bakılmaksızın başarılı olabilir.

SOSCID

Abonelik, *SDCID* alanında sağlanan ilinti tanıtıcısını kullanmaktır. Bu seçenek belirtilmezse, abonelik sırasında kuyruk yöneticisi tarafından otomatik olarak bir ilinti tanıtıcısı yaratılır ve *SDCID* alanında uygulamaya döndürülür. Daha fazla bilgi için bkz. [SDCID \(24 baytlık bit dizgisi\) SDCID](#) .

SOSETI

Abonelik, *SDACC* ve *SDAID* alanlarında sağlanan muhasebe simgesini ve uygulama kimlik verilerini kullanmaktır.

Bu seçenek belirtilirse, hedef kuyruğa 00SETI ile MQOPEN çağrısı kullanılarak erişildiği gibi, SOMAN seçeneğinin de kullanıldığı durumlar dışında, aynı yetki denetimi gerçekleştirilir; bu durumda hedef kuyrukta yetki denetimi yoktur.

Bu seçenek belirtilmezse, bu aboneye gönderilen yayınlarla ilişkili varsayılan bağlam bilgileri aşağıdaki gibi olur:

<i>Çizelge 731. Bu aboneye gönderilen yayınlar için varsayılan bağlam bilgileri</i>	
MQMD 'deki Alan	Kullanılan değer
<i>MDUID</i>	Abonelik yapıldığında abonelik ile ilişkilendirilen kullanıcı kimliği.
<i>MDACC</i>	Olanaklıysa ortamdan saptanır; yoksa ACNONE olarak ayarlanır.
<i>MDAID</i>	Boşluklara ayarla

Bu seçenek yalnızca SOCRE ve SOALT ile geçerlidir. SORES ile kullanılırsa, *SDACC* ve *SDAID* alanları yoksayılır, bu nedenle bu seçeneğin etkisi olmaz.

Abonelik, daha önce aboneliğin sağladığı kimlik bağlamı bilgilerinin bulunduğu bu seçenek kullanılmadan değiştirilirse, değiştirilen abonelik için varsayılan bağlam bilgileri oluşturulur.

Farklı kullanıcı kimliklerinin bunu SOAUID seçeneğiyle kullanmasına izin veren bir abonelik farklı bir kullanıcı kimliği tarafından sürdürüldüğünde, artık aboneliğe sahip olan yeni kullanıcı kimliği için varsayılan kimlik bağlamı oluşturulur ve yeni kimlik bağlamını içeren sonraki yayınlar teslim edilir.

SOFIQ

Kuyruk yöneticisi susturma durumundaysa MQSUB çağrısı başarısız olur. z/OS sistemlerinde, bir CICS ya da IMS uygulaması için bu seçenek, bağlantı susturma durumundaysa MQSUB çağrısına da başarısız olmasına neden olur.

SDAU (12 baytlık karakter dizilimi)

SOALTU belirtirseniz, bu alan, uygulamanın çalışmakta olduğu kullanıcı kimliği yerine, aboneliğe ve hedef kuyruğa (MQSUB çağrısının **Hobj** değiştirgesinde belirtilir) ilişkin yetkiyi denetlemek için kullanılan diğer bir kullanıcı kimliği içerir.

Başarılı olursa, bu alanda belirtilen kullanıcı kimliği, uygulamanın çalışmakta olduğu kullanıcı kimliği yerine, abonelik sahibi kullanıcı kimliği olarak kaydedilir.

SOALTU belirtilirse ve bu alan ilk boş karaktere ya da alanın sonuna kadar tamamen boşsa, abonelik ancak belirtilen seçeneklerle bu konuya ya da çıkış için hedef kuyruğa hiçbir kullanıcı yetkisinin abone olması gerekmediğinde başarılı olabilir.

SOALTU belirtilmezse, bu alan yoksayılır.

SORESkullanılarak MQSUB çağrısından dönüş yapılıyorsa, bu alan değişmez.

Bu bir giriş alanıdır. Bu alanın uzunluğu LNUID tarafından verilir. Bu alanın ilk değeri 12 boş karakterdir.

SDPRI (10 basamaklı işaretli tamsayı)

Bu değer, bu abonelikte eşleşen tüm yayın iletilerinin ileti tanımlayıcısının (MQMD) MQPRI alanında bulunan değerdir. MQMD 'deki MQPRI alanıyla ilgili daha fazla bilgi için bkz. [MDPRI](#).

Değer sıfırdan büyük ya da sıfıra eşit olmalıdır; sıfır en düşük önceliklidir. Aşağıdaki özel değerler de kullanılabilir:

PRQDEF

MQSUB çağrısındaki Hobj alanında bir abonelik kuyruğu sağlandığında ve yönetilen bir tanıtıcı olmadığında, iletinin önceliği bu kuyruğun **DefPriority** özniteliğinden alınır. Bu şekilde tanımlanan kuyruk bir küme kuyruğuysa ya da kuyruk adı çözme yolunda birden çok tanımlama varsa, MDPRI için açıklandığı gibi, yayınlama iletisi kuyruğa konduğunda öncelik belirlenir.

MQSUB çağrısı yönetilen bir tanıtıcı kullanıyorsa, iletinin önceliği, abone olunan konuyla ilişkili model kuyruğunun **DefPriority** özniteliğinden alınır.

PRPUB

İletinin önceliği, özgün yayının önceliğidir. Bu, alanın ilk değeridir.

SOALT seçeneği kullanılarak var olan bir abonelik değiştiriliyorsa, ilerideki yayın iletilerinin *MQPRI* değeri değiştirilebilir.

SORES kullanılarak bir MQSUB çağrısından döndüğünüzde, bu alan abonelik için kullanılmakta olan yürürlükteki önceliğe ayarlanır.

SDRO (MQCHARV)

SDRO, kuyruk yöneticisi *SDON* içinde sağlanan adı çözdükten sonra uzun nesne adıdır.

Uzun nesne adı *SDOS* içinde sağlanırsa ve *SDON* içinde hiçbir şey sağlanmazsa, bu alanda döndürülen değer, *SDOS* içinde sağlananla aynıdır.

Bu alan atlanırsa (*SDRO.VSBufSize* sıfır), *SDRO* döndürülmez, ancak uzunluk *SDRO.VSLength*. Uzunluk tam *SDRO* değerinden kısaysa, kesilir ve sağlanan uzunluğa sığabilecek en sağdaki karakterlerin sayısını döndürür.

SDRO yanlış belirtilirse, MQCHARV yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı RC2520 neden koduyla başarısız olur.

SDSID (4 baytlık karakter dizgisi)

Bu yapı tanıtıcısıdır; değer şu olmalıdır:

SDSIDV

Abonelik Tanımlayıcı yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri *SDSIDV* ' dir.

SDSL (10 basamaklı işaretli tamsayı)

Bu, abonelik ile ilişkili düzeydir. Yayınlar, yalnızca yayın sırasında kullanılan *PubLevel* düzeyinden küçük ya da bu değere eşit en yüksek *SDSL* değerine sahip abonelikler kümesinde yer alıyorsa bu aboneliğe teslim edilir.

Değer, sıfır ile 9 aralığında olmalıdır. Sıfır en düşük düzeydir.

Bu alanın ilk değeri 1 'dir.

Var olan bir abonelik *SOALT* seçeneği kullanılarak değiştiriliyorsa, *SDSL* değiştirilemez.

SDSN (MQCHARV)

SDSN abonelik adını belirtir.

Bu alan yalnızca *SDOPT SODUR* seçeneğini belirtiyorsa, ancak sağlandıysa *SONDUR* kuyruk yöneticisi tarafından da kullanılır. Belirtilirse, *SDSN* kuyruk yöneticisi içinde benzersiz olmalıdır; bu alan abonelikleri tanımlamak için kullanılır.

SDSN uzunluk üst sınırı 10240 'tır.

Bu alan iki amaca hizmet eder. *SODUR* aboneliği için, abonelik tanıtıcısını kapattıysanız (*COKPSB* seçeneğini kullanarak) ya da kuyruk yöneticisiyle bağlantınız kesildiyse, bu abonelik oluşturulduktan sonra sürdürülecek bir aboneliği tanımlama aracıdır. Yaratıldıktan sonra kaldırılacak aboneliğin tanımlanması, *SORES* seçeneğiyle *MQSUB* çağrısı kullanılarak gerçekleştirilir. *SDSN* alanı, *DISPLAY SBSTATUS* içindeki *SDSN* alanındaki aboneliklerin yönetim görünümünde de görüntülenir.

SDSN yanlış belirtilirse, *MQCHARV* yapısının nasıl kullanılacağına ya da uzunluk üst sınırını aştığına ya da gerektiğinde atlandığına (*SDSN*) göre. *VCHRL* sıfır) ya da uzunluk üst sınırını aşarsa, çağrı RC2440 neden koduyla başarısız olur.

Bu bir giriş alanıdır. Bu yapıdaki alanların başlangıç değerleri, *MQCHARV* yapısındaki değerlerle aynıdır.

Var olan bir abonelik *SOALT* seçeneği kullanılarak değiştiriliyorsa, abonelik adı, aboneliği tanımlamak için kullanılan alan olduğundan değiştirilemez. *SORES* seçeneğiyle bir *MQSUB* çağrısının çıkışında değiştirilmez.

SDSS (MQCHARV)

SDSS , bir konudan gelen iletilere abone olurken kullanılan seçim ölçütlerini sağlayan dizedir.

Bu değişken uzunluklu alan, *SORES* seçeneği kullanılarak bir *MQSUB* çağrısının çıkışında, bir arabellek sağlandıysa ve *VSBufSize* içinde de pozitif bir arabellek uzunluğu varsa döndürülür. Çağrıda arabellek sağlanmazsa, *MQCHARV* ' nin *VSL* kalınlık alanında yalnızca seçim dizgisinin uzunluğu döndürülür. Sağlanan arabellek alanı döndürmek için gereken alandan küçükse, sağlanan arabellekte yalnızca *VSBufSize* byte döndürülür.

SDSS yanlış belirtilirse, *MQCHARV* yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı RC2519 neden koduyla başarısız olur.

SDSUD (MQCHARV)

Bu alandaki abonelikte sağlanan veriler, bu aboneliğe gönderilen her yayının *mq.SubUserData* ileti özelliği olarak içerilir.

SDSUD uzunluk üst sınırı 10240 'tır.

SDSUD yanlış belirtilirse, *MQCHARV* yapısının nasıl kullanılacağına ilişkin açıklamaya göre ya da uzunluk üst sınırını aşarsa, çağrı başarısız olur; neden kodu RC2431.

Bu bir giriş alanıdır. Bu yapıdaki alanların başlangıç değerleri, *MQCHARV* yapısındaki değerlerle aynıdır.

Var olan bir abonelik *SOALT* seçeneği kullanılarak değiştiriliyorsa, abonelik kullanıcı verileri değiştirilebilir.

Bu değişken uzunluklu alan, bir arabellek sağlandıysa ve *VSBufLen* içinde pozitif bir arabellek uzunluğu varsa, *SORES* seçeneği kullanılarak *MQSUB* çağrısının çıkışında döndürülür. Çağrıda arabellek sağlanmazsa, *MQCHARV* ' nin *VCHRL* alanında yalnızca abonelik kullanıcı verilerinin uzunluğu döndürülür. Sağlanan arabellek, alanı döndürmek için gereken alandan küçükse, sağlanan arabellekte yalnızca *VSBufLen* byte döndürülür.

SDVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

SDVER1

Version-1 Abonelik Tanımlayıcı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

SDVERC

Abonelik Tanımlayıcı yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Alanın ilk değeri *SDVER1* ' dir.

Başlangıçtaki değerler

<i>Çizelge 732. MQSD 'deki Alanlar</i>		
Alan adı	Değişmezin adı	Değişmezin değeri
<i>SDSID</i>	<i>SDSIDV</i>	'SD--'
<i>SDVER</i>	<i>SDVER1</i>	1

Çizelge 732. MQSD ' deki Alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
SDOPT	SONDUR.	0
SDON	Yok	Boşluklar
SDAU	Yok	Boşluklar
SDASI	SINONE.	Boş Değerler
SDEXP	EIULIM	-1
SDOS	MQCHARV için tanımlanan adlar ve değerler	
SDSN	MQCHARV için tanımlanan adlar ve değerler	
SDSUD	MQCHARV için tanımlanan adlar ve değerler	
SDCID	TAMAMDİR.	Boş Değerler
SDPRI	PRQDEF	-3
SDACC	YOK	Boş Değerler
SDAID	Yok	Boşluklar
SDSL	Yok	1
SDRO	MQCHARV ' de tanımlandığı şekilde adlar ve değerler	
Not:		
1. - simgesi, tek bir boş karakteri temsil eder.		

RPG bildirim

```

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 SDRDP       341    356*
D SDRDO       357    360I 0
D SDRDS       361    364I 0
D SDRDL       365    368I 0
D SDRDC       369    372I 0

```

IBM i

IBM i üzerinde MQSMPO (İleti özelliği seçeneklerini ayarla)

MQSMPO yapısı, uygulamaların, ileti özelliklerinin nasıl ayarlanacağını denetleyen seçenekleri belirtmesine olanak sağlar.

Genel Bakış

Amaç: Yapı, **MQSETMP** çağrısında bir giriş parametresidir.

Karakter kümesi ve kodlama: **MQSMPO** içindeki veriler, uygulamanın karakter kümesinde ve kodlamasında (ENNAT) olmalıdır.

- [“Alanlar” sayfa 1207](#)
- [“Başlangıçtaki değerler” sayfa 1208](#)
- [“RPG bildirim” sayfa 1209](#)

Alanlar

MQSMPO yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

SPOPT (10 basamaklı işaretli tamsayı)

Konum seçenekleri: Aşağıdaki seçenekler, özellik imleciyle karşılaştırıldığında özelliğin görel konumuyla ilgilidir:

SPSETF

Belirtilen adla eşleşen ilk özelliğin değerini ayarlar ya da yoksa, eşleşen bir sıradüzene sahip diğer tüm özelliklerden sonra yeni bir özellik ekler.

SPSETC

Özellik imlecinin gösterdiği özelliğin değerini belirler. Özellik imlecinin gösterdiği özellik, IPINQF ya da IPINQN seçeneği kullanılarak en son sorulması gereken özelliktir.

İleti tanıtıcısı yeniden kullanıldığında ya da MQGET çağrısındaki MQGMO yapısının *HMSG* alanında ya da MQPUT çağrısındaki MQPMO yapısında ileti tanıtıcısı belirtildiğinde özellik geçici çizelgesi ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında ya da özellik imleci tarafından gösterilen özellik silindiğinde bu seçenek kullanılırsa, çağrı tamamlanma kodu CCFAIL ve neden kodu RC2471 ile başarısız olur.

SPSETA

Özellik imlecinin gösterdiği özellikten sonra yeni bir özellik belirler. Özellik imlecinin gösterdiği özellik, IPINQF ya da IPINQO seçeneği kullanılarak en son sorulması gereken özelliktir.

İleti tanıtıcısı yeniden kullanıldığında ya da MQGET çağrısındaki MQGMO yapısının HMSG alanında ya da MQPUT çağrısındaki MQPMO yapısında ileti tanıtıcısı belirtildiğinde özellik geçici çizelgesi ilk durumuna getirilir.

Özellik imleci henüz oluşturulmadığında ya da özellik imleci tarafından gösterilen özellik silindiğinde bu seçenek kullanılırsa, çağrı tamamlanma kodu CCFAIL ve neden kodu RC2471 ile başarısız olur.

Açıklanan seçeneklerin hiçbirine gerek duymazsanız, aşağıdaki seçeneği kullanın:

SPYOK

Seçenek belirtilmedi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri SPSETF 'dir.

SPSID (10 basamaklı işaretli tamsayı)

Bu yapı tanıtıcısıdır; değer şu olmalıdır:

SPSIDV

İleti özellik seçeneklerini ayarlama yapısına ilişkin tanıtıcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **SPSIDV**.

SPVAKCSI (10 basamaklı işaretli tamsayı)

Değer bir karakter dizgisiyse ayarlanacak özellik değerinin karakter kümesi.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **CSAPL**.

SPVALENC (10 basamaklı işaretli tamsayı)

Değer sayıysa ayarlanacak özellik değerinin kodlaması.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **ENNAT**.

SPVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

SPVER1

Version-1 ileti özelliği seçenekleri yapısını ayarlayın.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

SPVERC (GIZLI)

İleti ayarlama özelliği seçenekleri yapısının yürürlükteki sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri şudur: **SPVER1**.

Başlangıçtaki değerler

Çizelge 733. MQSMPO içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
SPSID	SPSIDV	' SMPO '
SPVER	SPVER1	1
SPOPT	SPYOK	0
SPVALENC	GIRIŞ	Ortama bağlıdır

Çizelge 733. MQSMPO içindeki alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
SPVALCSI	CSAPL	-3

RPG bildirim

```
D* MQSMPO Structure
D*
D*
D* Structure identifier
D SPSID          1      4      INZ('SMPO')
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)
```

IBM i

IBM i üzerinde MQSRO (Abonelik İsteği Seçenekleri)

MQSRO yapısı, uygulamanın abonelik isteğinin nasıl yapılacağını denetleyen seçenekleri belirtmesini sağlar.

Genel Bakış

Amaç: Yapı, MQSUBRQ çağrısında bir giriş/çıkış değiştirgesidir.

Sürüm: MQSRO ' nun yürürlükteki sürümü SRVER1.

- [“Alanlar” sayfa 1209](#)
- [“Başlangıçtaki değerler” sayfa 1210](#)
- [“RPG bildirim” sayfa 1210](#)

Alanlar

MQSRO yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

SRNMP (10 basamaklı işaretli tamsayı)

Bu, bu çağrı sonucunda abonelik kuyruğuna gönderilen yayınların sayısını belirtmek için uygulamaya döndürülen bir çıkış alanıdır. Bu çağrı sonucunda bu sayıda yayın gönderilmesine rağmen, özellikle kalıcı olmayan iletiler olduğunda, uygulamanın bu kadar çok iletiyi alabileceğine dair bir garanti yoktur.

Abone olunan konu bir genel arama karakteri içeriyorsa, birden çok yayın olabilir. *HSUB* tarafından temsil edilen abonelik oluşturulduğunda konu dizisinde joker karakter yoksa, bu çağrı sonucunda en çok bir yayın gönderilir.

SROPT (10 basamaklı işaretli tamsayı)

Aşağıdaki seçeneklerden biri belirtilmelidir. Tek bir seçenek belirlenebilir.

Diğer seçenekler: Aşağıdaki seçenek, kuyruk yöneticisi susturulduğunda ne olacağını denetler:

SRFIQ

Kuyruk yöneticisi susturma durumundaysa MQSUBRQ çağrısı başarısız olur.

Varsayılan seçenek: Önceden açıklanan seçenek gerekli değilse, aşağıdaki seçenek kullanılmalıdır:

SRNONE

Başka seçenek belirtilmediğini belirtmek için bu değeri kullanın; tüm seçenekler varsayılan değerlerini varsayar.

SRNONE, program belgelerine yardımcı olur. Değeri sıfır olduğu için bu seçeneğin başka bir seçenekle birlikte kullanılması amaçlanmasa da, bu kullanım algılanamaz.

SRSID (4 baytlık karakter dizilimi)

Bu yapı tanıtcısıdır; değer şu olmalıdır:

SRSIDV

Abonelik İsteği SROPT yapısına ilişkin tanıtcı.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri SRSIDV ' dir.

SRVER (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır; değer şu olmalıdır:

SRVER1

Version-1 Abonelik İsteği Seçenekleri yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

SRVERC

Abonelik İsteği Seçenekleri yapısının geçerli sürümü.

Bu her zaman bir giriş alanıdır. Bu alanın ilk değeri SRVER1' dir.

Başlangıçtaki değerler

Çizelge 734. MQSRO 'daki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
SRSID	SRSIDV	'SRO¬'
SRVER	SRVER1	1
SROPT	SRNONE	0
SRNMP	Yok	0

Notlar:

- ¬ simgesi, tek bir boş karakteri temsil eder.
- Boş değerli dizilim ya da boşluk değeri, C dizgisindeki boş dizgiyi ve diğer programlama dillerindeki boş karakterleri gösterir.

RPG bildirim

```
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
```

MQSTS yapısı, MQSTAT komutunun döndürdüğü durum yapısındaki verileri tanımlar.

Genel Bakış

Karakter kümesi ve kodlama: MQSTS ' deki karakter verileri yerel kuyruk yöneticisinin karakter kümesinde; bu *CodedCharSetId* kuyruk yöneticisi özneliği tarafından verilir. MQSTS ' deki sayısal veriler yerli makine kodlamasında; bu, *ENNAT* tarafından verilir.

Kullanım: Durum bilgilerini almak için MQSTAT komutu kullanılır. Bu bilgiler bir MQSTS yapısında döndürülür. MQSTAT hakkında bilgi için bkz. [“IBM i üzerinde MQSTAT \(Durum bilgilerini al\)” sayfa 1335](#).

- [“Alanlar” sayfa 1211](#)
- [“Başlangıçtaki değerler” sayfa 1214](#)
- [“RPG bildirimi” sayfa 1214](#)

Alanlar

MQSTS yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

STSCC (10 basamaklı işaretli tamsayı)

Bu, MQSTS yapısında bildirilen ilk hatanın sonucu olan tamamlanma kodudur.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri CCOK ' dur.

STSFCC (10 basamaklı işaretli tamsayı)

Bu, başarısız olan zamanuyumsuz koyma çağrılarının sayısıdır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır.

STSOBJN (48 baytlık karakter dizgisi)

Bu, ilk hataya dahil olan nesnenin yerel adıdır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 48 boş karakterdir.

STSOQMGR (48 baytlık karakter dizgisi)

Bu, *STSOBJN* nesnesinin tanımlandığı kuyruk yöneticisinin adıdır. İlk boş değere kadar ya da alanın sonuna kadar tümüyle boş bırakılan bir ad, uygulamanın bağlı olduğu kuyruk yöneticisini (yerel kuyruk yöneticisi) gösterir.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 48 boş karakterdir.

STS00 (10 basamaklı işaretli tamsayı)

Raporlanan nesneyi açmak için kullanılan STS00 . Yalnızca MQSTS ya da üstü sürüm 2 'de bulunur.

STS00 değeri, MQSTAT **STYPE** parametresinin değerine bağlıdır.

DURUM

Sfır.

DURUM

Sfır.

DURUM

Hata oluştuğunda kullanılan STS00 . Hatanın nedeni, MQSTS yapısındaki *STSCC* ve *STSRC* alanlarında raporlanır.

STS00 bir çıkış alanıdır. İlk değeri sıfırdır.

STSOS (MQCHARV)

Raporlanan başarısız olan nesnenin uzun nesne adı. Yalnızca MQSTS ya da üstü sürüm 2 'de bulunur. STSOS , uzunluk üst sınırı 10240 olan bir MQCHARV alanıdır. MQCHARV yapısının nasıl kullanılacağına ilişkin açıklamalar için [MQCHARV](#) kısmına bakın.

STSOS yorumu, MQSTAT **STYPE** parametresinin değerine bağlıdır.

DURUM

Bu, MQPUT işleminde kullanılan kuyruğun ya da konunun başarısız olan uzun nesne adıdır.

DURUM

Sıfır uzunluklu dizgi

DURUM

Bu, yeniden bağlantının başarısız olmasına neden olan nesnenin uzun nesne adıdır.

STSOS bir çıkış alanıdır. İlk değeri sıfır uzunluklu bir dizedir.

STSOT (10 basamaklı işaretli tamsayı)

ObjectName içinde adlandırılmakta olan nesnenin tipi. Olası değerler şunlardır:

OTALSQ

Diğer ad kuyruğu.

OTLOCQ

Yerel kuyruk.

OTMODQ

Model kuyruğu.

OTQ

Sıraya girin.

OTREMQ

Uzak kuyruk.

OTTOP

Konu.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri OTQ 'dur.

STSRC (10 basamaklı işaretli tamsayı)

Bu, MQSTS yapısında bildirilen ilk hatadan kaynaklanan neden kodudur.

Bu her zaman bir çıkış alanıdır. Bu alanın ilk değeri RCNONE.

STSRBJN (48 baytlık karakter dizisi)

Bu, yerel kuyruk yöneticisi adı çözdükten sonra *STSOBJN* içinde adı belirtilen hedef kuyruğun adıdır. Döndürülen ad, *STSRQMGR* ile tanıtilen kuyruk yöneticisinde bulunan bir kuyruğun adıdır.

Boş olmayan bir değer, nesne göz atma, giriş ya da çıkış (ya da herhangi bir birleşim) için açılmış tek bir kuyruksa döndürülür. Açılan nesne aşağıdakilerden biriye, *STSRBJN* boş olarak ayarlanır:

- Konu
- Bir kuyruk, ancak göz atma, giriş ya da çıkış için açılmadı

Bu bir çıkış alanıdır. Bu alanın ilk değeri 48 boş karakterdir.

STSRQMGR (48 baytlık karakter dizilimi)

Bu, yerel kuyruk yöneticisi adı çözdükten sonra hedef kuyruk yöneticisinin adıdır. Döndürülen ad, *STSRBJN* ile tanıtilen kuyruğun iyisi olan kuyruk yöneticisinin adıdır. *STSRQMGR* , yerel kuyruk yöneticisinin adı olabilir.

*STSR**OBJN* , yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun sahip olduğu paylaşılan bir kuyruksa, *STSR**QMGR* , kuyruk paylaşım grubunun adıdır. Kuyruk başka bir kuyruk paylaşım grubuna aitse, *STSR**OBJN* kuyruk paylaşım grubunun adı ya da kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisinin adı olabilir (döndürülen değerin türü, yerel kuyruk yöneticisinde var olan kuyruk tanımlamalarına göre belirlenir).

Boş olmayan bir değer, nesne göz atma, giriş ya da çıkış (ya da herhangi bir birleşim) için açılmış tek bir kuyruksa döndürülür. Açılan nesne aşağıdakilerden biriye, *STSR**QMGR* boş olarak ayarlanır:

- Konu
- Bir kuyruk, ancak göz atma, giriş ya da çıkış için açılmadı
- OOBNDN belirtilmiş bir küme kuyruğu (ya da **DefBind** kuyruk özniteliği OOBNDN değerine sahip olduğunda geçerli olan OOBNDQ ile)

Bu bir çıkış alanıdır. Bu alanın ilk değeri 48 boş karakterdir.

STSSC (10 basamaklı işaretli tamsayı)

Bu, başarılı olan zamanuyumsuz koyma çağrılarının sayısıdır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dir.

STSSID (4 baytlık karakter dizgisi)

Bu, yapı tanıtıcısıdır. Değer şöyle olmalıdır:

STSSID

Durum raporlama yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri STSSID 'dir.

STSSO (10 basamaklı işaretli tamsayı)

Hatalı aboneliği açmak için kullanılan STSSO . Yalnızca MQSTS ya da üstü sürüm 2 'de bulunur.

STSSO yorumu, MQSTAT **STYPE** parametresinin değerine bağlıdır.

DURUM

Sıfır.

DURUM

Sıfır.

DURUM

Hata oluştuğunda kullanılan STSSO . Hatanın nedeni, MQSTS yapısındaki *STSCC* ve *STSRC* alanlarında raporlanır. Başarısızlık bir konuya abone olma ile ilgili değilse, döndürülen değer sıfırdır.

STSSO bir çıkış alanıdır. İlk değeri sıfırdır.

STSSUN (MQCHARV)

Başarısız olan aboneliğin adı. Yalnızca MQSTS ya da üstü sürüm 2 'de bulunur.

STSSUN , 10240 uzunluk üst sınırı olan bir MQCHARV alanıdır. MQCHARV yapısının nasıl kullanılacağına ilişkin açıklamalar için [MQCHARV](#) kısmına bakın.

STSSUN yorumu, MQSTAT **STYPE** parametresinin değerine bağlıdır.

DURUM

Sıfır uzunluklu dizgi.

DURUM

Sıfır uzunluklu dizgi.

DURUM

Yeniden bağlantının başarısız olmasına neden olan aboneliğin adı. Abonelik adı yoksa ya da hata bir abonelikle ilgili değilse, bu, sıfır uzunluklu bir dizedir.

STSSUN bir çıkış alanıdır. İlk değeri sıfır uzunluklu bir dizedir.

STSVR (10 basamaklı işaretli tamsayı)

Bu, yapı sürümü numarasıdır. Değer şöyle olmalıdır:

STSVR1

Durum raporlama yapısına ilişkin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

STSVRC

Durum raporlama yapısının geçerli sürümü.

Bu alanın ilk değeri STSVR1' dir.

STSWC (10 basamaklı işaretli tamsayı)

Bu, bir uyarıyla tamamlanan zamanuyumsuz koyma çağrılarının sayısıdır.

Bu bir çıkış alanıdır. Bu alanın ilk değeri 0 'dır.

Başlangıçtaki değerler

Çizelge 735. MQSTS 'deki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
STSSID	STSID	
STSVR	STSVRC	STSVR1
STSCC	CCOK	0
STSRC	RCNONE	0
STSSC	Yok	0
STSWC	Yok	0
STSF	Yok	0
STSOT	Yok	0
STSOBJN	Yok	Boşluklar
STSOQMGR	Yok	Boşluklar
STSRBJN	Yok	Boşluklar
STSRQMGR	Yok	Boşluklar
STSOS	MQCHARV için tanımlanan adlar ve değerler	
STSSUN	MQCHARV için tanımlanan adlar ve değerler	
STS00	Yok	0
STSS0	Yok	0

RPG bildirim

D*..1.....2.....3.....4.....5.....6.....7..
D* MQSTS Structure
D*

```

D* Structure identifier
D STSSID 1 4
D* Structure version number
D STSVER 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 STSFC 25 28I 0
D* Object type
D STSOT 29 32I 0
D* Object name
D STSOBJN 33 80
D* Object queue manager
D STSQMGR 81 128
D* Resolved object name
D STSR OBJN 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-Tetikleyici iletisi

MQTM yapısı, kuyruk yöneticisi tarafından bir kuyruk için bir tetikleyici olayı oluştuğunda tetikleyici uygulamasına gönderilen tetikleyici iletideki verileri tanımlar.

Genel Bakış

Amaç: Bu yapı, IBM MQ çerçeve arabirimlerinden biri olan IBM MQ Tetikleyici İzleme Arabirimi 'nin (TMI) bir parçasıdır.

Biçim adı: FMTM.

Karakter kümesi ve kodlama: MQTM 'deki karakter verileri, MQTM' yi oluşturan kuyruk yöneticisinin karakter kümesinde yer almaktadır. MQTM 'deki sayısal veriler, MQTM' yi oluşturan kuyruk yöneticisinin makine kodlamasında yer aldı.

MQTM ' nin karakter kümesi ve kodlaması *MDCSI* ve *MDENC* alanları tarafından aşağıdaki alanlarda verilir:

- MQMD (MQTM yapısı ileti verilerinin başındadır) ya da
- MQTM yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Kullanım: Bir tetikleyici uygulamasının tetikleyici iletisindeki bilgilerin bir kısmını ya da tümünü tetikleyici izleyici uygulaması tarafından başlatılan uygulamaya aktarması gerekebilir. Başlatılan uygulamanın gereksinim duyabileceği bilgiler arasında *TMQN*, *TMTD* ve *TMUD* yer alır. Tetikleyici izleyici uygulaması MQTM yapısını doğrudan başlatılan uygulamaya geçirebilir ya da ortamın izin verdiği ve başlatılan uygulama için uygun olduğuna bağlı olarak bir MQTMC2 yapısı geçirebilir. MQTMC2 hakkında bilgi için bkz. [“MQTMC2 \(Tetikleyici iletisi 2 karakterli biçimi\)- IBM i” sayfa 1220](#).

- IBM sistemlerinde, IBM MQ ile sağlanan tetikleyici izleyicisi uygulaması, başlatılan uygulamaya bir MQTMC2 yapısı geçirir.

Tetikleyicilerle ilgili bilgi için [Tetikleme önkoşulları](#) başlıklı konuya bakın.

- [“Tetikleyici iletisine ilişkin MQMD” sayfa 1216](#)
- [“Alanlar” sayfa 1217](#)
- [“Başlangıçtaki değerler” sayfa 1219](#)
- [“RPG bildirim” sayfa 1219](#)

Tetikleyici iletisine ilişkin MQMD

Çizelge 736. Kuyruk yöneticisi tarafından oluşturulan bir tetikleyici iletisine ilişkin MQMD 'deki alanlara ilişkin ayarlar

MQMD 'deki Alan	Kullanılan değer
MDSID	MDSIDV
MDVER	MDVER1
MDREP	YAYINI
MDMT	MTDGRM
MDEXP	EIULIM
MDFB	FBNONE
MDENC	GIRIŞ
MDCSI	Kuyruk yöneticisinin CodedCharSetId özneliği
MDFMT	FMTM
MDPRI	Başlatma kuyruğunun DefPriority özneliği
MDPER	PENPER
MDMID	Benzersiz bir değer
MDCID	TAMAMDİR.
MDBOC	0
MDRQ	Boşluklar
MDRM	Kuyruk yöneticisinin adı
MDUID	Boşluklar
MDACC	YOK
MDAID	Boşluklar
MDPAT	ATQM ya da ileti kanalı aracısı için uygun olduğu şekilde
MDPAN	Kuyruk yöneticisi adının ilk 28 baytı
MDPD	Tetikleyici iletisinin gönderildiği tarih
MDPT	Tetikleyici iletisinin gönderildiği zaman

MQMD ' deki Alan	Kullanılan değer
MDAOD	Boşluklar

Aşağıdakiler dışında benzer değerleri ayarlamak için bir tetikleyici iletisi oluşturan bir uygulama önerilir:

- *MDPRI* alanı PRQDEF olarak ayarlanabilir (kuyruk yöneticisi, ileti konduğunda başlatma kuyruğuna ilişkin varsayılan önceliğe çevirir).
- *MDRM* alanı boş olarak ayarlanabilir (kuyruk yöneticisi, yerleştirdiği iletide bu değeri yerel kuyruk yöneticisinin adına çevirir).
- Bağlam alanları, uygulama için uygun şekilde ayarlanmalıdır.

Alanlar

MQTM yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

TMAI (256 baytlık karakter dizgisi)

Uygulama tanıtıcısı.

Bu, başlatılacak uygulamayı tanıtan ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılan bir karakter dizilimidir. Kuyruk yöneticisi bu alanı, *TMPN* alanıyla tanıtılan süreç nesnesinin **App1Id** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. "[IBM i üzerinde süreç tanımlamalarına ilişkin öznitelikler](#)" sayfa 1374 . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

TMAI ' in anlamı, tetikleyici-izleyici uygulaması tarafından belirlenir. IBM MQ tarafından sağlanan tetikleyici, *TMAI* ' in yürütülebilir bir programın adı olmasını gerektirir.

Bu alanın uzunluğu LNPROA tarafından verilir. Bu alanın ilk değeri 256 boş karakterdir.

TMAT (10 basamaklı işaretli tamsayı)

Uygulama tipi.

Bu, başlatılacak programın türünü tanımlar ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılır. Kuyruk yöneticisi bu alanı, *TMPN* alanıyla tanıtılan süreç nesnesinin **App1Type** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. "[IBM i üzerinde süreç tanımlamalarına ilişkin öznitelikler](#)" sayfa 1374 . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

TMAT aşağıdaki standart değerlerden birine sahip olabilir. Kullanıcı tanımlı tipler de kullanılabilir, ancak ATUFST-ATULST aralığındaki değerlerle sınırlandırılmalıdır:

SAATCICS

CICS hareketi.

ATVSE.

CICS/VSE hareketi.

AT400

IBM i uygulaması.

ATUFST

Kullanıcı tanımlı uygulama tipi için en düşük değer.

ATULST

Kullanıcı tanımlı uygulama tipi için en yüksek değer.

Bu alanın ilk değeri 0 'dır.

TMED (128 baytlık karakter dizgisi)

Ortam verileri.

Bu, başlatılacak uygulamayla ilgili ortamla ilgili bilgileri içeren ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılan bir karakter dizilimidir. Kuyruk yöneticisi bu alanı, *TMPN* alanıyla tanıtilan süreç nesnesinin **EnvData** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. [“IBM i üzerinde süreç tanımlamalarına ilişkin öznitelikler” sayfa 1374](#) . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

Bu alanın uzunluğu LNPROE tarafından verilir. Bu alanın ilk değeri 128 boş karakterdir.

TMPN (48 baytlık karakter dizgisi)

Süreç nesnesinin adı.

Bu, tetiklenen kuyruk için belirtilen kuyruk yöneticisi işlem nesnesinin adıdır ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılabilir. Kuyruk yöneticisi bu alanı, *TMQN* alanıyla tanıtilan kuyruğun **ProcessName** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. [“Kuyruklara ilişkin öznitelikler” sayfa 1345](#) .

Alanın tanımlı uzunluğundan daha kısa olan adlar her zaman sağa boşluklarla doldurulur; boş bir karakterle zamanından önce sona erdirilmezler.

Bu alanın uzunluğu LNPRON tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

TMQN (48 baytlık karakter dizgisi)

Tetiklenen kuyruğun adı.

Bu, tetikleyici olayının oluştuğu kuyruğun adıdır ve tetikleyici izleme uygulaması tarafından başlatılan uygulama tarafından kullanılır. Kuyruk yöneticisi, tetiklenen kuyruğun **QName** özniteliğinin değeriyle bu alanı kullanıma hazırlar; bu özniteliğin ayrıntıları için bkz. [“Kuyruklara ilişkin öznitelikler” sayfa 1345](#) .

Alanın tanımlı uzunluğundan daha kısa olan adlar sağa boşluklarla doldurulur; bunlar boş bir karakterle zamanından önce sona erdirilmez.

Bu alanın uzunluğu LNQN tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

TMSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

TMSIDV

Tetikleyici ileti yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri TMSIDV 'dir.

TMTD (64 baytlık karakter dizgisi)

Tetikleyici veri.

Bu, tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılmak üzere serbest biçimli verilerdir. Kuyruk yöneticisi bu alanı, *TMQN* alanıyla tanıtilan kuyruğun **TriggerData** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. [“Kuyruklara ilişkin öznitelikler” sayfa 1345](#) . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

Bu alanın uzunluğu LNTRGD tarafından verilir. Bu alanın ilk değeri 64 boş karakterdir.

TMUD (128 baytlık karakter dizgisi)

Kullanıcı verileri.

Bu, başlatılacak uygulamayla ilgili kullanıcı bilgilerini içeren ve tetikleyici iletisini alan tetikleyici izleme uygulaması tarafından kullanılan bir karakter dizilimidir. Kuyruk yöneticisi bu alanı, *TMPN* alanıyla tanıtilan süreç nesnesinin **UserData** özniteliğinin değeriyle başlatır; bu özniteliğin ayrıntıları için bkz. [“IBM i üzerinde süreç tanımlamalarına ilişkin öznitelikler” sayfa 1374](#) . Bu verilerin içeriği kuyruk yöneticisi için önemli değildir.

Bu alanın uzunluğu LNPROU tarafından verilir. Bu alanın ilk değeri 128 boş karakterdir.

TMVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

TMVER1

Tetikleyici ileti yapısı için sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

TMVERC

Tetikleyici ileti yapısının yürürlükteki sürümü.

Bu alanın ilk değeri TMVER1' dir.

Başlangıçtaki değerler

Çizelge 737. MQTM içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
TMSID	TMSIDV	' TM ' ' '
TMVER	TMVER1	1
TMQN	Yok	Boşluklar
TMPN	Yok	Boşluklar
TMTD	Yok	Boşluklar
TMAT	Yok	0
TMAI	Yok	Boşluklar
TMED	Yok	Boşluklar
TMUD	Yok	Boşluklar

Notlar:

1. - simgesi, tek bir boş karakteri temsil eder.

RPG bildirimi

```
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
```

Tetikleyici izleme uygulaması başlatma kuyruğundan bir tetikleyici iletisi (MQTM) aldığında, tetikleyici izleme programının tetikleyici iletisindeki bilgilerin bir kısmını ya da tümünü tetikleyici izleme programı tarafından başlatılan uygulamaya geçirmesi gerekebilir.

Genel Bakış

Amaç: Başlatılan uygulamanın gereksinim duyabileceği bilgiler arasında *TC2QN*, *TC2TD* ve *TC2UD* yer alır. Tetikleyici izleyici uygulaması MQTM yapısını doğrudan başlatılan uygulamaya geçirebilir ya da ortamın izin verdiği ve başlatılan uygulama için uygun olduğuna bağlı olarak bir MQTM C2 yapısı geçirebilir.

Bu yapı, IBM MQ çerçeve arabirimlerinden biri olan IBM MQ Trigger Monitor Interface (TMI) olanağının bir parçasıdır.

Karakter kümesi ve kodlama: MQTM C2 içindeki karakter verileri, yerel kuyruk yöneticisinin karakter kümesinde yer almaktadır; bu, **CodedCharSetId** kuyruk yöneticisi özneliği tarafından verilir.

Kullanım: MQTM C2 yapısı, MQTM yapısının biçimine benzer. Fark, MQTM 'deki karakter olmayan alanların MQTM C2 ' de aynı uzunlukta karakter alanlarına değiştirilmesi ve kuyruk yöneticisi adının yapının sonuna eklenmeleridir.

- IBM sistemlerinde, IBM MQ ile sağlanan tetikleyici izleyicisi uygulaması, başlatılan uygulamaya bir MQTM C2 yapısı geçirir.
- [“Alanlar” sayfa 1220](#)
- [“Başlangıçtaki değerler” sayfa 1221](#)
- [“RPG bildirim” sayfa 1221](#)

Alanlar

MQTM C2 yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

TC2AI (256 baytlık karakter dizgisi)

Uygulama tanıtıcısı.

MQTM yapısındaki *TMAI* alanına bakın.

TC2AT (4 baytlık karakter dizilimi)

Uygulama tipi.

Bu alan, özgün tetikleyici iletisinin MQTM yapısındaki *TMAT* alanındaki değer ne olursa olsun, her zaman boşluk içerir.

TC2ED (128 baytlık karakter dizgisi)

Ortam verileri.

MQTM yapısındaki *TMED* alanına bakın.

TC2PN (48 baytlık karakter dizgisi)

Süreç nesnesinin adı.

MQTM yapısındaki *TMPN* alanına bakın.

TC2QMN (48 baytlık karakter dizilimi)

Kuyruk yöneticisi adı.

Bu, tetikleme olayının oluştuğu kuyruk yöneticisinin adıdır.

TC2QN (48 baytlık karakter dizilimi)

Tetiklenen kuyruğun adı.

MQTM yapısındaki *TMQN* alanına bakın.

TC2SID (4 baytlık karakter dizilimi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

TCSIDV

Tetikleyici ileti (karakter biçimi) yapısına ilişkin tanıtıcı.

TC2TD (64 baytlık karakter dizgisi)

Tetikleyici veri.

MQTM yapısındaki *TMTD* alanına bakın.

TC2UD (128 baytlık karakter dizilimi)

Kullanıcı verileri.

MQTM yapısındaki *TMUD* alanına bakın.

TC2VER (4 baytlık karakter dizilimi)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

TCVER2

Sürüm 2 tetikleyici iletisi (karakter biçimi) yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

TTCVERC

Tetikleyici ileti (karakter biçimi) yapısının yürürlükteki sürümü.

Başlangıçtaki değerler

Çizelge 738. MQTMC2 içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
<i>TC2SID</i>	TCSIDV	'TMC¬'
<i>TC2VER</i>	TCVER2	'¬¬¬2'
<i>TC2QN</i>	Yok	Boşluklar
<i>TC2PN</i>	Yok	Boşluklar
<i>TC2TD</i>	Yok	Boşluklar
<i>TC2AT</i>	Yok	Boşluklar
<i>TC2AI</i>	Yok	Boşluklar
<i>TC2ED</i>	Yok	Boşluklar
<i>TC2UD</i>	Yok	Boşluklar
<i>TC2QMN</i>	Yok	Boşluklar

Notlar:

- ¬ simgesi, tek bir boş karakteri temsil eder.

RPG bildirimi

D*..1.....2.....3.....4.....5.....6.....7..

D*	MQTM2 Structure		
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

IBM i üzerinde MQWIH (İş bilgileri üstbilgisi)

MQWIH yapısı, z/OS iş yükü yöneticisi tarafından işlenecek bir iletinin başında bulunması gereken bilgileri tanımlar.

Genel Bakış

Biçim adı: FMWIH.

Karakter kümesi ve kodlama: MQWIH yapısındaki alanlar, MQWIH ' den önce gelen üstbilgi yapısındaki *MDCSI* ve *MDENC* alanları tarafından verilen karakter kümesi ve kodlamadır ya da MQWIH uygulama iletisi verilerinin başlangıcında ise MQMD yapısındaki bu alanlar tarafından.

Karakter takımı, kuyruk adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır.

Kullanım: Bir ileti z/OS iş yükü yöneticisi tarafından işlenecek olursa, iletinin bir MQWIH yapısıyla başlaması gerekir.

- [“Alanlar” sayfa 1222](#)
- [“Başlangıçtaki değerler” sayfa 1224](#)
- [“RPG bildirimi” sayfa 1224](#)

Alanlar

MQWIH yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

WICSI (10 basamaklı işaretli tamsayı)

MQWIH ' yi izleyen verilerin karakter kümesi tanıtıcısı.

MQWIH yapısını izleyen verilerin karakter kümesi tanıtıcısını belirtir; MQWIH yapısının kendisindeki karakter verileri için geçerli değildir.

MQPUT ya da MQPUT1 çağrısında, uygulamanın bu alanı veriler için uygun değere ayarlaması gerekir. Aşağıdaki özel değer kullanılabilir:

CSINHT

Bu yapının karakter kümesi tanıtıcısını edin.

Bu yapı, *izleyen* verilerdeki karakter verileri, bu yapıyla aynı karakter kümesinde yer almaktadır.

Kuyruk yöneticisi, iletide gönderilen yapıdaki bu değeri, yapının gerçek karakter kümesi tanıtıcısına değiştirir. Hata oluşmaması durumunda, MQGET çağrısı CSINHT değerini döndürmez.

MQMD ' deki *MDPAT* alanının değeri ATBRKR ise, CSINHT kullanılamaz.

Bu alanın ilk deęeri CSUNDF 'dir.

WIENC (10 basamaklı iřaretli tamsayı)

MQWIIH ' yi izleyen verilerin sayısal kodlaması.

MQWIIH yapısını izleyen verilerin sayısal kodlamasını belirtir; MQWIIH yapısının kendisindeki sayısal veriler için geerli deęildir.

MQPUT ya da MQPUT1 aęrısında, uygulamanın bu alanı veriler için uygun deęere ayarlaması gerekir.

Bu alanın ilk deęeri 0 'dır.

WIFLG (10 basamaklı iřaretli tamsayı)

İřaretler

Deęer řöyle olmalıdır:

KAZAN

Bayrak yok.

Bu alanın ilk deęeri WINONE.

WIFMT (8 baytlık karakter dizgisi)

MQWIIH ' yi izleyen verilerin biçim adı.

MQWIIH yapısını izleyen verilerin biçim adını belirtir.

MQPUT ya da MQPUT1 aęrısında, uygulamanın bu alanı veriler için uygun deęere ayarlaması gerekir. Bu alanı kodlamaya iliřkin kurallar, MQMD ' deki *MDFMT* alanıyla aynıdır.

Bu alanın uzunluęu LNFMT tarafından verilir. Bu alanın ilk deęeri FMNONE.

WILEN (10 basamaklı iřaretli tamsayı)

MQWIIH yapısının uzunluęu.

Deęer řöyle olmalıdır:

WILEN1

version-1 iř bilgileri üstbilgi yapısının uzunluęu.

Ařaęıdaki deęiřmez, yürürlükteki sürümün uzunluęunu belirtir:

WILENC

İř bilgileri üstbilgi yapısının yürürlükteki sürümünün uzunluęu.

Bu alanın ilk deęeri WILEN1' dir.

WIRSV (32 baytlık karakter dizgisi)

Ayrılmıř.

Bu ayrılmıř bir alandır; boş olmalıdır.

WISID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Deęer řöyle olmalıdır:

WISIDV

İř bilgileri üstbilgi yapısına iliřkin tanıtıcı.

Bu alanın ilk deęeri WISIDV 'dir.

WISNM (32 baytlık karakter dizgisi)

Hizmet adı.

Bu, iletiyi iřleyecek hizmetin adıdır.

Bu alanın uzunluęu LNSVNM tarafından verilir. Bu alanın ilk deęeri 32 boş karakterdir.

WISST (8 baytlık karakter dizgisi)

Hizmet adımı adı.

Bu, iletinin ilgili olduğu *WISNM* adımının adıdır.

Bu alanın uzunluğu LNSVST tarafından verilir. Bu alanın ilk değeri 8 boş karakterdir.

WITOK (16 baytlık bit dizgisi)

İleti simgesi.

Bu, iletiyi benzersiz olarak tanıtan bir ileti simgesidir.

MQPOT ve MQPUT1 çağrıları için bu alan yoksaılır. Bu alanın uzunluğu LNMTOk tarafından verilir. Bu alanın ilk değeri MTKNON değeridir.

WIVER (10 basamaklı işaretili tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

WIVER1

Version-1 iş bilgileri üstbilgi yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

WIVERC

İş bilgileri üstbilgi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri WIVER1' dir.

Başlangıçtaki değerler

Çizelge 739. MQWIH ' deki Alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
WISID	WISIDV	'WIH↵'
WIVER	WIVER1	1
WILEN	WILEN1	120
WIENC	Yok	0
WICSI	CSUNDF	0
WIFMT	FMNONE	Boşluklar
WIFLG	KAZANÇ	0
WISNM	Yok	Boşluklar
WISST	Yok	Boşluklar
WITOK	MTKN	Boş Değerler
WIRSV	Yok	Boşluklar
Notlar:		
1. ↵ simgesi, tek bir boş karakteri temsil eder.		

RPG bildirim

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'00000000000000-
D 0000000000000000')
D* Reserved
D WIRSV 89 120 INZ

```

IBM i

IBM i üzerinde MQXQH (İletim kuyruğu üstbilgisi)

MQXQH yapısı, iletim kuyruklarındayken iletilerin uygulama ileti verilerine önek olarak eklenen bilgileri tanımlar.

Genel Bakış

Amaç: İletim kuyruğu, uzak kuyruklara (yerel kuyruk yöneticisine ait olmayan kuyruklara) yönelik iletileri geçici olarak tutan özel bir yerel kuyruk tipidir. Bir iletim kuyruğu, USTRAN değerine sahip **Usage** kuyruk özneliğiyle gösterilir.

Biçim adı: FMXQH.

Karakter kümesi ve kodlama: MQXQH ' deki veriler, **CodedCharSetId** kuyruk yöneticisi özneliği ve C programlama dili için ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır.

MQXQH ' nin karakter kümesi ve kodlaması aşağıdaki alanlarda *MDCSI* ve *MDENC* alanlarına ayarlanmalıdır:

- Ayrı MQMD (MQXQH yapısı ileti verilerinin başındadır) ya da
- MQXQH yapısından önce gelen üstbilgi yapısı (diğer tüm durumlar).

Kullanım: İletim kuyruğundaki bir iletinin *iki* ileti tanımlayıcısı vardır:

- Bir ileti tanımlayıcısı, ileti verilerinden ayrı olarak saklanır; bu, *ayrı ileti tanımlayıcısı* olarak adlandırılır ve ileti iletim kuyruğuna yerleştirildiğinde kuyruk yöneticisi tarafından oluşturulur. Ayrı ileti tanımlayıcısındaki bazı alanlar, MQPUT ya da MQPUT1 çağrısındaki uygulama tarafından sağlanan ileti tanımlayıcısından kopyalanır.

Ayrı ileti tanımlayıcısı, ileti iletim kuyruğundan kaldırıldığında MQGET çağrısının **MSGDSC** değiştirgesinde uygulamaya döndürülen iletidir.

- İleti verilerinin bir parçası olarak MQXQH yapısında ikinci bir ileti tanımlayıcısı saklanır; bu, *yerleşik ileti tanımlayıcısı* olarak adlandırılır ve MQPUT ya da MQPUT1 çağrısında (ikincil çeşitlemelerle) uygulama tarafından sağlanan ileti tanımlayıcısının bir kopyasıdır.

Yerleşik ileti tanımlayıcısı her zaman bir version-1 MQMD 'dir. Uygulama tarafından konan iletide MQMD 'deki version-2 alanlarından biri ya da daha fazlası için varsayılan olmayan değerler varsa, MQMDE yapısı MQXQH' yi izler ve ardından uygulama ileti verileri (varsa) gelir. MQMDE aşağıdakilerden biri olabilir:

- Kuyruk yöneticisi tarafından üretilir (uygulama iletiyi koymak için version-2 MQMD kullanıyorsa) ya da

- Uygulama ileti verilerinin başlangıcında zaten var (uygulama iletiyi koymak için version-1 MQMD kullanıyorsa).

Yerleşik ileti tanımlayıcısı, ileti son hedef kuyruktan kaldırıldığında MQGET çağrısının **MSGDSC** değıştirilmesinde uygulamaya döndürülen iletidir.

- “Ayrı ileti tanımlayıcısındaki alanlar” sayfa 1226
- “Katiştirilmiş ileti tanımlayıcısındaki alanlar” sayfa 1227
- “Uzak kuyruklara ileti konulması” sayfa 1227
- “İletilerin doğrudan iletim kuyruklarına yerleştirilmesi” sayfa 1228
- “İletim kuyruklarından ileti alma” sayfa 1228
- “Alanlar” sayfa 1228
- “Başlangıçtaki değerler” sayfa 1229
- “RPG bildirim” sayfa 1229

Ayrı ileti tanımlayıcısındaki alanlar

Ayrı ileti tanımlayıcısındaki alanlar, kuyruk yöneticisi tarafından aşağıdaki listede gösterildiği gibi ayarlanır. Kuyruk yöneticisi version-2 MQMD ' yi desteklemiyorsa, işlev kaybı olmadan version-1 MQMD kullanılır.

Çizelge 740. Kullanılan ayrı ileti tanımlayıcısındaki ve değerlerindeki alanlar

Ayrı MQMD ' deki alan	Kullanılan değer
MDSID	MDSIDV
MDVER	MDVER2
MDREP	Gömülü ileti tanımlayıcısından kopyalandı, ancak ROAUXM tarafından tanımlanan bitler sıfır olarak ayarlandı. (Bu, bir ileti bir iletim kuyruğuna yerleştirildiğinde ya da iletiden kaldırıldığında bir COA ya da COD rapor iletinin oluşturulmasını önler.)
MDMT	Yerleşik ileti tanımlayıcısından kopyalandı.
MDEXP	Yerleşik ileti tanımlayıcısından kopyalandı.
MDFB	Yerleşik ileti tanımlayıcısından kopyalandı.
MDENC	GIRIş
MDCSI	Kuyruk yöneticisinin CodedCharSetId özniteliği.
MDFMT	FMXQH
MDPRI	Yerleşik ileti tanımlayıcısından kopyalandı.
MDPER	Yerleşik ileti tanımlayıcısından kopyalandı.
MDMID	Kuyruk yöneticisi tarafından yeni bir değer oluşturulur. Bu ileti tanıtıcısı, kuyruk yöneticisinin yerleşik ileti tanımlayıcısı için oluşturduğu <i>MDMID</i> ' den farklıdır (daha önce açıklanan açıklamaya bakın).
MDCID	Yerleşik ileti tanımlayıcısından <i>MDMID</i> .
MDBOC	0
MDRQ	Yerleşik ileti tanımlayıcısından kopyalandı.
MDRM	Yerleşik ileti tanımlayıcısından kopyalandı.
MDUID	Yerleşik ileti tanımlayıcısından kopyalandı.
MDACC	Yerleşik ileti tanımlayıcısından kopyalandı.
MDAID	Yerleşik ileti tanımlayıcısından kopyalandı.

Ayrı MQMD ' deki alan	Kullanılan değer
MDPAT	ÖZNKY
MDPAN	Kuyruk yöneticisi adının ilk 28 baytı.
MDPD	İletinin iletim kuyruğuna konma tarihi.
MDPT	İletinin iletim kuyruğuna konma süresi.
MDAOD	Boşluklar
MDGID	GINONE
MDSEQ	1
MDOFF	0
MDMFL	MFNONE
MDOLN	OLUNDF

Katıştırılmış ileti tanımlayıcısındaki alanlar

Gömülü ileti tanımlayıcısındaki alanlar, aşağıdakiler dışında, MQPUT ya da MQPUT1 çağrısının **MSGDSC** değiştirilmesiyle aynı değerlere sahiptir:

- *MDVER* alanında her zaman MDVER1değeri vardır.
- *MDPRI* alanında PRQDEF değeri varsa, bu değer kuyruğun **DefPriority** özneliğinin değeriyle değiştirilir.
- *MDPER* alanında PEQDEF değeri varsa, bu değer kuyruğun **DefPersistence** özneliğinin değeriyle değiştirilir.
- *MDMID* alanında MINONE değeri varsa ya da PMNMID seçeneği belirtildiyse ya da ileti bir dağıtım listesi iletiliyse, *MDMID* , kuyruk yöneticisi tarafından oluşturulan yeni bir ileti tanımlayıcısıyla değiştirilir.

Bir dağıtım listesi iletilisi, farklı iletim kuyruklarına yerleştirilen daha küçük dağıtım listesi iletilerine bölündüğünde, yeni yerleşik ileti tanımlayıcılarının her birindeki *MDMID* alanı, özgün dağıtım listesi iletilisindekiyle aynıdır.

- PMNCID seçeneği belirtildiyse, *MDCID* yerine kuyruk yöneticisi tarafından oluşturulan yeni bir ilinti tanımlayıcısı konur.
- Bağlam alanları, **PMO** parametresinde belirtilen PM* seçenekleriyle belirtildiği şekilde ayarlanır; bağlam alanları şunlardır:

- *MDACC*
- *MDAID*
- *MDAOD*
- *MDPAN*
- *MDPAT*
- *MDPD*
- *MDPT*
- *MDUID*

- version-2 alanları (varsa) MQMD ' den kaldırılır ve version-2 alanlarından birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa MQMDE yapısına taşınır.

Uzak kuyruklara ileti konulması

: Bir uygulama uzak kuyruğa bir ileti koyduğunda (uzak kuyruğun adını doğrudan belirterek ya da uzak kuyruğun yerel tanımlamasını kullanarak), yerel kuyruk yöneticisi:

- Yerleşik ileti tanımlayıcısını içeren bir MQXQH yapısı yaratır
- Gerekliyse ve önceden yoksa bir MQMDE ekler
- Uygulama ileti verilerini ekler
- İletiyi uygun bir iletim kuyruğuna yerleştirir

İletilerin doğrudan iletim kuyruklarına yerleştirilmesi

Bir uygulamanın bir iletiyi doğrudan bir iletim kuyruğuna koyması da mümkündür. Bu durumda uygulama, uygulama iletileri verilerine bir MQXQH yapısıyla önek eklemeli ve alanları uygun değerlerle kullanıma hazırlamalıdır. Buna ek olarak, MQPUT ya da MQPUT1 çağrısının **MSGDSC** parametresindeki *MDFMT* alanı FMXQH değerine sahip olmalıdır.

Uygulama tarafından yaratılan MQXQH yapısındaki karakter verileri, yerel kuyruk yöneticisinin (**CodedCharSetId** kuyruk yöneticisi özneliği tarafından tanımlanan) karakter kümesinde olmalı ve tamsayı verileri yerel makine kodlamasında bulunmalıdır. Buna ek olarak, MQXQH yapısındaki karakter verileri, alanın tanımlı uzunluğuna kadar boşluklarla doldurulacaktır; kuyruk yöneticisi, boş ve sonraki karakterleri MQXQH yapısında boşluklara dönüştürmediği için, veriler boş bir karakter kullanılarak zamanından önce sona erdirilmemelidir.

Ancak, kuyruk yöneticisinin bir MQXQH yapısının var olup olmadığını ya da alanlar için geçerli değerlerin belirtildiğini denetlemediğini unutmayın.

İletim kuyruklarından ileti alma

Bir iletim kuyruğundan ileti alan uygulamaların, MQXQH yapısındaki bilgileri uygun şekilde işlemesi gerekir. Uygulama iletileri verilerinin başında MQXQH yapısının varlığı, MQGET çağrısının **MSGDSC** parametresindeki *MDFMT* alanında döndürülen FMXQH değeriyle gösterilir. **MSGDSC** değiştirgesindeki *MDCSI* ve *MDENC* alanlarında döndürülen değerler, MQXQH yapısındaki karakter kümesi ve tamsayı verilerinin kodlamasını gösterir. Uygulama iletileri verilerinin karakter kümesi ve kodlaması, yerleşik ileti tanımlayıcısındaki *MDCSI* ve *MDENC* alanları tarafından tanımlanır.

Alanlar

MQXQH yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırayla** açıklanır:

XQMD (MQMD1)

Özgün ileti tanımlayıcısı.

Bu, yerleşik ileti tanımlayıcısıdır ve ileti uzak kuyruğa ilk yerleştirildiğinde MQPUT ya da MQPUT1 çağrısında **MSGDSC** değiştirgesi olarak belirtilen ileti tanımlayıcı MQMD 'nin yakın bir kopyasıdır.

Not: Bu bir version-1 MQMD 'dir.

Bu yapıdaki alanların başlangıç değerleri, MQMD yapısındaki değerlerle aynıdır.

XQRQ (48 baytlık karakter dizilimi)

Hedef kuyruğun adı.

Bu, iletinin görünen nihai hedefi olan ileti kuyruğunun adıdır (örneğin, bu kuyruk *XQRQM* adresinde başka bir uzak kuyruğun yerel tanımı olarak tanımlandıysa, bu, gerçek nihai hedef olmayabilir).

İleti bir dağıtım listesi iletiyse (yani, yerleşik ileti tanımlayıcısındaki *MDFMT* alanı *FMDH* ise), *XQRQ* alanı boştur.

Bu alanın uzunluğu LNQN tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

XQRQM (48 baytlık karakter dizgisi)

Hedef kuyruk yöneticisinin adı.

Bu, iletinin görünen nihai hedefi olan kuyruğun iyisi olan kuyruk yöneticisinin ya da kuyruk paylaşım grubunun adıdır.

İleti bir dağıtım listesi iletiyse, XQRQM boş olur.

Bu alanın uzunluğu LNQMN tarafından verilir. Bu alanın ilk değeri 48 boş karakterdir.

XQSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

XQSIDV

İletim kuyruğu üstbilgi yapısına ilişkin tanıtıcı.

Bu alanın ilk değeri XQSIDV 'dir.

XQVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

XQVER1

İletim kuyruğu üstbilgi yapısına ilişkin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

XQVERC

İletim kuyruğu üstbilgi yapısının yürürlükteki sürümü.

Bu alanın ilk değeri XQVER1' dir.

Başlangıçtaki değerler

Çizelge 741. MQXQH içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
XQSID	XQSIDV	'XQH↯'
XQVER	XQVER1	1
XQRQ	Yok	Boşluklar
XQRQM	Yok	Boşluklar
XQMD	MQMD ile aynı adlar ve değerler; bkz. Çizelge 709 sayfa 1132	-

Notlar:

- ↯ simgesi, tek bir boş karakteri temsil eder.

RPG bildirim

```
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'0000000000000000-0000000000000000000000-000000000000')
D	XQ1CID	177	200		INZ(X'0000000000000000-0000000000000000000000-000000000000')
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'0000000000000000-0000000000000000000000-0000000000000000000000-000000')
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 IBM i üzerinde işlev çağrıları

IBM i programlamada kullanılabilen işlev çağrıları hakkında bilgi edinmek için bu bilgileri kullanın.

IBM i üzerindeki arama açıklamalarında kullanılan kurallar

Her çağrı için, bu konular derlemi, çağrıya ilişkin parametrelerin ve kullanımın açıklamasını verir. Bunu, RPG programlama dilinde çağrının tipik çağrıları ve parametrelerinin tipik bildirimleri izler.

Önemli: IBM MQ API çağrılarını kodlarken, ilgili tüm parametrelerin (aşağıdaki bölümlerde açıklandığı gibi) sağlandığından emin olmanız gerekir. Bunun yapılmaması öngörülemez sonuçlar üretebilir.

Her aramanın açıklaması aşağıdaki bölümleri içerir:

Arama adı

Arama adının ardından, aramanın amacına ilişkin kısa bir açıklama gelir.

Parametreler

Her değiştirge için, adı parantez () içindeki veri tipi izler. ve yönü; örneğin:

CMPCOD (9 basamaklı ondalık tamsayı)-çıkış

“Temel veri tipleri” sayfa 982’inde yapı veri tipleriyle ilgili daha fazla bilgi vardır.

Parametrenin yönü şu olabilir:

Giriş

Bu parametreyi sizin (programcı) sağlamanız gerekir.

Çıktı

Çağrı bu parametreyi döndürür.

Giriş/çıkış

Bu parametreyi sağlamanız gerekir, ancak çağrı tarafından değiştirilir.

Parametrenin amacına ilişkin kısa bir açıklama ve parametrenin alabileceği değerlerin bir listesi de vardır.

Her çağrıdaki son iki parametre bir tamamlanma kodu ve bir neden kodudur. Tamamlanma kodu, aramanın başarıyla mı, kısmen mi, yoksa hiç mi tamamlandığını gösterir. Neden kodunda kısmi başarı ya da aramanın başarısız olmasıyla ilgili daha fazla bilgi verilmiştir.

Kullanım notları

Aramayı nasıl kullanacağınızı ve kullanımıyla ilgili kısıtlamaları açıklayan ek bilgiler.

RPG çağırma

RPG ' de çağrı ve parametrelerinin bildirilmesi için tipik çağrı.

Diğer kavramsal kurallar şunlardır:

Değişmezler

Değişmezlerin adları büyük harfle gösterilir; örneğin, OOOO.

Diziler

Bazı çağrılarda, değiştirgeler değişmez büyüklükte olmayan karakter dizilerinden oluşan dizilerdir. Bu parametrelerin açıklamalarında, küçük harfli n bir sayısal sabiti temsil eder. Bu değiştirgeye ilişkin bildirim kodladığınızda, n değerini, gerek duyduğunuz sayısal değerle değiştirin.

IBM i IBM i üzerinde MQBACK (Geri dönüş değişiklikleri)

MQBACK çağrısı kuyruk yöneticisine, son eşitleme noktasından bu yana oluşan tüm iletilerin ve iletilerin geriletildiğini gösterir. Bir iş biriminin parçası olarak konan iletiler silinir; iş biriminin bir parçası olarak alınan iletiler kuyrukta yeniden kullanılır.

- Bu çağrı aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Windows

- [“Sözdizimi” sayfa 1231](#)
- [“Kullanım notları” sayfa 1231](#)
- [“Parametreler” sayfa 1232](#)
- [“RPG Bildirimi” sayfa 1233](#)

Sözdizimi

MQBACK (*Hconn, CompCode, Reason*)

Kullanım notları

MQBACK kullanırken bu kullanım notlarını göz önünde bulundurun.

1. Bu çağrı yalnızca kuyruk yöneticisinin kendisi iş birimini koordine ettiğinde kullanılabilir. Bu, değişikliklerin yalnızca IBM MQ kaynaklarını etkilediği yerel bir iş birimidir.
2. Kuyruk yöneticisinin iş birimini koordine etmediği ortamlarda, MQBACK yerine uygun geri alma çağrısı kullanılmalıdır. Ortam, uygulamanın olağandışı sonlanmasından kaynaklanan örtük bir geri çekmeyi de destekleyebilir.
 - IBM i' de bu çağrı, kuyruk yöneticisi tarafından eşgüdümlü yerel iş birimleri için kullanılabilir. Bu, iş düzeyinde bir kesinleştirme tanımının var olmaması gerektiği, yani iş için **CMTSCOPE (*JOB)** parametresini içeren STRCMTCTL komutunun yayınlanmamış olması gerektiği anlamına gelir.
3. Bir uygulama, bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması uygulamanın olağan şekilde mi, yoksa olağandışı olarak mı bittiğine bağlıdır. Daha fazla ayrıntı için [“IBM i üzerinde MQDISC \(Kuyruk yöneticisinin bağlantısını kes\)” sayfa 1268](#) içindeki kullanım notlarına bakın.
4. Bir uygulama, mantıksal ileti gruplarına ya da bölümlerine ileti girdiğinde ya da aldığı anda, kuyruk yöneticisi, son başarılı MQPUT ve MQGET çağrılarında ilişkin ileti grubu ve mantıksal iletiyle ilgili bilgileri saklar. Bu bilgiler kuyruk tanıtıcısıyla ilişkilidir ve aşağıdaki gibi şeyleri içerir:
 - MQMD ' deki *MDGID, MDSEQ, MDOFF* ve *MDMFL* alanlarının değerleri.
 - İletinin bir iş biriminin parçası olup olmadığı.

- MQPUT çağrısı için: İletinin kalıcı mı, kalıcı mı olduğu.

Kuyruk yöneticisi, aşağıdakilerin her biri için bir küme olmak üzere üç grup ve bölüm bilgisi kümesini tutar:

- Son başarılı MQPUT çağrısı (bu, bir iş biriminin parçası olabilir).
- Kuyruktan bir iletiyi kaldıran son başarılı MQGET çağrısı (bu, bir iş biriminin parçası olabilir).
- Kuyruktaki bir iletiye göz atan son başarılı MQGET çağrısı (bu, bir iş biriminin parçası olamaz).

Uygulama iletileri bir iş biriminin parçası olarak yerleştirir ya da alırsa ve uygulama iş birimini geri çevirmeye karar verirse, grup ve bölüm bilgileri daha önce sahip olduğu değere geri yüklenir:

- MQPUT çağrısıyla ilişkili bilgiler, yürürlükteki iş biriminde o kuyruk tanıtıcısı için başarılı olan ilk MQPUT çağrısından önce sahip olduğu değere geri yüklenir.
- MQGET çağrısıyla ilişkili bilgiler, yürürlükteki iş biriminde o kuyruk tanıtıcısı için ilk başarılı MQGET çağrısından önce sahip olduğu değere geri yüklenir.

İş birimi başlatıldıktan sonra uygulama tarafından güncellenen, ancak iş birimi kapsamı dışında kalan kuyruklar, iş birimi geri çekilirse, grup ve bölüm bilgilerini geri yüklenmez.

Bir iş birimi geriletildiğinde grup ve bölüm bilgilerinin önceki değerine geri yüklenmesi, uygulamanın çeşitli iş birimlerine birçok kesim içeren büyük bir ileti grubu ya da büyük bir mantıksal ileti yaymasına ve iş birimlerinden birinin başarısız olması durumunda ileti grubunda ya da mantıksal iletide doğru noktada yeniden başlamasına olanak sağlar. Yerel kuyruk yöneticisinde yalnızca sınırlı kuyruk saklama alanı varsa, birkaç iş birimi kullanılması avantajlı olabilir. Ancak, bir sistem arızası ortaya çıkarsa, uygulamanın iletileri doğru noktada yeniden koymayı ya da almayı yeniden başlatabilmesi için yeterli bilgiyi koruması gerekir. Bir sistem hatasından sonra doğru noktada nasıl yeniden başlatılacağına ilişkin ayrıntılar için [“IBM i üzerinde MQPMO \(Put-message options\)” sayfa 1154](#) içinde açıklanan PMLOGO seçeneğine ve [“IBM i üzerinde MQGMO \(Get-message seçenekleri\)” sayfa 1060](#) içinde açıklanan GMLOGO seçeneğine bakın.

Geri kalan kullanım notları yalnızca kuyruk yöneticisi iş birimlerini koordine ettiğinde geçerlidir:

1. Bir iş birimi, bağlantı tanıtıcısıyla aynı kapsama sahiptir. Bu, belirli bir iş birimini etkileyen tüm IBM MQ çağrılarının aynı bağlantı tanıtıcısı kullanılarak gerçekleştirilmesi gerektiği anlamına gelir. Farklı bir bağlantı tanıtıcısı kullanılarak verilen çağrılar (örneğin, başka bir uygulama tarafından verilen çağrılar) farklı bir iş birimini etkiler. Bağlantı tanıtıcılarının kapsamıyla ilgili bilgi için [“IBM i üzerinde MQCONN \(Connect kuyruk yöneticisi\)” sayfa 1255](#) içinde açıklanan **HCONN** değiştirgesine bakın.
2. Yalnızca yürürlükteki iş biriminin bir parçası olarak konan ya da alınan iletiler bu çağrıdan etkilenir.
3. Bir iş birimi içinde MQGET, MQPUT ya da MQPUT1 çağrılarını veren, ancak hiçbir zaman kesinleştirme ya da geriletme çağrısı yapmayan uzun süreli bir uygulama, kuyrukların diğer uygulamalar tarafından kullanılmayan iletilerle dolmasına neden olabilir. Bu olasılığa karşı koruma sağlamak için, denetimcinin **MaxUncommittedMsgs** kuyruk yöneticisi özneteliğini, kuyrukları dolduran kaçak uygulamaları önleyecek kadar düşük, ancak beklenen ileti sistemi uygulamalarının düzgün çalışmasını sağlayacak kadar yüksek bir değere ayarlaması gerekir.

Parametreler

MQBACK çağrısındaki değiştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *HCONN* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *COMCOD*.

COMCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

COMCOD CCFAIL ise:

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2101

(2101, X'835 ') Nesne zarar gördü.

RC2123

(2123, X'84B') Kesinleştirme ya da geri yükleme işleminin sonucu karışık.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQBACK(HCONN : COMCOD : REASON)
```


Çağrıya ilişkin prototip tanımlaması:

```
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 (İş birimini başlat)- IBM i**

MQBEGIN çağrısı, kuyruk yöneticisi tarafından koordine edilen ve dış kaynak yöneticilerini içerebilecek bir iş birimi başlatır.

- Bu çağrı aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Windows

- [“Sözdizimi” sayfa 1234](#)
- [“Kullanım notları” sayfa 1234](#)
- [“Parametreler” sayfa 1235](#)
- [“RPG Bildirimi” sayfa 1236](#)

Sözdizimi

MQBEGIN (*HCONN, BEGOP, CMPCOD, REASON*)

Kullanım notları

1. MQBEGIN çağrısı, kuyruk yöneticisi tarafından eşgüdümlü bir iş birimini başlatmak için kullanılabilir ve bu, diğer kaynak yöneticilerinin sahip olduğu kaynaklarda değişiklikler içerebilir. Kuyruk yöneticisi üç tip iş birimini destekler:

Kuyruk yöneticisi-eşgüdümlü yerel iş birimi

Bu, kuyruk yöneticisinin katılan tek kaynak yöneticisi olduğu ve kuyruk yöneticisinin iş birimi eşgüdücüsü olarak hareket ettiği bir iş birimidir.

- Bu iş birimi tipini başlatmak için, iş birimindeki ilk MQPUT, MQPUT1 ya da MQGET çağrısında PMSYP ya da GMSYP seçeneği belirtilmelidir.

Uygulamanın iş birimini başlatmak için MQBEGIN çağrısının yayınlaması gerekmez, ancak MQBEGIN kullanılırsa, çağrı CCWARN ve RC2121 neden koduyla tamamlanır.

- Bu iş birimi tipini kesinleştirmek ya da geri almak için MQCMIT ya da MQBACK çağrısı kullanılmalıdır.

Kuyruk yöneticisi-eşgüdümlü genel iş birimi

Bu, kuyruk yöneticisinin hem IBM MQ kaynaklar için, hem de diğer kaynak yöneticilerine ait kaynaklar için iş birimi eşgüdücüsü olarak davrandığı bir iş birimidir. Bu kaynak yöneticileri, iş birimindeki kaynaklarda yapılan tüm değişikliklerin kesinleştirilmesini ya da birlikte geriletilmesini sağlamak için kuyruk yöneticisiyle işbirliği yapar.

- Bu iş birimi tipini başlatmak için MQBEGIN çağrısı kullanılmalıdır.
- Bu iş birimini kesinleştirmek ya da geri almak için MQCMIT ve MQBACK çağrıları kullanılmalıdır.

Harici eşgüdümlü genel iş birimi

Bu, kuyruk yöneticisinin katılımcı olduğu, ancak kuyruk yöneticisinin iş birimi eşgüdücüsü olarak işlev görmediği bir iş birimidir. Bunun yerine, kuyruk yöneticisinin birlikte çalıştığı bir dış iş birimi koordinatörü vardır.

- Bu iş birimi tipini başlatmak için, dış iş birimi koordinatörü tarafından sağlanan ilgili çağrı kullanılmalıdır.

İş birimini başlatmayı denemek için MQBEGIN çağrısı kullanılırsa, çağrı RC2012 neden koduyla başarısız olur.

- Bu iş birimi tipini kesinleştirmek ya da geri almak için, dış iş birimi eşgüdücüsü tarafından sağlanan kesinleştirme ve gerileme çağrıları kullanılmalıdır.

İş birimini kesinleştirmek ya da geri almak için MQCMIT ya da MQBACK çağrısı kullanılırsa, çağrı RC2012 neden koduyla başarısız olur.

2. Uygulama, bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması, uygulamanın olağan şekilde mi, yoksa olağandışı olarak mı bittiğine bağlıdır. Daha fazla ayrıntı için

“IBM i üzerinde MQDISC (Kuyruk yöneticisinin bağlantısını kes)” sayfa 1268 içindeki kullanım notlarına bakın.

3. Bir uygulama aynı anda yalnızca bir iş birimine katılabilir. MQBEGIN çağrısı RC2128 neden koduyla başarısız oluyor; uygulama için var olan bir iş birimi varsa, hangi iş birimi olduğundan bağımsız olarak.
4. MQBEGIN çağrısı, IBM MQ istemci ortamında geçerli değil. Aramayı kullanma girişimi RC2012neden koduyla başarısız oldu.
5. Kuyruk yöneticisi genel iş birimleri için iş birimi eşgüdümçüsü olarak hareket ederken, iş birimine katılabilecek kaynak yöneticileri kuyruk yöneticisinin yapılandırma dosyasında tanımlanır.
6. IBM işletim sisteminde, üç iş birimi tipi aşağıdaki gibi desteklenir:
 - **Kuyruk yöneticisi eşgüdümlü yerel iş birimleri** yalnızca iş düzeyinde bir kesinleştirme tanımı yoksa, yani iş için **CMTSCOPE (*JOB)** parametresiyle birlikte STRCMTCTL komutu verilmediğinde kullanılabilir.
 - **Kuyruk yöneticisi-eşgüdümlü genel iş birimleri** desteklenmez.
 - **Harici eşgüdümlü genel iş birimleri** yalnızca iş düzeyinde bir kesinleştirme tanımı varsa, yani iş için **CMTSCOPE (*JOB)** parametresiyle STRCMTCTL komutu verildiğinde kullanılabilir. Bu yapıldıysa, IBM i COMMIT ve ROLLBACK işlemleri, IBM MQ kaynaklarının yanı sıra, katılan diğer kaynak yöneticilerine ait kaynaklar için de geçerlidir.

Parametreler

MQBEGIN çağrısı aşağıdaki parametreleri içeriyor:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *HCONN* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

BEGOP (MQBO)-giriş/çıkış

MQBEGIN işlemini denetleyen seçenekler.

Ayrıntılar için bkz. “IBM i üzerinde MQBO (Başlangıç seçenekleri)” sayfa 1003.

Herhangi bir seçenek gerekmiyorsa, C ya da S/390 çevirici dilinde yazılan programlar, bir MQBO yapısının adresini belirtmek yerine boş bir değiştirge adresi belirtebilir.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2121

(2121, X'849 ') Katılımcı kaynak yöneticisi kaydedilmedi.

RC2122

(2122, X'84A') Kaynak yöneticisine katılım sağlanmıyor.

CMPCOD CCFAIL ise:

RC2134

(2134, X'856 ') Başlangıç seçenekleri yapısı geçersiz.

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2012

(2012, X'7DC') Çağrısı ortamda geçerli değil.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2046

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RC2128

(2128, X'850 ') İş birimi zaten başlatıldı.

RPG Bildirimi

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQBEGIN(HCONN : BEGOP : CMPCOD :
C                                REASON)

```

Çağrıya ilişkin prototip tanımlaması:

```

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

```

IBM i IBM i üzerinde MQBUFMMH (Arabelleği ileti tanıtıcısına dönüştür)

MQBUFMMH işlev çağrısı, bir arabelleği ileti tanıtıcısına dönüştürür ve MQMHBUF çağrısının tersidir.

Bu çağrı, arabellekte bir ileti tanımlayıcısı ve MQRFH2 özelliklerini alır ve bunları bir ileti tanıtıcısı aracılığıyla kullanılabilir kılar. İleti verilerindeki MQRFH2 özellikleri isteğe bağlı olarak kaldırılır. Gerekliyse, ileti tanımlayıcısının *Encoding, CodedCharSetIdve Format* alanları, özellikler kaldırıldıktan sonra arabelleğin içeriğini doğru şekilde tanımlamak için güncellenir.

- [“Sözdizimi” sayfa 1237](#)
- [“Kullanım notları” sayfa 1237](#)
- [“Parametreler” sayfa 1237](#)
- [“RPG Bildirimi” sayfa 1239](#)

Sözdizimi

MQBUFMH (*Hconn, Hmsg, BufMsgHOpts, MsgDesc, Buffer, BufferLength, DataLength, CompCode, Reason*)

Kullanım notları

MQBUFMH çağrıları API çıkışları tarafından engellenemez-bir arabellek uygulama alanında ileti tanıtıcısına dönüştürülür; çağrı kuyruk yöneticisine ulaşmaz.

Parametreler

MQBUFMH çağrısındaki değıştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *HCONN* değeri, **Hmsg** değıştirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eşleşmelidir.

İleti tanıtıcısı HCUNAS kullanılarak yaratıldıysa, arabelleği ileti tanıtıcısına dönüştüren iş parçacığında geçerli bir bağlantı kurulmalıdır. Geçerli bir bağlantı kurulmazsa, arama RC2009 ile başarısız olur.

HMSG (20 basamaklı işaretli tamsayı)-giriş

Bu tanıtıcı, arabellek gerektiren ileti tanıtıcısıdır. Değer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

BMHOPT (MQBMHO)-GİRİŞ

MQBMHO yapısı, uygulamaların, ileti tanıtıcılarının arabelleklerden nasıl üretildiğini denetleyen seçenekleri belirtmesine olanak sağlar.

Ayrıntılar için bkz. [“IBM i üzerinde MQBMHO \(Arabellek-ileti işleme seçenekleri\)” sayfa 1002.](#)

MSGDSC (MQMD)-giriş/çıkış

MSGDSC yapısı, ileti tanımlayıcı özelliklerini içerir ve arabellek alanının içeriğini açıklar.

Çağrı çıkışında, özellikler isteğe bağlı olarak arabellek alanından kaldırılır ve bu durumda, ileti tanımlayıcısı arabellek alanını doğru şekilde tanımlayacak şekilde güncellenir.

Bu yapıdaki veriler, uygulamanın karakter kümesi ve kodlamasında olmalıdır.

BUFLEN (10 basamaklı işaretli tamsayı)-giriş

BUFLEN, Arabellek alanının bayt cinsinden uzunluğudur.

Sıfır baytlık bir *BUFLEN* geçerli ve arabellek alanının veri içermediğini gösterir.

BUFFER (1 baytlık bit dizilimi x BUFLEN)-giriş/çıkış

BUFFER, ileti arabelleğini içeren alanı tanımlar. Çoğu veri için, arabelleği 4 baytlık bir sınırdan hizalayın.

BUFFER karakter ya da sayısal veri içeriyorsa, **MSGDSC** parametresindeki *CodedCharSetId* ve *Encoding* alanlarını veriler için uygun değerlere ayarlayın; bu, gerekirse verilerin dönüştürülmesini sağlar.

İleti arabelleğinde özellikler bulunursa, isteğe bağlı olarak bunlar kaldırılır; bunlar daha sonra çağrıdan döndüklerinde ileti tanıtıcısından kullanılabilir olur.

C programlama dilinde parametre, herhangi bir veri tipinin adresinin parametre olarak belirtileceği anlamına gelen bir işaretçi-to-void olarak bildirilir.

BUFLEN parametresi sıfırda, **BUFFER** parametresine başvuruda bulunulmaz. Bu durumda, C ya da System/390 çevirici dilinde yazılmış programlardan geçirilen parametre adresi boş olabilir.

DATLEN (10 basamaklı işaretli tamsayı)-çıkış

DATLEN, özelliklerin kaldırılabilceği arabelleğin bayt cinsinden uzunluğudur.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

CCOK

Başarıyla tamamlandı.

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCFail ise:

RC2204

(2204, X'089C') Bağdaştırıcı kullanılamıyor.

RC2130

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

RC2157

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

RC2489

(2489, X'09B9') İleti işleme seçeneklerine ilişkin arabellek yapısı geçersiz.

RC2004

(2004, X'07D4') Arabellek değiştirgesi geçersiz.

RC2005

(2005, X'07D5') Arabellek uzunluğu değiştirgesi geçersiz.

RC2219

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

RC2009

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2460

(2460, X'099C') İleti tanıtıcısı geçersiz.

RC2026

(2026, X'07EA') İleti tanımlayıcısı geçersiz.

RC2499

(2499, X'09C3') İleti tanıtıcısı kullanımda.

RC2046

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2334

(2334, X'091E') MQRFH2 yapısı geçerli değil.

RC2421

(2421, X'0975 ') Özellikleri içeren bir MQRFH2 klasörü ayrıştırılmadı.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQBUFMH(HCONN : HMSG : BMHOPT :
                          MSGDSC : BUFLN : BUFFER :
                          DATLEN : CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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 BUFLN           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

IBM i üzerinde MQCB (Geri çağırmaı yönet)

MQCB çağırısı, belirtilen nesne tanıtıcısı için bir geri çağırmaı yeniden çağırır ve etkinleştirmeyi ve geri çağırma değışikliklerini denetler.

Geri çağırma, belirli olaylar oluştüğında IBM MQ tarafından çağırılan bir kod parçasıdır (devingen olarak bağlanabilen bir işlevin adı olarak ya da bir işlev göstergesi olarak belirtilir).

Bir V7 istemcisinde MQCB ve MQCTL kullanabilmek için bir V7 sunucusuna bağlanmanız ve kanalın **SHARECNV** parametresinin sıfır dışında bir değeri olması gerekir.

Genel iş birimleri hakkında bilgi için bkz. [Genel iş birimleri](#).

Tanımlanabilecek geri çağırma tipleri şunlardır:

İleti tüketicisi

Belirtilen seçim ölçütlerini karşılayan bir ileti, bir nesne tanıtıcısında kullanılabilir olduğunda, ileti tüketicisi geri çağırma işlevi çağırılır.

Her nesne tanıtıcısı için tek bir geri çağırma işlevi kaydedilebilir. Tek bir kuyruk birden çok seçim ölçütüyle okunacaksa, kuyruk birden çok kez açılmalı ve her bir tanıtıcıda bir tüketicisi işlevi kaydedilmelidir.

Olay işleyici

Olay işleyici, tüm geri çağırma ortamını etkileyen koşullar için çağırılır.

Bir olay koşulu (örneğin, bir kuyruk yöneticisi ya da bağlantı durdurma ya da susturma) oluştüğında işlev çağırılır.

İşlev, tek bir ileti tüketicisine özgü koşullar için çağırılmaz; örneğin, RC2016; ; ancak, geri çağırma işlevi olağan şekilde sona ermezse çağırılır.

- [“Sözdizimi” sayfa 1240](#)
- [“MQCB için kullanım notları” sayfa 1240](#)
- [“MQCB için değıştirgeler” sayfa 1241](#)
- [“RPG Bildirimi” sayfa 1247](#)

Sözdizimi

MQCB (HCONN, OPERATN, HOBJ, CBDSC, MSGDSC, GMO, CMPCOD, REASON)

MQCB için kullanım notları

1. MQCB, kuyrukta var olan belirtilen ölçütlerle eşleşen her ileti için çağrılacak işlemi tanımlamak için kullanılır. İşlem işlendiğinde, ileti kuyruktan kaldırılır ve tanımlı ileti tüketicisine iletilir ya da iletiyi almak için kullanılan bir ileti simgesi sağlanır.
2. MQCB, MQCTL ile tüketime başlamadan önce geri çağırma yordamlarını tanımlamak için kullanılabilir ya da bir geri çağırma yordamından kullanılabilir.
3. MQCB ' yi bir geri çağırma yordamının dışından kullanmak için, önce MQCTL kullanarak ileti tüketimini askıya almanız ve daha sonra tüketimi sürdürmeniz gerekir.

İleti tüketici geri çağırma sırası

Bir tüketiciyi, tüketicinin yaşam çevrimi boyunca temel noktalarda geri çağırma üzere yapılandırabilirsiniz. Örneğin:

- Tüketici ilk kez kaydedildiğinde,
- bağlantı başlatıldığında,
- bağlantı durdurulduğunda ve
- Tüketici bir MQCLOSE tarafından belirtik ya da örtük olarak kayıttan kaldırıldığında.

Çizelge 742. MQCTL fiil tanımlamaları	
Komut	Anlamı
MQCTL (BAŞLAT)	CTLSR İşlemi kullanılarak MQCTL çağırısı
MQCTL (DURDUR)	CTLSP İşlemi kullanılarak MQCTL çağırısı
MQCTL (BEKLE)	CSLSW işlemi kullanarak MQCTL çağırısı

Tüketicinin, tüketiciyle ilişkili durumu korumasını sağlar. Bir uygulama tarafından geri çağırma istendiğinde, tüketici çağırma kuralları aşağıdaki gibidir:

KAYDOL

Her zaman geri çağırmanın ilk tipidir.

MQCB (CBREG) çağırısıyla aynı iş parçacığında her zaman çağrılır.

START

MQCTL (START) komutu ile her zaman zamanuyumlu olarak çağrılır.

- Tüm START geri çağırımlar, MQCTL (START) komutu döndürülmeden önce tamamlanır.

CTLTHR istenirse, ileti teslimiyle aynı iş parçacığında.

Örneğin, MQCTL (START) sırasında önceki bir geri çağırma MQCTL (STOP) komutu verirse, başlatma ile çağrı garanti edilmez.

DUR

Bağlantı yeniden başlatılıncaya kadar, bu çağrıdan sonra başka ileti ya da olay teslim edilmez.

Uygulama daha önce START ya da ileti ya da olay için çağrıldıysa, STOP garanti edilir.

DEREGISTER

Her zaman geri çağırmanın son türüdür.

Uygulamanızın START ve STOP geri çağırılarında iş parçacığı tabanlı başlatma ve temizleme gerçekleştirdiğinden emin olun. REGISTER ve DEREGISTER geri çağırılıyla iş parçacığı tabanlı olmayan kullanıma hazırlama ve temizleme yapabilirsiniz.

Belirtilen iş parçacığının kullanım ömrü ve kullanılabilirliği hakkında herhangi bir varsayımda bulunmayın. Örneğin, DEREGISTER ' e son çağrıdan sonra canlı kalan bir iş parçacığına güvenmeyin.

Benzer şekilde, CTLTHR ' yi kullanmamayı seçtiğinizde, bağlantı her başlatıldığında iş parçacığının var olduğunu varsaymayın.

Uygulamanızın iş parçacığı özelliklerine ilişkin belirli gereksinimleri varsa, her zaman buna göre bir iş parçacığı yaratabilir ve MQCTL (WAIT) kullanabilir. Bu adım, zamanuyumsuz ileti teslimi için *bağışlar* iş parçacığını IBM MQ ' e gönderir.

İleti tüketici bağlantısı kullanımı

Olağan durumda, bir uygulama henüz tamamlanmamışken başka bir MQI çağrısı yayınladığında, çağrı RC2219neden koduyla başarısız olur.

Ancak, uygulamanın önceki çağrı tamamlanmadan önce başka bir MQI çağrısı yürütmesi gereken özel durumlar vardır. Örneğin, tüketici CBRE ile MQCB çağrısı sırasında çağrılabilir.

Böyle bir durumda, uygulamanın MQCB ya da MQCTL komutu vermesinin sonucu olarak uygulama geri çağrıldığında, uygulamanın başka bir MQI çağrısı yapmasına izin verilir. Bu yönetim ortamı, CBCTRC CBCCALLT tipiyle çağrıldığında tüketici işlevinde MQOPEN çağrısı gibi bir ileti yayınlayabileceğiniz anlamına gelir. MQDISC dışında MQI çağrılarında izin verilir.

MQCB için deęiřtirgeler

MQCB çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı iřaretli tamsayı)-giriř

Geri çağırma iřlevini yönet-HCONN parametresi.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *HCONN* deęeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

OPERATN (10 basamaklı iřaretli tamsayı)-giriř

Geri çağırma iřlevini yönet-OPERATN parametresi.

Belirtilen nesne tanıtıcısı için tanımlanan geri çağırma sırasında iřlenmekte olan iřlem. Ařaęıdaki seęeneklerden birini belirtmeniz gerekir; birden çok seęenek gerekliyse, deęerler eklenebilir (aynı sabiti bir kereden fazla eklemeyin) ya da bitsel OR iřlemi kullanılarak birleřtirilebilir (programlama dili bit iřlemlerini destekliyorsa).

Geęerli olmayan birleřimler belirtilir; dięer tüm birleřimler geęerlidir.

CBREG

Belirtilen nesne tanıtıcısı için geri çağırma iřlevini tanımlayın. Bu iřlem, çağrılacak iřlevi ve kullanılacak seęim ölçütlerini tanımlar.

Nesne tanıtıcısı için önceden tanımlanmış bir geri çağırma iřlevi varsa, tanımlama deęiřtirilir. Geri çağırma deęiřtirilirken bir hata saptanırsa, iřlevin kaydı kaldırılır.

Bir geri çağırma daha önce kayıttan kaldırıldıęı aynı geri çağırma iřlevine kaydedilirse, bu bir yerine koyma iřlemi olarak iřlenir; ilk ya da son çağrılar çağrılmaz.

CSLSU ya da CTLRE ile CBREG kullanabilirsiniz.

CBI

Nesne tanıtıcısı için ileti tüketimini durdurun ve geri çağırma için seęilenlerden tutamacı kaldırır.

İliřkili tanıtıcı kapatılırsa, bir geri çağırmanın kaydı otomatik olarak kaldırılır.

CBUNR bir tüketici içinden çağrılırsa ve geri çağırmanın tanımlı bir durdurma çağrısı varsa, bu çağrı tüketicinin dönüşünden sonra çağrılır.

Bu iřlem, kayıtlı tüketicisi olmayan bir *Hobj* için yayınlandıysa, çağrı RC2448ile geri döner.

CTLSU

Nesne tanıtıcısı için ileti tüketimini askıya alır.

Bu iřlem bir olay iřleyiciye uygulanırsa, olay iřleyici askıya alınmış durumdayken olayları alamaz ve askıya alınmış durumdayken kaçırılan olaylar, sürdürüldüęünde iřleme saęlanmaz.

Askıya alındığında, tüketici işlevi denetim tipi geri çağırımları almaya devam eder.

CTLRE

Nesne tanıtıcısı için iletileri tüketmeye devam edin.

Bu işlem bir olay işleyiciye uygulanırsa, olay işleyici askıya alınmış durumdayken olayları alamaz ve askıya alınmış durumdayken kaçırılan olaylar, sürdürüldüğünde işleme sağlanmaz.

CBDSC (MQCBD)-giriş

Geri çağırma işlevini yönet-CBDSC parametresi.

Bu, uygulama tarafından kaydedilmekte olan geri çağırma işlevini ve kayıt sırasında kullanılan seçenekleri tanımlayan bir yapıdır.

Yapının ayrıntıları için bkz. "MQCBD-Geri çağırma tanımlayıcısı" sayfa 289 .

Geri çağırma tanımlayıcısı yalnızca CBREG seçeneği için gereklidir; tanımlayıcı gerekli değilse, geçirilen parametre adresi boş değerli olabilir.

HOBJ (10 basamaklı işaretli tamsayı)-giriş

Geri çağırma işlevini yönet-HOBJ parametresi.

Bu tanıtıcı, iletinin tüketileceği nesne için oluşturulan erişimi gösterir. Bu, önceki bir MQOPEN ya da MQSUB çağrısından (**HOBJ** değiştirilmesinde) döndürülen bir tanıtıcıdır.

Bir olay işleyici yordamı (CBTEH) tanımlanırken **HOBJ** gerekli değildir ve HONONE olarak belirtilmelidir.

Bu *Hobj* bir MQOPEN çağrısından döndürüldüyse, kuyruk aşağıdaki seçeneklerden biriyle ya da daha fazlasıyla açılmış olmalıdır:

- OOINPS
- OOINPX
- OOINPQ
- OOBROW

MSGDSC (MQMD)-giriş

Geri çağırma işlevini -MSGDSC parametresini yönetin.

Bu yapı, gerekli iletinin özniteliklerini ve alınan iletinin özniteliklerini açıklar.

MsgDesc değiştirgesi, tüketicinin gerektirdiği iletilerin özniteliklerini ve ileti tüketicisine geçirilecek MQMD sürümünü tanımlar.

MQMD 'deki *MsgId*, *CorrelId*, *GroupId*, *MsgSeqNumber* ve *Offset* , **GetMsgOpts** değiştirilmesinde belirtilen seçeneklere bağlı olarak ileti seçimi için kullanılır.

GMCONV seçeneğini belirlerseniz, ileti dönüştürme için *Encoding* ve *CodedCharSetId* kullanılır.

Ayrıntılar için bkz. MQMD .

MsgDesc yalnızca CBREG için ve herhangi bir alan için varsayılan dışında değerler gerekiyorsa kullanılır. *MsgDesc* bir olay işleyici için kullanılmaz.

Tanımlayıcı gerekli değilse, geçirilen parametre adresi boş değerli olabilir.

Çakışan seçicilerle aynı kuyruğa birden çok tüketici kaydedilirse, her ileti için seçilen tüketicinin tanımlanmamış olduğunu unutmayın.

GMO (MQGMO)-giriş

Geri çağırma işlevini yönet-GMO parametresi.

İleti tüketicisinin iletileri nasıl aldığı denetleyen seçenekler.

Tüm seçenekler, bir MQGET çağrısında kullanıldığında "IBM i üzerinde MQGMO (Get-message seçenekleri)" sayfa 1060 içinde açıklandığı gibi anlamlara gelir, ancak:

GMSSIG

Bu seçeneğine izin verilmez.

GMBRWF, GMBRWN, GMMBH, GMMBC

Bir göz atma tüketicisine gönderilen iletilerin sırası, bu seçeneklerin birleşimleri tarafından belirlenmiştir. Önemli birleşimler şunlardır:

GMBRWF

Kuyruktaki ilk ileti tekrar tekrar tüketiciye teslim edilir. Bu, tüketici geri çağırma iletiyi yıkıcı bir şekilde tükettiğinde kullanışlıdır. Bu seçeneği dikkatli kullanın.

GMBRWN

Yürürlükteki imleç konumundan kuyruğun sonuna ulaşıncaya kadar, kuyruktaki her ileti tüketiciye verilir.

GMBRWF + GMBRWN

İmleç, kuyruğun başlangıcını belirtir. Daha sonra, imleç kuyruğun sonuna ulaşıncaya kadar her ileti tüketiciye verilir.

GMBRWF + GMMBH ya da GMMBC

Kuyruğun başından başlayarak, tüketiciye kuyrukta ilk işaretlenmemiş ileti verilir; bu ileti daha sonra bu tüketici için işaretlenir. Bu birleşim, tüketicinin geçerli imleç noktasının arkasına eklenen yeni iletileri alabilmesini sağlar.

GMBRWN + GMMBH ya da GMMBC

İmleç konumundan başlayarak, tüketiciye kuyrukta imlenmemiş bir sonraki ileti verilir ve bu ileti bu tüketici için işaretlenir. İletiler geçerli imleç konumunun arkasındaki kuyruğa eklenebildiği için bu bileşimi dikkatli bir şekilde kullanın.

GMBRWF + GMBRWN + GMMBH ya da GMMBC

Bu birleşime izin verilmez, arama kullanılırsa RC2046değerini döndürür.

GMNWT, GMWT ve GMWI

Bu seçenekler, tüketicinin nasıl çağrılacağını denetler.

GMNWT

Tüketici hiçbir zaman RC2033ile çağrılmaz. Tüketici yalnızca iletiler ve olaylar için çağrılır

Sıfır GMWI ile GMWT

RC2033 kodu yalnızca ileti olmadığında ve

- tüketici başlatıldı
- Son ileti yok neden kodundan bu yana tüketici en az bir ileti teslim etti.

Bu, sıfır bekleme aralığı belirtildiğinde tüketicinin meşgul döngüde yoklama yapmasını önler.

GMWT ve pozitif bir GMWI

Kullanıcı, belirtilen bekleme aralığından sonra RC2033neden koduyla çağrılır. Bu çağrı, tüketiciye iletilerin teslim edilip edilmediğine bakılmaksızın yapılır. Bu, kullanıcının sağlıklı işletim bildirimini ya da toplu iş tipi işlemesi gerçekleştirmesini sağlar.

GMWT ve GMWI (WIULIM)

Bu, RC2033döndürülmeden önce sonsuz bir bekleme olduğunu belirtir. Tüketici hiçbir zaman RC2033ile çağrılmaz.

GMO yalnızca CBREG için ve herhangi bir alan için varsayılan dışında değerler gerekiyorsa kullanılır. *GMO* bir olay işleyici için kullanılmaz.

Seçenekler gerekli değilse, geçirilen parametre adresi boş değerli olabilir.

MQGMO yapısında bir ileti özellikleri tanıtıcısı sağlanırsa, MQGMO yapısında tüketici geri çağırmasına geçirilen bir kopya sağlanır. MQCB çağrısından döndüğünüzde, uygulama ileti özellikleri tanıtıcısını silebilir.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Geri çağırma işlevini yönet-CMPCOD parametresi.

Tamamlanma kodu; aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Geri çağırma işlevini yönet-REASON parametresi.

Aşağıdaki neden kodları, kuyruk yöneticisinin **REASON** parametresi için döndürebileceği kodlardır.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode CCFAIL ise:

RC2204

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

RC2133

(2133, X'855 ') Veri dönüştürme hizmetleri modülleri yüklenemiyor.

RC2130

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

RC2374

(2374, X' 946 ') API çıkışı başarısız oldu.

RC2183

(2183, X'887 ') API çıkışı yüklenemiyor.

RC2157

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

RC2005

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

RC2219

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

RC2487

(2487, X'9B7') Geri çağırma tipi alanı yanlış.

RC2448

(2448, X' 990 ') Kayıtlı geri çağırma olmadığı için kayıttan kaldırılamıyor, askıya alınamıyor ya da sürdürülemiyor.

RC2486

(2486, X'9B6') *CallbackFunction* ya da *CallbackName* belirtilmelidir, ancak her ikisi de belirtilmemelidir.

RC2483

(2483, X'9B3') Geri çağırma tipi alanı yanlış.

RC2484

(2484, X'9B4') Yanlış MQCBD seçenekleri alanı.

RC2140

(2140, X'85C') Bekleme isteği CICStarafından reddedildi.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2217

(2217, X'8A9') Bağlantı için yetkili değil.

- RC2202**
(2202, X'89A') Bağlantı susturuyor.
- RC2203**
(2203, X'89B') Bağlantı kapatılıyor.
- RC2207**
(2207, X'89F') İlinti-tanıtıcı hatası.
- RC2010**
(2010, X'7DA') Veri uzunluğu parametresi geçersiz.
- RC2016**
(2016, X'7E0') Kuyruk için engellendi.
- RC2351**
(2351, X'92F') Genel iş birimleri çakışması.
- RC2186**
(2186, X'88A') Get-message seçenekleri yapısı geçersiz.
- RC2353**
(2353, X' 931 ') Genel iş birimi için kullanımda.
- RC2018**
(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.
- RC2019**
(2019, X'7E3') Nesne tanıtıcısı geçerli değil.
- RC2259**
(2259, X'8D3') Tutarsız göz atma belirtimi.
- RC2245**
(2245, X'8C5') İş birimi belirtimi tutarsız.
- RC2246**
(2246, X'8C6') İmleç altındaki ileti alma için geçerli değil.
- RC2352**
(2352, X' 930 ') Genel iş birimi yerel iş birimiyle çakışıyor.
- RC2247**
(2247, X'8C7') Eşleştirme seçenekleri geçerli değil.
- RC2485**
(2485, X'9B4') *MaxMsgLength* alanı yanlış.
- RC2026**
(2026, X'7EA') İleti tanımlayıcısı geçersiz.
- RC2497**
(2497, X'9C1') Belirtilen işlev giriş noktası modülde bulunamadı.
- RC2496**
(2496, X'9C0') Modül bulundu, ancak yanlış tipte; 32 bit, 64 bit ya da geçerli bir dinamik bağlantı kitaplığı değil.
- RC2495**
(2495, X'9BF') Modül arama yolunda bulunamadı ya da yükleme yetkisi yok.
- RC2250**
(2250, X'8CA') İleti sıra numarası geçersiz.
- RC2331**
(2331, X'91B') İleti simgesi kullanımı geçersiz.
- RC2033**
(2033, X'7F1') İleti yok.
- RC2034**
(2034, X'7F2') Göz at imleci iletide konumlandırılmadı.

- RC2036**
(2036, X'7F4') Kuyruk göz atmak için açık değil.
- RC2037**
(2037, X'7F5') Kuyruk giriş için açık değil.
- RC2041**
(2041, X'7F9') Nesne tanımı açıldıktan sonra değiştirildi.
- RC2101**
(2101, X'835 ') Nesne zarar gördü.
- RC2206**
(2206, X'89E') API Çağrısı 'nda işlem kodu yanlış.
- RC2046**
(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.
- RC2193**
(2193, X'891 ') Sayfa kümesi veri kümesine erişilirken hata oluştu.
- RC2052**
(2052, X'804 ') Kuyruk silindi.
- RC2394**
(2394, X'95A') Kuyruk yanlış dizin tipine sahip.
- RC2058**
(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.
- RC2059**
(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.
- RC2161**
(2161, X'871 ') Kuyruk yöneticisi susturuyor.
- RC2162**
(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.
- RC2102**
(2102, X'836 ') Sistem kaynakları yetersiz.
- RC2069**
(2069, X'815 ') Bu tutamaç için bekleyen işaret.
- RC2071**
(2071, X'817 ') Kullanılabilir bellek yetersiz.
- RC2109**
(2109, X'83D') Çağrı çıkış programı tarafından engellendi.
- RC2024**
(2024, X'7E8') Yürürlükteki iş birimi içinde başka ileti işlenmez.
- RC2072**
(2072, X'818 ') Syncpoint desteği kullanılmıyor.
- RC2195**
(2195, X'893 ') Beklenmeyen hata oluştu.
- RC2354**
(2354, X' 932 ') Genel iş birimine kayıt başarısız oldu.
- RC2355**
(2355, X' 933 ') İş birimi çağrılarının karışımı desteklenmez.
- RC2255**
(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.
- RC2090**
(2090, X'82A') MQGMO' daki bekleme aralığı geçersiz.
- RC2256**
(2256, X'8D0') Yanlış MQGMO sürümü sağlandı.

RC2257

(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

RC2298

(2298, X'8FA') İstenen işlev yürürlükteki ortamda kullanılamıyor.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCB(HCONN : OPERATN : CBDSC :
                   HOBJ : MSGDSC : GMO :
                   DATLEN : CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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

IBM i üzerinde MQCLOSE (Nesneyi kapat)

MQCLOSE çağrısı bir nesneye erişimi geri alır ve MQOPEN çağrısının tersini oluşturur.

- “Sözdizimi” sayfa 1247
- “Kullanım notları” sayfa 1247
- “Parametreler” sayfa 1248
- “RPG Bildirimi” sayfa 1253

Sözdizimi

MQCLOSE (*HCONN*, *HOBJ*, *OPTS*, *CMPCOD*, *REASON*)

Kullanım notları

1. Bir uygulama MQDISC çağrısıyla ilgili bir sorun oluşturduğunda ya da olağan dışı ya da olağandışı bir şekilde sona erdiğinde, uygulama tarafından açılan ve hala açık olan nesnelere CONONE seçeneğiyle otomatik olarak kapanır.
2. Kapatılan nesne bir *kuyruk* ise aşağıdaki noktalar geçerlidir:
 - Kuyruktaki işlemler iş biriminin bir parçası olarak gerçekleştirilirse, kuyruk, eşitleme noktasının sonucunu etkilemeden, eşitleme noktasından önce ya da sonra kapatılabilir.
 - Kuyruk OOBROW seçeneğiyle açıldıysa, göz atma imleci yok edilir. Kuyruk daha sonra OOBROW seçeneğiyle yeniden açılırsa, yeni bir göz atma imleci yaratılır (MQOPEN içinde açıklanan OOBROW seçeneğine bakın).
 - MQCLOSE çağrısı sırasında bu tanıtıcı için bir ileti kilitliyse, kilit serbest bırakılır (“IBM i üzerinde MQGMO (Get-message seçenekleri)” sayfa 1060 içinde açıklanan GMLK seçeneğine bakın).

3. Kapatılan nesne bir *dinamik kuyruk* (kalıcı ya da geçici) ise aşağıdaki noktalar geçerlidir:

- Devingen bir kuyruk için, ilgili MQOPEN çağrısında belirtilen seçeneklerden bağımsız olarak CODEL ya da COPURG seçenekleri belirtilebilir.
- Dinamik bir kuyruk silindiğinde, GMWT seçeneğiyle, kuyrukta bekleyen tüm MQGET çağrıları iptal edilir ve RC2052 neden kodu döndürülür. "[IBM i üzerinde MQGMO \(Get-message seçenekleri\)](#)" sayfa [1060](#) içinde açıklanan GMWT seçeneğine bakın.

Dinamik bir kuyruk silindikten sonra, daha önce edinilmiş bir *HOBJ* tanıtıcısı kullanılarak kuyruğa başvurmayı deneyen herhangi bir çağrı (MQCLOSE dışında) RC2052neden koduyla başarısız olur.

Silinen bir kuyruğa uygulamalar tarafından erişilemese de, kuyruğa sistemden kaldırılamaz ve kuyruğa gönderme yapan tüm tanıtıcıları kapatıncaya ve kuyruğu etkileyen tüm iş birimleri kesinleştirilinceye ya da yedekleninceye kadar, ilişkili kaynaklar serbest bırakılmaz.

- Kalıcı bir dinamik kuyruk silindiğinde, MQCLOSE çağrısında belirtilen *HOBJ* tanıtıcısı, kuyruğu yaratan MQOPEN çağrısının döndürdüğü tanıtıcı değilse, MQOPEN çağrısının geçerliliğini denetlemek için kullanılan kullanıcı kimliğinin kuyruğu silme yetkisi olduğu denetlenir. MQOPEN çağrısında OOALTU seçeneği belirtildiyse, denetlenen kullanıcı kimliği *ODAU* olur.

Bu denetim aşağıdaki durumda gerçekleştirilmez:

- Belirtilen tanıtıcı, kuyruğu yaratan MQOPEN çağrısıyla döndürülen tanıtıcı.
- Silinmekte olan kuyruk geçici bir dinamik kuyruk.
- Geçici bir dinamik kuyruk kapatıldığında, MQCLOSE çağrısında belirtilen *HOBJ* tanıtıcısı, kuyruğu yaratan MQOPEN çağrısıyla döndürülen tanıtıcıysa, kuyruk silinir. Bu durum, MQCLOSE çağrısında belirtilen kapatma seçeneklerinden bağımsız olarak oluşur. Kuyrukta ileti varsa, bunlar atılır; rapor iletisi oluşturulmaz.

Kuyruğu etkileyen kesinleştirilmemiş iş birimleri varsa, kuyruk ve iletileri silinmeye devam eder, ancak bu, iş birimlerinin başarısız olmasına neden olmaz. Ancak, daha önce açıklandığı gibi, iş birimleriyle ilişkili kaynaklar, her bir iş birimi kesinleştirilinceye ya da geri çekilinceye kadar serbest bırakılmaz.

4. Kapatılan nesne bir *dağıtım listesi* ise aşağıdaki noktalar geçerlidir:

- Dağıtım listesi için geçerli olan tek kapatma seçeneği CONONE değeridir; başka seçenek belirtildiyse, çağrı RC2046 ya da RC2045 neden koduyla başarısız olur.
- Bir dağıtım listesi kapatıldığında, listedeki kuyruklar için tek tek tamamlama kodları ve neden kodları döndürülmez; yalnızca, tanılama amacıyla çağrıya ilişkin **CMPCOD** ve **REASON** parametreleri kullanılabilir.

Kuyruklardan biri kapatılırken bir hata oluşursa, kuyruk yöneticisi işlemeye devam eder ve dağıtım listesinde kalan kuyrukları kapatmayı dener. Daha sonra, çağrıya ilişkin **CMPCOD** ve **REASON** parametreleri, hatayı açıklayan bilgileri döndürecek şekilde ayarlanır. Bu nedenle, kuyrukların çoğu başarıyla kapatılmış olsa da, tamamlanma kodunun CCFAIL olması mümkündür. Hatayla karşılaşan kuyruk tanımlanmaz.

Birden çok kuyrukta bir hata varsa, **CMPCOD** ve **REASON** parametrelerinde hangi hatanın bildirildiği tanımlanmaz.

Parametreler

MQCLOSE çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı iřaretli tamsayı)-giriř

Baęlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik baęlantıyı gösterir. *HCONN* deęeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

HOBJ (10 basamaklı iřaretli tamsayı)-giriř/çıkıř

Nesne tanıtıcısı.

Bu tanıtıcı, kapatılan nesneyi gösterir. Nesne herhangi bir tipte olabilir. *HOBJ* değeri önceki bir *MQOPEN* çağrısı tarafından döndürüldü.

Çağrı başarıyla tamamlandığında, kuyruk yöneticisi bu parametreyi ortam için geçerli bir tanıtıcı olmayan bir değere ayarlar. Bu değer:

HOUNUH.

Kullanılmayan nesne tanıtıcısı.

OPTS (10 basamaklı işaretli tamsayı)-giriş

MQCLOSE işlemini denetleyen seçenekler.

OPTS parametresi, nesnenin nasıl kapatılacağını denetler. Yalnızca kalıcı dinamik kuyruklar ve abonelikler birden çok şekilde kapatılabilir. Kalıcı dinamik kuyruklar saklanabilir ya da silinebilir; bunlar, *QDPERM* değerine sahip **DefinitionType** özniteliğine sahip kuyruklardır ("Kuyruklara ilişkin öznitelikler" sayfa 1345 içinde açıklanan **DefinitionType** özniteliğine bakın). Kapanış seçenekleri, bu konunun sonraki kısımları bir tabloda özetlenir.

Sürekli abonelikler alınabilir ya da kaldırılabilir; bunlar *SODUR* seçeneğiyle *MQSUB* çağrısı kullanılarak yaratılır.

Tanıtıcıyı yönetilen bir hedefe kapatırken (*SOMAN* seçeneğini kullanan bir *MQSUB* çağrısında döndürülen **Hobj** değiştirgesi), ilişkili abonelik de kaldırıldığında kuyruk yöneticisi alınmamış yayınları temizler. Bu, *MQSUB* çağrısında döndürülen **Hsub** değiştirgesinde *CORMSB* seçeneği kullanılarak yapılır. *CORMSB* ' nin kalıcı olmayan bir abonelik için *MQCLOSE* üzerinde varsayılan davranış olduğunu unutmayın.

Bir tutamacı yönetilmeyen bir hedefe kapatırken, yayınların gönderildiği kuyruğu temizlemek sizin sorumluluğunuzdadır. Önce *CORMSB* kullanarak aboneliği kapatmanız ve sonra ileti kalmayınca kadar kuyruktaki iletileri işlemeniz önerilir.

Aşağıdakilerden biri (ve yalnızca biri) belirtilmelidir:

Dinamik kuyruk kapatma seçenekleri

Bu seçenekler, kalıcı dinamik kuyrukların nasıl kapatılacağını denetler:

KOD

Kuyruğu silin.

Aşağıdakilerden biri doğruysa, kuyruk silinir:

- Bu, önceki bir *MQOPEN* çağrısı tarafından yaratılan kalıcı bir dinamik kuyruktur ve kuyruktaki ileti yoktur ve kuyruk için kesinleştirilmemiş alma ya da yerleştirme istekleri yoktur (yürürlükteki görev ya da başka bir görev için).
- *HOBJ* ögesini döndüren, *MQOPEN* çağrısıyla yaratılan geçici dinamik kuyruktur. Bu durumda, kuyruktaki tüm iletiler temizlenir.

Hobj ' in bir *MQSUB* çağrısında döndürüldüğü durum da içinde olmak üzere, diğer tüm durumlarda çağrı *RC2045* neden koduyla başarısız olur ve nesne silinmez.

KOPYALANIR

Kuyruktaki iletileri temizleyip kuyruğu silin.

Aşağıdakilerden biri doğruysa, kuyruk silinir:

- Bu, önceki bir *MQOPEN* çağrısı tarafından yaratılan kalıcı dinamik bir kuyruktur ve kuyruk için (yürürlükteki görev ya da başka bir görev için) kesinleştirilmemiş alma ya da koyma isteği yoktur.
- *HOBJ* ögesini döndüren, *MQOPEN* çağrısıyla yaratılan geçici dinamik kuyruktur.

Hobj ' in bir *MQSUB* çağrısında döndürüldüğü durum da içinde olmak üzere, diğer tüm durumlarda çağrı *RC2045* neden koduyla başarısız olur ve nesne silinmez.

Sonraki tablo, hangi kapatma seçeneklerinin geçerli olduğunu ve nesnenin korunduğunu ya da silindiğini gösterir.

<i>Çizelge 743. Alıkonan ya da silinen nesnelere kullanılmak üzere geçerli kapatma seçenekleri</i>			
Nesne ya da kuyruk tipi	KONONE	KOD	KOPYALANIR
Kuyruk dışındaki bir nesne	Alıkondu	Geçerli değil	Geçerli değil
Önceden tanımlı kuyruk	Alıkondu	Geçerli değil	Geçerli değil
Kalıcı dinamik kuyruk	Alıkondu	Boşsa ve bekleyen güncelleme yoksa silindi	İletiler silindi; bekleyen güncelleme yoksa kuyruk silindi
Geçici dinamik kuyruk (kuyruk yaratan tarafından çağrı yayınlandı)	Silindiği Tarih	Silindiği Tarih	Silindiği Tarih
Geçici dinamik kuyruk (kuyruk yaratan tarafından çağrı yayınlanmadı)	Alıkondu	Geçerli değil	Geçerli değil
Dağıtım listesi	Alıkondu	Geçerli değil	Geçerli değil
Yönetilen abonelik hedefi	Alıkondu	Geçerli değil	Geçerli değil
Dağıtım listesi (abonelik kaldırıldı)	İletiler silindi; kuyruk silindi	Geçerli değil	Geçerli değil

Abonelik kapatma seçenekleri

Bu seçenekler, tutamaç kapatıldığında sürekli aboneliklerin kaldırılıp kaldırılmayacağını ve uygulama tarafından okunmayı bekleyen yayınların temizlenip temizlenmeyeceğini denetler. Bu seçenekler yalnızca, MQSUB çağrısının **Hsub** değiştirilmesinde döndürülen bir nesne tanıtıcısıyla kullanılmak üzere geçerlidir.

COKPSB

Abonelik tanıtıcısı kapatıldı, ancak yapılan abonelik korundu. Yayınlar, abonelikte belirtilen hedefe gönderilmeye devam edecektir. Bu seçenek, abonelik SODUR seçeneğiyle yapıldıysa geçerlidir. Abonelik dayanıklıysa varsayılan değer COKPSB 'dir

KORMSB

Abonelik kaldırılır ve abonelik tanıtıcısı kapatılır.

MQSUB çağrısının **Hobj** değiştirilmesi **Hsub** değiştirilmesi kapatılarak geçersiz kılınmaz ve geri kalan yayınları almak için MQGET ya da MQCB için kullanılmaya devam edilebilir. MQSUB çağrısının **Hobj** değiştirilmesi de kapatıldığında, yönetilen bir hedefse, alınmamış yayınlar kaldırılır.

Abonelik kalıcı değilse, CORMSB varsayılan değerdir.

Bu abonelik kapatma seçenekleri aşağıdaki tablolarda özetlenmiştir:

Sürekli bir abonelik tanıtıcısını kapatmak, ancak aboneliği çevresinde bırakmak için aşağıdaki abonelik kapatma seçeneklerini kullanın:

<i>Çizelge 744. Sürekli bir abonelik tanıtıcısını kapatmak ve aboneliği çevresinde bırakmak için görev seçenekleri</i>	
Görev	Abonelik kapatma seçeneği
Yayınları MQOPENed tutamacında tut	COKPSB
MQOPENed tanıtıcısındaki yayınları kaldır	İşleme izin verilmiyor
SOMAN ile yayınları tutma	COKPSB
SOMAN ile bir tutamaçtaki yayınların kaldırılması	İşleme izin verilmiyor

Sürekli bir abonelik tanıtıcısını kapatıp aboneliğini kaldırarak ya da kalıcı olmayan bir abonelik tanıtıcısını kapatarak aboneliğinizi kaldırmak için aşağıdaki abonelik kapatma seçeneklerini kullanın:

Çizelge 745. Aboneliği kaldırmak için görev seçenekleri	
Görev	Abonelik kapatma seçeneği
Yayınları MÇOPENed tutamacında tut	KORMSB
MÇOPENed tanıtıcısındaki yayınları kaldır	İşleme izin verilmiyor
SOMAN ile yayınları tutma	KORMSB
SOMAN ile bir tutamaçtaki yayınların kaldırılması	COPGSB

Önden okuma seçenekleri

Aşağıdaki seçenekler, bir uygulama tarafından istenmeden önce istemciye gönderilen ve henüz uygulama tarafından tüketilmeyen kalıcı olmayan iletilere ne olacağını denetler. Bu iletiler, uygulama tarafından istenmeyi bekleyen istemci önden okuma arabelleğinde saklanır ve MÇCLOSE tamamlanmadan önce atılabilir ya da kuyruktan tüketilebilir.

COIMM.

Nesne hemen kapatılır ve istenmeden önce istemciye gönderilen iletiler atılır ve herhangi bir uygulama tarafından tüketilemez. Bu varsayılan değerdir.

COQSC

Nesneyi kapatma isteğinde bulunursanız, ancak bir uygulama bunları istemeden önce istemciye gönderilen iletiler istemcinin önden okuma arabelleğinde bulunmaya devam ediyorsa, MÇCLOSE çağrısı RC2458uyarı koduyla döndürülür ve nesne tanıtıcısı geçerli kalır.

Daha sonra uygulama, ileti almak için nesne tanıtıcısını kullanmaya devam edebilir ve daha sonra, nesneyi yeniden kapatabilir. İstekte bulunan bir uygulamadan önce istemciye başka ileti gönderilmez, önden okuma kapatılmış olur.

Son MÇGET çağrısı ile COIMM kullanılırsa atılacak aşağıdaki MÇCLOSE arasında bir ileti olabileceği için, uygulamaların istemcide önden okuma arabelleğinde başka ileti olmadığı bir noktaya ulaşmaya çalışmak yerine COQSC kullanması önerilir.

Zamanuyumsuz bir geri çağırma işlevinden COQSC içeren bir MÇCLOSE yayınlandıysa, önden okuma iletileriyle aynı davranış geçerli olur. RC2458 uyarı kodu döndürülürse, geri çağırma işlevi en az bir kez daha çağrılır. Önden okunan son ileti geri çağırma işlevine geçirildiğinde, CBCFLG alanı CBCFBE olarak ayarlanır.

Varsayılan seçenek

Daha önce açıklanan seçeneklerden hiçbirine gerek duymazsanız, aşağıdaki seçeneği kullanabilirsiniz:

KONONE

İsteğe bağlı kapatma işlemi gerekmez.

Bu, aşağıdakiler için belirtilmelidir:

- Kuyruklar dışındaki nesnelere
- Önceden tanımlı kuyruklar
- Geçici dinamik kuyruklar (ancak, *HOB*J ' in kuyruğu yaratan MÇOPEN çağrısıyla döndürülen tanıtıcı olmadığı durumlarda).
- Dağıtım listeleri

Önceki tüm durumlarda nesne alıkonur ve silinmez.

Geçici dinamik kuyruk için bu seçenek belirtilirse:

- Kuyruk, *HOB*J ; döndüren MÇOPEN çağrısıyla yaratıldıysa silinir. kuyruktaki iletiler temizlenir.
- Diğer tüm durumlarda kuyruk (ve üzerindeki iletiler) alıkonur.

Kalıcı dinamik kuyruk için bu seçenek belirtilirse, kuyruk alıkonur ve silinmez.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2241

(2241, X'8C1') İleti grubu tamamlanmadı.

RC2242

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

CMPCOD CCFAIL ise:

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2019

(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

RC2035

(2035, X'7F3') Erişim için yetkili değil.

RC2101

(2101, X'835 ') Nesne zarar gördü.

RC2045

(2045, X'7FD') Seçenek, nesne tipi için geçerli değil.

RC2046

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2058

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

RC2059

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2055

(2055, X'807 ') Kuyruk bir ya da daha fazla ileti ya da kesinleştirilmemiş koyma ya da alma isteği içeriyor.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2063

(2063, X'80F') Güvenlik hatası oluştu.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCLOSE(HCONN : HOBJ : OPTS :
C                               CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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

IBM i üzerinde MQCMIT (Değişiklikleri kesinleştir)

MQCMIT çağrısı, kuyruk yöneticisine uygulamanın bir eşitleme noktasına ulaştığını ve son eşitleme noktasından bu yana oluşan tüm iletilerin kalıcı hale getirileceğini belirtir. Bir iş biriminin parçası olarak konan iletiler diğer uygulamalar tarafından kullanılabilir; bir iş biriminin parçası olarak alınan iletiler silinir.

- [“Sözdizimi” sayfa 1253](#)
- [“Kullanım notları” sayfa 1253](#)
- [“Parametreler” sayfa 1254](#)
- [“RPG Bildirimi” sayfa 1255](#)

Sözdizimi

MQCMIT (HCONN, COMCOD, REASON)

Kullanım notları

MQCMIT kullanırken bu kullanım notlarını göz önünde bulundurun.

1. Bu çağrı yalnızca kuyruk yöneticisinin kendisi iş birimini koordine ettiğinde kullanılabilir. Bu, değişikliklerin yalnızca IBM MQ kaynaklarını etkilediği yerel bir iş birimidir.
2. Kuyruk yöneticisinin iş birimini koordine etmediği ortamlarda, MQCMIT yerine uygun kesinleştirme çağrısı kullanılmalıdır. Ortam, uygulamanın olağan şekilde sonlandırılmasından kaynaklanan örtük bir kesinleştirmeyi de destekleyebilir.
 - IBM i' da bu çağrı, kuyruk yöneticisi tarafından eşgüdümlü yerel iş birimleri için kullanılabilir. Bu, iş düzeyinde bir kesinleştirme tanımının var olmaması gerektiği, yani iş için **CMTSCOPE (*JOB)** parametresini içeren STRCMTCTL komutunun yayınlanmamış olması gerektiği anlamına gelir.
3. Bir uygulama, bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması uygulamanın olağan şekilde mi, yoksa olağandışı olarak mı bittiğine bağlıdır. Daha fazla ayrıntı için

“IBM i üzerinde MQDISC (Kuyruk yöneticisinin bağlantısını kes)” sayfa 1268 içindeki kullanım notlarına bakın.

4. Bir uygulama, mantıksal ileti gruplarına ya da bölümlerine ileti girdiğinde ya da aldığı anda, kuyruk yöneticisi, son başarılı MQPUT ve MQGET çağrılarında ilişkin ileti grubu ve mantıksal iletiyle ilgili bilgileri saklar. Bu bilgiler kuyruk tanıtıcısıyla ilişkilidir ve aşağıdaki gibi şeyleri içerir:

- MQMD 'deki *MDGID*, *MDSEQ*, *MDOFF* ve *MDMFL* alanlarının değerleri.
- İletinin bir iş biriminin parçası olup olmadığı.
- MQPUT çağrısı için: İletinin kalıcı mı, kalıcı mı olduğu.

Bir iş birimi kesinleştirildiğinde, kuyruk yöneticisi grup ve bölüm bilgilerini saklar ve uygulama iletileri yürürlükteki ileti grubuna ya da mantıksal iletiye yerleştirmeye ya da almaya devam edebilir.

Bir iş birimi kesinleştirildiğinde grup ve bölüm bilgilerinin tutulması, uygulamanın çeşitli iş birimlerinde birçok bölümden oluşan büyük bir ileti grubu ya da büyük bir mantıksal ileti dağıtmasına olanak sağlar. Yerel kuyruk yöneticisinde yalnızca sınırlı kuyruk saklama alanı varsa, birkaç iş birimi kullanılması avantajlı olabilir. Ancak, bir sistem arızası ortaya çıkarsa, uygulamanın iletileri doğru noktada yeniden koymayı ya da almayı yeniden başlatılabilmesi için yeterli bilgiyi koruması gerekir. Bir sistem hatasından sonra doğru noktada nasıl yeniden başlatılacağına ilişkin ayrıntılar için “IBM i üzerinde MQPMO (Put-message options)” sayfa 1154 içinde açıklanan PMLOGO seçeneğine ve “IBM i üzerinde MQGMO (Get-message seçenekleri)” sayfa 1060 içinde açıklanan GMLOGO seçeneğine bakın.

Geri kalan kullanım notları yalnızca kuyruk yöneticisi iş birimlerini koordine ettiğinde geçerlidir:

1. Bir iş birimi, bağlantı tanıtıcısıyla aynı kapsama sahiptir. Bu, belirli bir iş birimini etkileyen tüm IBM MQ çağrılarının aynı bağlantı tanıtıcısı kullanılarak gerçekleştirilmesi gerektiği anlamına gelir. Farklı bir bağlantı tanıtıcısı kullanılarak verilen çağrılar (örneğin, başka bir uygulama tarafından verilen çağrılar) farklı bir iş birimini etkiler. Bağlantı tanıtıcılarının kapsamıyla ilgili bilgi için MQCONN içinde açıklanan **HCONN** değiştirilmesine bakın.
2. Yalnızca yürürlükteki iş biriminin bir parçası olarak konan ya da alınan iletiler bu çağrıdan etkilenir.
3. Bir iş birimi içinde MQGET, MQPUT ya da MQPUT1 çağrılarını veren, ancak hiçbir zaman kesinleştirme ya da gerileme çağrısı yapmayan uzun süreli bir uygulama, kuyrukların diğer uygulamalar tarafından kullanılmayan iletilerle dolmasına neden olabilir. Bu olasılığa karşı koruma sağlamak için, denetimcinin **MaxUncommittedMsgs** kuyruk yöneticisi özneliğini, kuyrukları dolduran kaçak uygulamaları önleyecek kadar düşük, ancak beklenen ileti sistemi uygulamalarının düzgün çalışmasını sağlayacak kadar yüksek bir değere ayarlaması gerekir.

Parametreler

MQCMIT çağrısındaki değiştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *HCONN* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

COMCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *COMCOD*.

COMCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

COMCOD CCWARN ise:

RC2003

(2003, X'7D3') Çalışma birimi geri çekildi.

RC2124

(2124, X'84C') Kesinleştirme işleminin sonucu beklemede.

COMCOD CCFAIL ise:

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2101

(2101, X'835 ') Nesne zarar gördü.

RC2123

(2123, X'84B') Kesinleştirme ya da geri yükleme işleminin sonucu karışık.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C                               CALLP      MQCMIT(HCONN : COMCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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

IBM i üzerinde MQCONN (Connect kuyruk yöneticisi)

MQCONN çağrısı, bir uygulama programını bir kuyruk yöneticisine bağlar. Sonraki ileti kuyruklama çağrılarında uygulama tarafından kullanılan bir kuyruk yöneticisi bağlantı tanıtıcısı sağlar.

- Uygulamalar kuyruk yöneticisine bağlanmak için MQCONN ya da MQCONNX çağrısı ve kuyruk yöneticisiyle bağlantıyı kesmek için MQDISC çağrısı kullanılmalıdır.

IBM MQ for Multiplatforms üzerinde, bir uygulamadaki her iş parçacığı farklı kuyruk yöneticilerine bağlanabilir. Diğer sistemlerde, bir işlemdeki tüm eşzamanlı bağlantıların aynı kuyruk yöneticisine olması gerekir.

- [“Sözdizimi” sayfa 1256](#)
- [“Kullanım notları” sayfa 1256](#)
- [“Parametreler” sayfa 1256](#)
- [“RPG Bildirimi” sayfa 1259](#)

Sözdizimi

MQCONN (QMNAME, HCONN, CMPCOD, REASON)

Kullanım notları

1. MQCONN çağrısı kullanılarak bağlantı kurulan kuyruk yöneticisine *yerel kuyruk yöneticisi* adı verilir.
2. Yerel kuyruk yöneticisinin sahip olduğu kuyruklar, uygulamada yerel kuyruklar olarak görünür. İletiler takılabilir ve bu kuyruklardan ileti alabilir.

Yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun sahip olduğu paylaşılan kuyruklar uygulamaya yerel kuyruklar olarak görünür. İletiler takılabilir ve bu kuyruklardan ileti alabilir.

Uzak kuyruk yöneticilerinin iyeliğindeki kuyruklar uzak kuyruklar olarak görünür. Bu kuyruklara ileti konması mümkündür, ancak bu kuyruklardan ileti almak mümkün değildir.

3. Bir uygulama çalışırken kuyruk yöneticisi başarısız olursa, sonraki IBM MQ çağrılarında kullanılacak yeni bir bağlantı tanıtıcısı elde etmek için uygulamanın MQCONN çağrısını yeniden yürütmesi gerekir. Uygulama, çağrı başarılı oluncaya kadar düzenli aralıklarla MQCONN çağrısına devam edebilir.

Bir uygulama kuyruk yöneticisine bağlı olup olmadığından emin değilse, uygulama bağlantı tanıtıcısı elde etmek için güvenli bir şekilde bir MQCONN çağrısı verebilir. Uygulama zaten bağlıysa, döndürülen tanıtıcı önceki MQCONN çağrısıyla aynıdır, ancak tamamlama kodu CCWARN ve neden kodu RC2002 ile döndürülür.

4. Uygulama IBM MQ çağrılarını kullanmayı bitirdiğinde, kuyruk yöneticisiyle bağlantısını kesmek için MQDISC çağrısına başvurmalıdır.
5. IBM üzerinde, olağandışı sona eren programların kuyruk yöneticisiyle bağlantısı otomatik olarak kesilmez. Bu nedenle, CCWARN tamamlama kodunu ve RC2002 neden kodunu döndüren MQCONN ya da MQCONNX çağrısı olasılığına izin vermek için uygulamalar yazılmalıdır. Bu durumda döndürülen bağlantı tanıtıcısı olağan olarak kullanılabilir.

Parametreler

MQCONN çağrısındaki deęiřtirgeler řunlardır:

QMNAME (48 baytlık karakter dizisi)-giriř

Kuyruk yöneticisinin adı.

Bu, uygulamanın bağlanmak istedięi kuyruk yöneticisinin adıdır. Ad ařaęıdaki karakterleri içerebilir:

- Büyük harf alfabetik karakterler (A-Z)
- Küçük harf alfabetik karakterler (a-z)
- Sayısal basamaklar (0-9)
- Nokta (.), eğik çizgi (/), altçizgi (_), yüzde (%)

Ad başta ya da gömülü boşluklar içermemelidir, ancak sonda boşluklar içerebilir. Addaki önemli verilerin sonunu göstermek için boş karakter kullanılabilir; boş değer ve ardından gelen karakterler boşluk olarak değerlendirilir. Belirtilen ortamlarda aşağıdaki kısıtlamalar geçerlidir:

- IBM üzerinde, komutlarda belirtildiğinde küçük harf, eğik çizgi ya da yüzde içeren adlar tırnak içine alınmalıdır. **QMNAME** değiştirilmesinde bu tırnak işaretleri belirtilmemelidir.

Ad tamamen boşluklardan oluşuyorsa, *varsayılan* kuyruk yöneticisinin adı kullanılır.

QMNAME için belirtilen ad, *bağlanabilir* bir kuyruk yöneticisinin adı olmalıdır.

Kuyruk paylaşım grupları: Birden çok kuyruk yöneticisinin bulunduğu ve bir kuyruk paylaşım grubu oluşturmak üzere yapılandırıldığı sistemlerde, kuyruk yöneticisi adı yerine **QMNAME** için kuyruk paylaşım grubunun adı belirtilebilir. Bu, uygulamanın kuyruk paylaşım grubunda bulunan *herhangi bir* kuyruk yöneticisine bağlanmasını sağlar. Sistem ayrıca, boş bir **QMNAME** varsayılan kuyruk yöneticisi yerine kuyruk paylaşım grubuyla bağlantıya neden olacak şekilde yapılandırılabilir.

QMNAME , kuyruk paylaşım grubunun adını belirtirse, ancak sistemde bu adı taşıyan bir kuyruk yöneticisi de varsa, öncekiyle bağlantı kurulur. Yalnızca bu bağlantı başarısız olursa, denenen kuyruk paylaşım grubundaki kuyruk yöneticilerinden biriyle bağlantı kurulur.

Bağlantı başarılı olursa, **MQCONN** ya da **MQCONNX** çağrısının döndürdüğü tanıtıcı, bağlantının kurulduğu kuyruk yöneticisine ait olan *tüm* kaynaklara (paylaşılan ve paylaşılmayan) erişmek için kullanılabilir. Bu kaynaklara erişim, tipik yetki denetimlerine tabidir.

Uygulama eşzamanlı bağlantı kurmak için iki **MQCONN** ya da **MQCONNX** çağrısı yaparsa ve bir ya da her iki çağrı kuyruk paylaşım grubunun adını belirtiyorsa, ikinci çağrı tamamlanma kodu **CCWARN** ve neden kodu **RC2002**döndürebilir. Bu durum, ikinci çağrı ilk çağrıyla aynı kuyruk yöneticisine bağlandığında ortaya çıkar.

Kuyruk paylaşım grupları yalnızca **z/OS** üzerinde desteklenir. Bir kuyruk paylaşım grubuna bağlantı yalnızca toplu iş, **RRS** toplu iş ve **TSO** ortamlarında desteklenir.

IBM MQ istemci uygulamaları: IBM MQ MQI client uygulamaları için, her istemci-bağlantı kanalı tanımlaması için belirtilen kuyruk yöneticisi adıyla bağlantı kurulmaya çalışılır. Ancak, kuyruk yöneticisi belirtilen adla aynı ada sahip olmalıdır. Tümü boş bir ad belirtilirse, kuyruk yöneticisi adı boş olan her istemci bağlantı kanalı, biri başarılı oluncaya kadar denir; bu durumda, kuyruk yöneticisinin gerçek adıyla ilgili bir denetim olmaz.

IBM MQ İstemci kuyruk yöneticisi grupları: Belirtilen ad bir yıldız işaretiyle (*) başlıyorsa, bağlantının kurulduğu gerçek kuyruk yöneticisinin adı uygulama tarafından belirtilenden farklı olabilir. Belirtilen ad (yıldız işareti olmadan), bağlantı için uygun kuyruk yöneticilerinden oluşan bir *grubu* tanımlar. Uygulama, bir bağlantının kurulabileceği bulunana kadar her birini alfabetik sırayla deneyerek gruptan bir tane seçer. Gruptaki kuyruk yöneticilerinden hiçbiri bağlantı için uygun değilse, çağrı başarısız olur. Her kuyruk yöneticisi yalnızca bir kez denir. Ad için tek başına bir yıldız imi belirtilirse, somutlama tanımlı bir varsayılan kuyruk yöneticisi grubu kullanılır.

Kuyruk yöneticisi grupları yalnızca MQ-istemcisi ortamında çalışan uygulamalar için desteklenir; istemci olmayan bir uygulama yıldız işaretiyle başlayan bir kuyruk yöneticisi adını belirtirse çağrı başarısız olur. Grup, gruptaki her kuyruk yöneticisiyle iletişim kurmak için, aynı kuyruk yöneticisi adıyla (yıldız imi olmadan belirtilen ad) birden çok istemci bağlantı kanalı tanımlaması sağlanarak tanımlanır. Varsayılan grup, her biri boş bir kuyruk yöneticisi adı olan bir ya da daha çok istemci bağlantı kanalı tanımlı sağlanarak tanımlanır (bu nedenle, tümü boş bir ad belirtilmesi, istemci uygulaması için ad için tek bir yıldız imi belirlenmesiyle aynı etkiye sahiptir).

Bir grubun bir kuyruk yöneticisine bağlandıktan sonra, bir uygulama, ileti ve nesne tanımlayıcılarındaki kuyruk yöneticisi ad alanlarında, uygulamanın gerçekten bağlandığı kuyruk yöneticisinin adını (*yerel kuyruk yöneticisi*) ifade etmek için tipik olarak boşlukları belirtebilir. Uygulamanın bu adı bilmesi gerekiyorsa, **QMGRNAME** kuyruk yöneticisi özneliğini sorgulamak için **MQINQ** çağrısı yayınlanabilir.

Bağlantı adına bir yıldız imi konması, uygulamanın gruptaki belirli bir kuyruk yöneticisine bağlanmasına bağımlı olmadığı anlamına gelir. Uygun uygulamalar şunlardır:

- İletileri koyan, ancak ileti almayan uygulamalar.

- İstek iletileri koyan ve yanıt iletilerini *geçici dinamik* kuyruktan alan uygulamalar.

Belirli bir kuyruk yöneticisindeki belirli bir kuyruktan ileti almak isteyenler uygun olmayan uygulamalardır; bu tür uygulamalar adın başına yıldız işareti koymamalıdır.

Bir yıldız imi belirlenirse, adın geri kalanının uzunluk üst sınırı 47 karakterdir.

Bu parametrenin uzunluğu LNQMN tarafından verilir.

HCONN (10 basamaklı işaretli tamsayı)-çıkış

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. Bu, uygulama tarafından yayınlanan sonraki tüm ileti kuyruklama çağrılarında belirtilmelidir. MQDISC çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde geçerli olmaz.

Tutamacın kapsamı, Uygulamanın çalıştığı platform tarafından desteklenen koşut işleme; tanıtıcı, MQCONN çağrısının verildiği koşut işleme biriminin dışında geçerli değil.

- IBM i' da, tanıtıcı kapsamı, çağrıyı yayınlayan iştir.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2002

(2002, X'7D2') Uygulama zaten bağlı.

CMPCOD CCFAIL ise:

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2267

(2267, X'8DB') Küme iş yükü çıkışı yüklenemiyor.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2035

(2035, X'7F3') Erişim için yetkili değil.

RC2137

(2137, X'859 ') Nesne başarıyla açılmadı.

RC2058

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

RC2059

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.

RC2161

(2161, X'871 ') Kuyruk yöneticisi susturuyor.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2063

(2063, X'80F') Güvenlik hatası oluştu.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCONN(QMNAME : HCONN : CMPCOD :
C                                REASON)

```

Çağrıya ilişkin prototip tanımlaması:

```

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 (IBM i üzerinde bağlantı kuyruğu yöneticisi (genişletilmiş))

MQCONNX çağrısı, bir uygulama programını bir kuyruk yöneticisine bağlar. Sonraki IBM MQ çağrılarında uygulama tarafından kullanılan bir kuyruk yöneticisi bağlantı tanıttıcısı sağlar.

MQCONNX çağrısı MQCONN çağrısına benzer, ancak MQCONNX, çağrılarının çalışma şeklini denetlemek için seçeneklerin belirtilmesine izin verir.

IBM MQ for Multiplatforms üzerinde, bir uygulamadaki her iş parçacığı farklı kuyruk yöneticilerine bağlanabilir. Diğer sistemlerde, bir işlemdeki tüm eşzamanlı bağlantıların aynı kuyruk yöneticisine olması gerekir.

- [“Sözdizimi” sayfa 1259](#)
- [“Parametreler” sayfa 1260](#)
- [“RPG Bildirimi” sayfa 1260](#)

Sözdizimi

MQCONNX (QMNAME, CNOPT, HCONN, CMPCOD, REASON)

Parametreler

MQCONN çağrısındaki deęiřtirgeler řunlardır:

QMNAME (48 baytlık karakter dizgisi)-giriř

Kuyruk yöneticisinin adı.

Ayrıntılar için “IBM i üzerinde MQCONN (Connect kuyruk yöneticisi)” sayfa 1255 içinde açıklanan **QMNAME** deęiřtirgesine bakın.

CNOPT (MQCNO)-giriř/çıkıř

MQCONN iřlemini denetleyen seenekler.

Ayrıntılar için bkz. “IBM i üzerinde MQCNO (Baęlantı seenekleri)” sayfa 1031.

HCONN (10 basamaklı iřaretili tamsayı)-çıkıř

Baęlantı tanıtıcısı.

Ayrıntılar için “IBM i üzerinde MQCONN (Connect kuyruk yöneticisi)” sayfa 1255 içinde açıklanan **HCONN** deęiřtirgesine bakın.

CMPCOD (10 basamaklı iřaretili tamsayı)-çıkıř

Tamamlanma kodu.

Ayrıntılar için “IBM i üzerinde MQCONN (Connect kuyruk yöneticisi)” sayfa 1255 içinde açıklanan **CMPCOD** deęiřtirgesine bakın.

REASON (10 basamaklı iřaretili tamsayı)-çıkıř

Neden kodu niteleyici *CMPCOD*.

Olası neden kodlarına iliřkin ayrıntılar için “IBM i üzerinde MQCONN (Connect kuyruk yöneticisi)” sayfa 1255 içinde açıklanan **REASON** deęiřtirgesine bakın.

MQCONN çağrısıyla ařaęıdaki ek neden kodları döndürülebilir:

CMPCOD CCFAIL ise:

RC2278

(2278, X'8E6') İstemci baęlantısı alanları geersiz.

RC2139

(2139, X'85B') Baęlantı seenekleri yapısı geersiz.

RC2046

(2046, X'7FE') Seenekler geerli deęil ya da tutarlı deęil.

RPG Bildirimi

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCONN(QMNAME : HCONN : CMPCOD :
C                               REASON)
```

Çaęrıya iliřkin prototip tanımlaması:

```
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
```

IBM i IBM i üzerinde MQCRTMH (İleti tanıtıcısı yarat)

MQCRTMH çağrısı bir ileti tanıtıcısı döndürür.

Bir uygulama, sonraki ileti kuyruklama çağrılarında bunu kullanabilir:

- İleti tanıtıcısının bir özelliğini ayarlamak için [MQSETMP](#) çağrısını kullanın.
- İleti tanıtıcısının bir özelliğinin değerini sorgulamak için [MQINQMP](#) çağrısını kullanın.
- İleti tanıtıcısının bir özelliğini silmek için [MQDLTMP](#) çağrısını kullanın.

İleti tanıtıcısı MQPUT ve MQPUT1 çağrılarında, ileti tanıtıcısının özelliklerini, konmakta olan iletinin özellikleriyle ilişkilendirmek için kullanılabilir. Benzer şekilde, MQGET çağrısında bir ileti tanıtıcısı belirtilerek, alınmakta olan iletinin özelliklerine, MQGET çağrısı tamamlandığında ileti tanıtıcısı kullanılarak erişilebilir.

İleti tanıtıcısını silmek için [MQDLTMH](#) komutunu kullanın.

- “Sözdizimi” sayfa 1261
- “Parametreler” sayfa 1261
- “RPG Bildirimi” sayfa 1263

Sözdizimi

MQCRTMH (*Hconn, CrtMsgH0pts, Hmsg, CompCode, Reason*)

Parametreler

MQCRTMH çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı iřaretli tamsayı)-giriř

Bu tanıtıcı, kuyruk yöneticisine yönelik baęlantıyı gösterir. *HCONN* deęeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü. Kuyruk yöneticisine yönelik baęlantı geçerlilięini keserse ve ileti tanıtıcısında IBM MQ çağrısı çalışmıyorsa, iletiyi silmek için [MQDLTMH](#) örtük olarak çağrılır.

Diđer bir seçenek olarak, ařaęıdaki deęeri belirtebilirsiniz:

HCUNAS

Baęlantı tanıtıcısı, belirli bir kuyruk yöneticisine yönelik bir baęlantıyı göstermiyor.

Bu deęer kullanıldığında, kendisine ayrılan saklama alanını serbest bırakmak için ileti tanıtıcısı [MQDLTMH](#) ' e yönelik belirtik bir çağrıyla silinmelidir; IBM MQ ileti tanıtıcısını hiçbir zaman örtük olarak silmez.

İleti tanıtıcısını yaratan iř parçacığında kurulmuř bir kuyruk yöneticisiyle en az bir geçerli baęlantı olmalıdır; tersi durumda, çağrı RC2018ile başarısız olur.

CRTOPT (MQCMHO)-giriř

MQCRTMH iřlemini denetleyen seçenekler. Ayrıntılar için bkz. [MQCMHO](#) .

HMSG (20 basamaklı iřaretli tamsayı)-çıkıř

Çıkıřta, ileti tanıtıcısının özelliklerini ayarlamak, sorgulamak ve silmek için kullanılacak bir ileti tanıtıcısı döndürülür. Bařlangıçta ileti tanıtıcısı özellik içermiyor.

İleti tanıtıcısının iliřkili bir ileti tanımlayıcısı da vardır. Bařlangıçta bu ileti tanımlayıcısı varsayılan deęerleri içerir. İliřkili ileti tanımlayıcı alanlarının deęerleri, MQSETMP ve MQINQMP çağrıları kullanılarak ayarlanabilir ve sorilebilir. MQDLTMP çağrısı, ileti tanımlayıcısının bir alanını varsayılan deęerine geri çevirir.

HCONN deęiřtirgesi *HCUNAS* deęeri olarak belirtilirse, dndrlen ileti tanıtıcısı *MQGET*, *MQPUT* ya da *MQPUT1* çağrılarında kullanılabilir; ancak, aynı anda yalnızca bir IBM MQ çağrısı tarafından kullanılabilir. İkinci bir IBM MQ çağrısı aynı ileti tanıtıcısını kullanmayı denedięinde tanıtıcı kullanılırsa, ikinci IBM MQ çağrısı *RC2499* neden koduyla başarısız olur.

HCONN parametresi *HCUNAS* deęilse, dndrlen ileti tanıtıcısı yalnızca belirtilen baęlantıda kullanılabilir.

Bu ileti iřlecinin kullanıldıęı sonraki MQI çağrılarında aynı *HCONN* deęiřtirge deęeri kullanılmalıdır:

- *MQDLTMH*
- *MQSETMP*
- *MQINQMP*
- *MQDLTMP*
- *MQMHBUF*
- *MQBUFMH*

İleti tanıtıcısı için *MQDLTMH* çağrısı yayınlandıęında ya da tanıtıcı kapsamını tanımlayan iřlem birimi sona erdięinde, dndrlen ileti tanıtıcısı geęerli olmaktan çıkar. İleti tanıtıcısı yaratıldıęında belirli bir baęlantı saęlanırsa ve kuyruk yneticisine ynelik baęlantı geęerlilięini keserse (rneęin, *MQDBC* çağrılırsa) *MQDLTMH* rtk olarak çağrılır.

CMPCOD (10 basamaklı iřaretli tamsayı)-çıkıř

Tamamlanma kodu; ařaęıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı iřaretli tamsayı)-çıkıř

Neden kodu niteleyici *CMPCOD*.

CMPCOD **CCOK** ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD **CCFAIL** ise:

RC2204

(2204, X'089C') Baędařtırıcı kullanılmıyor.

RC2130

(2130, X'852 ') Baędařtırıcı hizmet modl yklenemiyor.

RC2157

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

RC2219

(2219, X'08AB') nceki arama tamamlanmadan nce girilen MQI çağrısı.

RC2461

(2461, X'099D') İleti tanıtıcısı yaratma seęenekleri yapısı geęersiz.

RC2273

(2273, X'7D9') Kuyruk yneticisiyle baęlantı kesildi.

RC2017

(2017, X'07E1') Bařka tanıtıcı yok.

RC2018

(2018, X'7E2') Baęlantı tanıtıcısı geęersiz.

RC2460

(2460, X'099C') İleti tanıtıcısı gstergesi geęersiz.

RC2046

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

Daha fazla ayrıntı için bkz. [“IBM i \(ILE RPG\) için dönüş kodları” sayfa 1402](#).

RPG Bildirimi

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQCRTMH(HCONN : CRTOPT : HMSG :
                          CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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 IBM i üzerinde MQCTL (Control callback)

MQCTL çağrısı, bir bağlantı için açılan nesne tanıttıcıları üzerinde denetleme işlemleri gerçekleştirir.

- [“Sözdizimi” sayfa 1263](#)
- [“Kullanım notları” sayfa 1263](#)
- [“Parametreler” sayfa 1263](#)
- [“RPG Bildirimi” sayfa 1268](#)

Sözdizimi

MQCTL (*Hconn, Operation, ControlOpts, CompCode, Reason*)

Kullanım notları

1. Geri çağırma yordamları, çağırdıkları tüm hizmetlerden gelen yanıtları denetlemelidir ve yordam çözülemeyen bir koşul saptarsa, geri çağırma yordamına yinelenen çağrılarını önlemek için bir MQCB (CBREG) komutu vermelidir.

Parametreler

MQCTL çağrısındaki değişirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bu tanıttıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *HCONN* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

OPERATN (10 basamaklı işaretli tamsayı)-giriş

Belirtilen nesne tanıttıcısı için tanımlanan geri çağırma sırasında işlenmekte olan işlem. Aşağıdaki seçeneklerden birini ve yalnızca birini belirtmeniz gerekir:

CTLSR

Belirtilen bağlantı tanıtıcısı için, tanımlı tüm ileti tüketicisi işlevleri için ileti tüketimini başlatın.

Geri çağırımlar, uygulama iş parçacıklarından farklı olan, sistem tarafından başlatılan bir iş parçacığında çalıştırılır.

Bu işlem, sisteme sağlanan bağlantı tanıtıcısının denetimini sağlar. Tüketici iş parçacığından başka bir iş parçacığı tarafından verilebilecek tek MQI çağrıları şunlardır:

- CTLSR İşlemiyle MQCTL
- CTLSU İşlemiyle MQCTL
- MQDISC-Bu, HConn bağlantısını kesmeden önce CTLSR işlemiyle MQCTL gerçekleştirir.

Bağlantı tanıtıcısı başlatılırken bir IBM MQ API çağrısı yayınlandıysa ve çağrı bir ileti tüketicisi işlevinden kaynaklanmıyorsa RC2500 döndürülür.

Bir bağlantı başarısız olursa, bu işlem mümkün olan en kısa sürede etkileşimi durdurur. Bu nedenle, ana iş parçacığında yayınlanmakta olan bir IBM MQ API çağrısı RC2500 dönüş kodunu bir süre için alabilir, ardından bağlantı durduruldu durumuna geri döndüğünde RC2009 dönüş kodu gelir.

Bu, bir tüketici işlevinde yayınlanabilir. Geri çağırma yordamına aynı bağlantı için tek amacı, önceden yayınlanan bir CTLSR işlemi iptal etmektir.

Uygulama, iş parçacıklı olmayan bir IBM MQ kitaplığıyla bağlıysa, bu seçenek desteklenmez.

CTLSW

Belirtilen bağlantı tanıtıcısı için, tanımlı tüm ileti tüketicisi işlevleri için ileti tüketimini başlatın.

İleti tüketicileri aynı iş parçacığında çalışır ve denetim, şu zamana kadar MQCTL ' yi çağırana döndürülmez:

- MQCTL CTLSR ya da CTLSU işlemlerinin kullanımıyla serbest bırakıldı ya da
- Tüm tüketici yordamlarının kaydı kaldırıldı ya da askıya alındı.

Tüm tüketicilerin kaydı kaldırılırsa ya da askıya alınırsa, örtük bir CTLSR işlemi yayınlanır.

Bu seçenek, yürürlükteki bağlantı tanıtıcısı ya da başka bir bağlantı tanıtıcısı için, bir geri çağırma yordamı içinden kullanılamaz. Arama girişiminde bulunulursa, RC2012 ile geri döner.

Bir CTLSW işlemi sırasında herhangi bir zamanda kayıtlı, askıya alınmamış tüketiciler yoksa, arama RC2446 neden koduyla başarısız olur.

Bir CTLSW işlemi sırasında bağlantı askıya alındıysa, MQCTL çağrısı RC2521; uyarı nedeni kodunu döndürür; bağlantı 'başlatıldı' olarak kalır.

Uygulama CTLSR ya da CTLSU yayını seçebilir. Bu durumda, CTLSU işlem blokları.

Bu seçenek, tek iş parçacıklı istemcide desteklenmez.

CTLP

İletilerin tüketilmesini durdurun ve bu seçenek tamamlanmadan önce tüm tüketicilerin işlemlerini tamamlamalarını bekleyin. Bu işlem, bağlantı tanıtıcısını serbest bırakır.

Bir geri çağırma yordamı içinden verilirse, yordam çıkıncaya kadar bu seçenek yürürlüğe girmez. Okunan iletilere ilişkin tüketici yordamları tamamlandıktan sonra ve (istenirse) geri çağırma yordamlarına yönelik durdurma çağrılarının ardından başka ileti tüketicisi rutinleri çağrılmaz.

Bir geri çağırma yordamı dışında verilirse, okunan iletilere ilişkin tüketici yordamları tamamlanıncaya kadar ve geri çağırımları durdurma çağrılarının (istendiyse) sonra, denetim çağırana dönmez. Ancak geri çağırımlar kayıtlı olmaya devam ediyor.

Bu işlevin önden okuma iletileri üzerinde etkisi yoktur. Tüketicilerin teslim edilecek başka ileti olup olmadığını saptamak için, geri çağırma işlevinden MQCLOSE (COQSC) çalıştırdığından emin olmanız gerekir.

CTLSU

İleti tüketimini duraklatır. Bu işlem, bağlantı tanıtıcısını serbest bırakır.

Bu, uygulamaya ilişkin iletilerin önceden okunmasını etkilemez. İletileri uzun bir süre kullanmayı durdurmak istiyorsanız, kuyruğu kapatmayı ve tüketimin devam etmesi gerekirken yeniden açmayı düşünün.

Bir geri çağırma yordamı içinden verilirse, yordam çıkıncaya kadar yürürlüğe girmez. Yürürlükteki yordam çıktıktan sonra başka ileti tüketicisi yordamı çağrılmayacak.

Bir geri çağırma dışında verilirse, denetim, yürürlükteki tüketici yordamı tamamlanıncaya ve başka çağrılınca kadar çağırana dönmez.

CTLRE

İletileri tüketmeye devam edin.

Bu seçenek genellikle ana uygulama iş parçacığından yayınlanır, ancak aynı yordamda daha önce yayınlanan bir askıya alma isteğini iptal etmek için bir geri çağırma yordamı içinden de kullanılabilir.

Bir CTLSW ' yi sürdürmek için CTLRE kullanılırsa, işlem blokları.

PCTLOP (MQCTLO)-giriş

MQCTL işlemini denetleyen seçenekler

Yapının ayrıntıları için bkz. MQCTLO .

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu; aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Aşağıdaki neden kodları, kuyruk yöneticisinin **Reason** parametresi için döndürebileceği kodlardır.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCFAIL ise:

RC2133

(2133, X'855 ') Veri dönüştürme hizmetleri modülleri yüklenemiyor.

RC2204

(2204, X'89C') Bağdaştırıcı kullanılamıyor.

RC2130

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

RC2374

(2374, X' 946 ') API çıkışı başarısız oldu.

RC2183

(2183, X'887 ') API çıkışı yüklenemiyor.

RC2157

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

RC2005

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

RC2487

(2487, X'9B7') Geri çağırma yordamı çağırılmıyor

RC2448

(2448, X' 990 ') Kayıtlı geri çağırma olmadığı için kayıt kaldırılamıyor, askıya alınamıyor ya da devam ettirilemiyor

RC2486

(2486, X'9B6') Bir CBREG çağırısında hem CallbackFunction , hem de CallbackName belirtildi ya da CallbackFunction ya da CallbackName belirtildi, ancak kayıtlı geri çağırma işleviyle eşleşmiyor.

RC2483

(2483, X'9B3') CallBackTip alanı yanlış.

RC2219

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağırısı.

RC2444

(2444, X'98C') Seçenek öbeği yanlış.

RC2484

(2484, X'9B4') Yanlış MQCBD seçenekleri alanı.

RC2140

(2140, X'85C') Bekleme isteği CICStarafından reddedildi.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2217

(2217, X'8A9') Bağlantı için yetkili değil.

RC2202

(2202, X'89A') Bağlantı susturuyor.

RC2203

(2203, X'89B') Bağlantı kapatılıyor.

RC2207

(2207, X'89F') İlinti-tanıtıcı hatası.

RC2016

(2016, X'7E0') Kuyruk için engellendi.

RC2351

(2351, X'92F') Genel iş birimleri çakışması.

RC2186

(2186, X'88A') Get-message seçenekleri yapısı geçersiz.

RC2353

(2353, X' 931 ') Genel iş birimi için kullanımda.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2019

(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

RC2259

(2259, X'8D3') Tutarsız göz atma belirtimi.

RC2245

(2245, X'8C5') İş birimi belirtimi tutarsız.

RC2246

(2246, X'8C6') İmleç altındaki ileti alma için geçerli değil.

RC2352

(2352, X' 930 ') Genel iş birimi yerel iş birimiyle çakışıyor.

RC2247

(2247, X'8C7') Eşleştirme seçenekleri geçerli değil.

- RC2485**
(2485, X'9B5') Yanlıř MaxMsgUzunluk alanı
- RC2026**
(2026, X'7EA') İleti tanımlayıcısı geçersiz.
- RC2497**
(2497, X'9C1') Belirtilen işlev giriş noktası modülde bulunamadı.
- RC2496**
(2496, X'9C0') Modül bulundu, ancak yanlıř tipte (32 bit ya da 64 bit) ya da geçerli bir dll deęil.
- RC2495**
(2495, X'9BF') Modül arama yolunda bulunamadı ya da yükleme yetkisi yok.
- RC2206**
(2206, X'89E') İleti tanıtıcısı hatası.
- RC2250**
(2250, X'8CA') İleti sıra numarası geçersiz.
- RC2331**
(2331, X'91B') İleti simgesi kullanımı geçersiz.
- RC2036**
(2036, X'7F4') Kuyruk göz atmak için açık deęil.
- RC2037**
(2037, X'7F5') Kuyruk giriş için açık deęil.
- RC2041**
(2041, X'7F9') Nesne tanımı açıldıktan sonra deęiřtirildi.
- RC2101**
(2101, X'835 ') Nesne zarar gördü.
- RC2488**
(2488, X'9B8') API Çaęırısı 'nda İşlem Kodu Yanlıř
- RC2046**
(2046, X'7FE') Seçenekler geçerli deęil ya da tutarlı deęil.
- RC2193**
(2193, X'891 ') Sayfa kümesi veri kümesine eriřilirken hata oluřtu.
- RC2052**
(2052, X'804 ') Kuyruk silindi.
- RC2394**
(2394, X'95A') Kuyruk yanlıř dizin tipine sahip.
- RC2058**
(2058, X'80A') Kuyruk yöneticisi adı geçerli deęil ya da bilinmiyor.
- RC2059**
(2059, X'80B') Kuyruk yöneticisi baęlantı için kullanılamıyor.
- RC2161**
(2161, X'871 ') Kuyruk yöneticisi susturuyor.
- RC2162**
(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.
- RC2102**
(2102, X'836 ') Sistem kaynakları yetersiz.
- RC2069**
(2069, X'815 ') Bu tutamaç için bekleyen işaret.
- RC2071**
(2071, X'817 ') Kullanılabilir bellek yetersiz.
- RC2109**
(2109, X'83D') Çaęrı çıkıř programı tarafından engellendi.

RC2072

(2072, X'818 ') Syncpoint desteği kullanılamıyor.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RC2354

(2354, X' 932 ') Genel iş birimine kayıt başarısız oldu.

RC2355

(2355, X' 933 ') İş birimi çağrılarının karışımı desteklenmez.

RC2255

(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.

RC2090

(2090, X'82A') MQGMO' daki bekleme aralığı geçersiz.

RC2256

(2256, X'8D0') Yanlış MQGMO sürümü sağlandı.

RC2257

(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

RC2298

(2298, X'8FA') İstenen işlem yürürlükteki ortamda kullanılamıyor.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C                               CALLP      MQCTL(HCONN : OPERATN : PCTLOP :
                                       CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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**IBM i üzerinde MQDISC (Kuyruk yöneticisinin bağlantısını kes)**

MQDISC çağrısı, kuyruk yöneticisi ile uygulama programı arasındaki bağlantıyı kesiyor ve MQCONN ya da MQCONNX çağrısının tersi.

- [“Sözdizimi” sayfa 1268](#)
- [“Kullanım notları” sayfa 1269](#)
- [“Parametreler” sayfa 1269](#)
- [“RPG Bildirimi” sayfa 1270](#)

Sözdizimi

MQDISC (HCONN, CMPCOD, REASON)

Kullanım notları

1. Uygulamada açık nesnelere varken bir MQDISC çağrısı verilirse, bu nesnelere kuyruk yöneticisi tarafından kapatılır ve kapatma seçenekleri CONONE olarak ayarlanır.
2. Uygulama bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması uygulamanın nasıl sona erdiğine bağlıdır:
 - a. Uygulama, sona erdirmeden önce MQDISC çağrısı yaparsa:
 - Kuyruk yöneticisi eşgüdümlü bir iş birimi için, kuyruk yöneticisi uygulama adına MQCMIT çağrısı yayınlar. Mümkünse iş birimi kesinleştirildi ve kesinleştirilmediyse geri çekildi.
 - Dışarıdan eşgüdümlü bir iş biriminde, iş biriminin durumunda herhangi bir değişiklik yoktur; ancak, kuyruk yöneticisi, iş birimi eşgüdümçüsü tarafından istendiğinde, iş biriminin kesinleştirilmesi gerektiğini belirtir.
 - b. Uygulama olağan bir şekilde sona erer, ancak MQDISC çağrısı yayınlanmazsa, iş birimi geriletir.
 - c. Uygulama MQDISC çağrısı yayınlanmadan *olağandışı şekilde* sona ererse, iş birimi geriletir.

Parametreler

MQDISC çağrısındaki değişirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş/çıkış

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *HCONN* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

Çağrı başarıyla tamamlandığında, kuyruk yöneticisi *HCONN* değerini ortam için geçerli bir tanıtıcı olmayan bir değere ayarlar. Bu değer:

HCUNUH.

Kullanılamaz bağlantı tanıtıcısı.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCFail ise:

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2058

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

RC2059

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C                               CALLP      MQDISC(HCONN : CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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**IBM i üzerinde MQDLTMH (İleti tanıtıcısını sil)**

MQDLTMH çağrısı bir ileti tanıtıcısını siler ve MQCRTMH çağrısının tersidir.

- “Sözdizimi” sayfa 1270
- “Kullanım notları” sayfa 1270
- “Parametreler” sayfa 1272
- “RPG Bildirimi” sayfa 1273

Sözdizimi

MQDLTMH ((*Hconn*, *Hmsg*, *DltMsgHOpts*, *CompCode*, *Reason*))

Kullanım notları

1. Bu çağrıyı yalnızca kuyruk yöneticisinin kendisi iş birimini koordine ettiğinde kullanabilirsiniz. Bu şöyle olabilir:
 - Değişikliklerin yalnızca IBM MQ kaynaklarını etkilediği yerel bir iş birimi.
 - Değişikliklerin diğer kaynak yöneticilerine ait kaynakları etkileyebileceği ve IBM MQ kaynaklarını etkileyebileceği genel bir iş birimi.Yerel ve genel iş birimleriyle ilgili daha fazla ayrıntı için bkz. [“MQBEGIN \(İş birimini başlat\)- IBM i” sayfa 1233.](#)
2. Kuyruk yöneticisinin iş birimini koordine etmediği ortamlarda, MQBACK yerine uygun geri alma çağrılarını kullanın. Ortam, uygulamanın olağandışı sonlanmasından kaynaklanan örtük bir geri dönüşü de destekleyebilir.

- z/OS' da aşağıdaki çağrılarını kullanın:
 - İş birimi yalnızca IBM MQ kaynaklarını etkiliyorsa, toplu iş programları (IMS toplu DL/I programları da içinde olmak üzere) MQBACK çağrılarını kullanabilir. Ancak, iş birimi hem IBM MQ kaynaklarını, hem de diğer kaynak yöneticilerine ait kaynakları (örneğin, Db2) etkiliyorsa, z/OS Recoverable Resource Service (RRS) tarafından sağlanan SRRBACK çağrısına bakın. SRRBACK çağrısı, RRS koordinasyonu için etkinleştirilen kaynak yöneticilerine ait kaynaklarda yapılan değişiklikleri geri çevirir.
 - CICS uygulamaları, iş birimini geri almak için EXEC CICS SYNCPOINT ROLLBACK komutunu kullanmalıdır. CICS uygulamaları için MQBACK çağrılarını kullanmayın.
 - IMS uygulamaları (toplu DL/I programlarından başka), iş birimini geri almak için ROLB gibi IMS çağrılarını kullanmalıdır. IMS uygulamaları (toplu DL/I programları dışında) için MQBACK çağrılarını kullanmayın.
- IBM işletim sistemi üzerinde, kuyruk yöneticisi tarafından eşgüdümlü yerel iş birimleri için bu çağrıyı kullanın. Bu, iş düzeyinde bir kesinleştirme tanımının var olmaması gerektiği, yani iş için **CMTSCOPE (*JOB)** parametresini içeren STRCMTCTL komutunun yayınlanmamış olması gerektiği anlamına gelir.

3. Bir uygulama, bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması uygulamanın olağan şekilde mi, yoksa olağandışı olarak mı bittiğine bağlıdır. Daha fazla ayrıntı için [“IBM i üzerinde MQDISC \(Kuyruk yöneticisinin bağlantısını kes\)” sayfa 1268](#) içindeki kullanım notlarına bakın.

4. Bir uygulama, mantıksal ileti gruplarına ya da bölümlerine ileti girdiğinde ya da aldığı anda, kuyruk yöneticisi, son başarılı MQPUT ve MQGET çağrılarına ilişkin ileti grubu ve mantıksal iletiyle ilgili bilgileri saklar. Bu bilgiler kuyruk tanıtıcısıyla ilişkilidir ve aşağıdaki gibi şeyleri içerir:

- MQMD ' deki *GroupId*, *MsgSeqNumber*, *Offset* ve *MsgFlags* alanlarının değerleri.
- İletinin bir iş biriminin parçası olup olmadığı.
- MQPUT çağrısı için: İletinin kalıcı mı, kalıcı mı olduğu.

Kuyruk yöneticisi, aşağıdakilerin her biri için bir küme olmak üzere üç grup ve bölüm bilgisi kümesini tutar:

- Son başarılı MQPUT çağrısı (bu, bir iş biriminin parçası olabilir).
- Kuyruktan bir iletiyi kaldıran son başarılı MQGET çağrısı (bu, bir iş biriminin parçası olabilir).
- Kuyruktaki bir iletiye göz atan son başarılı MQGET çağrısı (bu, bir iş biriminin parçası olamaz).

Uygulama iletileri bir iş biriminin parçası olarak yerleştirir ya da alırsa ve uygulama iş birimini geri çevirirse, grup ve bölüm bilgileri daha önce sahip olduğu değere geri yüklenir:

- MQPUT çağrısıyla ilişkili bilgiler, yürürlükteki iş biriminde o kuyruk tanıtıcısı için başarılı olan ilk MQPUT çağrıdan önce sahip olduğu değere geri yüklenir.
- MQGET çağrısıyla ilişkili bilgiler, yürürlükteki iş biriminde o kuyruk tanıtıcısı için ilk başarılı MQGET çağrıdan önce sahip olduğu değere geri yüklenir.

İş birimi başlatıldıktan sonra uygulama tarafından güncellenen, ancak iş birimi kapsamı dışında kalan kuyruklarda, iş birimi getirilmediyse, grup ve bölüm bilgileri geri yüklenmez.

Bir iş birimi getirildiğinde grup ve bölüm bilgilerinin önceki değerine geri yüklenmesi, uygulamanın çeşitli iş birimlerine birçok kesim içeren büyük bir ileti grubu ya da büyük bir mantıksal ileti yaymasına ve iş birimlerinden birinin başarısız olması durumunda ileti grubunda ya da mantıksal iletide doğru noktada yeniden başlamasına olanak sağlar. Yerel kuyruk yöneticisinde yalnızca sınırlı kuyruk saklama alanı varsa, birkaç iş birimi kullanılması avantajlı olabilir. Ancak, bir sistem arızası oluşursa, uygulamanın iletileri doğru noktada yeniden koymayı ya da almayı yeniden başlatabilmesi için yeterli bilgiyi koruması gerekir.

Bir sistem hatasından sonra doğru noktada nasıl yeniden başlatılacağına ilişkin ayrıntılar için [PMOPT \(10 basamaklı imzalı tamsayı\)](#) içinde açıklanan PMLOGO seçeneğine ve [GMOPT \(10 basamaklı işaretli tamsayı\)](#) içinde açıklanan GMLOGO seçeneğine bakın.

Geri kalan kullanım notları yalnızca kuyruk yöneticisi iş birimlerini koordine ettiğinde geçerlidir:

5. Bir iş birimi, bağlantı tanıtıcısıyla aynı kapsama sahiptir. Belirli bir iş birimini etkileyen tüm IBM MQ çağrılarının aynı bağlantı tanıtıcısı kullanılarak gerçekleştirilmesi gerekir. Farklı bir bağlantı tanıtıcısı kullanılarak verilen çağrılar (örneğin, başka bir uygulama tarafından verilen çağrılar) farklı bir iş birimini etkiler. Bağlantı tanıtıcılarının kapsamına ilişkin bilgi için bkz. [HCONN \(10 basamaklı işaretli tamsayı\)-çıkış](#) .
6. Yalnızca yürürlükteki iş biriminin bir parçası olarak konan ya da alınan iletiler bu çağrıdan etkilenir.
7. Bir iş birimi içinde MQGET, MQPUT ya da MQPUT1 çağrılarını veren, ancak hiçbir zaman kesinleştirme ya da geriletme çağrısı yapmayan uzun süreli bir uygulama, kuyrukları diğer uygulamalar tarafından kullanılmayan iletilerle doldurabilir. Bu olasılığa karşı koruma sağlamak için, denetimcinin **MaxUncommittedMsgs** kuyruk yöneticisi özneliğini, kuyrukları dolduran, ancak beklenen ileti sistemi uygulamalarının düzgün çalışmasını sağlayacak kadar yüksek olan kaçak uygulamaları önleyecek kadar düşük bir değere ayarlaması gerekir.

Parametreler

MQDLTMH çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriř

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir.

Deęer, **HMSG** deęiřtirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eřleřmelidir.

İleti tanıtıcısı HCUNAS kullanılarak yaratıldıysa, ileti tanıtıcısını silen iş parçacığında geçerli bir bağlantı kurulmalıdır; tersi durumda, çağrı RC2009 ile başarısız olur.

HMSG (20 basamaklı işaretli tamsayı)-giriř/çıkıř

Bu, silinecek ileti tanıtıcısıdır. Deęer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

Çaęrı başarıyla tamamlandığında, tanıtıcı ortam için geçersiz bir değere ayarlanır. Bu deęer:

Hmunuh.

Kullanılmayan ileti tanıtıcısı.

Aynı ileti tanıtıcısı iletilen başka bir IBM MQ çağrısı devam ediyorsa, ileti tanıtıcısı silinemez.

DLTOPT (MQDMHO)-giriř

Ayrıntılar için bkz. [MQDMHO](#) .

CMPCOD (10 basamaklı işaretli tamsayı)-çıkıř

Tamamlanma kodu; ařaęıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkıř

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCFAIL ise:

RC2204

(2204, X'089C') Baędařtırıcı kullanılmıyor.

RC2130

(2130, X'852 ') Baędařtırıcı hizmet modülü yüklenemiyor.

RC2157

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

RC2219

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağırısı.

RC2009

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2462

(2462, X'099E') Silme iletisi tanıtıcı seçenekleri yapısı geçerli değil.

RC2460

(2460, X'099C') İleti tanıtıcısı göstergesi geçersiz.

RC2499

(2499, X'09C3') İleti tanıtıcısı kullanımda.

RC2046

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

Daha fazla ayrıntı için bkz. [“IBM i \(ILE RPG\) için dönüş kodları” sayfa 1402 .](#)

RPG Bildirimi

```
C*.1.....2.....3.....4.....5.....6.....7..
C                CALLP      MQDLTMH(HCONN : HMSG : DLTOPT :
                          CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
DMQDLTMH          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-İleti özelliğini sil

MQDLTMP çağırısı bir özelliği ileti tanıtıcısından siler ve MQSETMP çağırısının tersidir.

- [“Sözdizimi” sayfa 1273](#)
- [“Parametreler” sayfa 1274](#)
- [“RPG Bildirimi” sayfa 1275](#)

Sözdizimi

MQDLTMP (*Hconn*, *Hmsg*, *DltPropOpts*, *Name*, *CompCode*, *Reason*)

Parametreler

MQDLTMP çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı iřaretli tamsayı)-Giriř

Bu tanıtıcı, kuyruk yöneticisine yönelik baęlantıyı gösterir. Deęer, **HMSG** deęiřtirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan baęlantı tanıtıcısıyla eřleřmelidir.

İleti tanıtıcısı HCUNAS kullanılarak yaratıldıysa, ileti tanıtıcısını silen iř parçacıęında geęerli bir baęlantı kurulmalıdır; tersi durumda, çağrı RC2009ile başarısız olur.

HMSG (20 basamaklı iřaretli tamsayı)-giriř

Bu, silinecek özellięi içeren ileti tanıtıcısıdır. Deęer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

DLTOPT (MQDMPO)-Giriř

Ayrıntılar için [MQDMPO](#) veri tipine bakın.

PRNAME (MQCHARV)-giriř

Silinecek özellięin adı. Özellik adlarıyla ilgili ek bilgi için [Özellik adları](#) konusuna bakın.

Özellik adında genel arama karakterlerine izin verilmez.

CMPCOD (10 basamaklı iřaretli tamsayı)-çıkıř

Tamamlanma kodu; ařaęıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı iřaretli tamsayı)-çıkıř

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2471

(2471, X'09A7') Özellik kullanılmıyor.

RC2421

(2421, X'0975 ') Özellikleri içeren bir MQRFH2 klasörü ayrıştırılmadı.

CMPCOD CCFail ise:

RC2204

(2204, X'089C') Baędařtırıcı kullanılmıyor.

RC2130

(2130, X'0852 ') Baędařtırıcı hizmet modülü yüklenemiyor.

RC2157

(2157, X'086D') Birincil ve ana ASID' ler farklıdır.

RC2219

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

RC2009

(2009, X'07D9') Kuyruk yöneticisine baęlantı kaybedildi.

RC2481

(2481, X'09B1') İleti silme özelliği seçenekleri yapısı geçersiz.

RC2460

(2460, X'099C') İleti tanıtıcısı geçersiz.

RC2499

(2499, X'09C3') İleti tanıtıcısı kullanımda.

RC2046

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2442

(2442, X'098A') Özellik adı geçersiz.

RC2111

(2111, X'083F') Özellik adı kodlanmış karakter takımı tanıtıcısı geçersiz.

RC2195

(2195, X'0893 ') Beklenmeyen hata oluştu.

Bu kodlarla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

RPG Bildirimi

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQDLTMP(HCONN : HMSG : DLTOPT :
                      PRNAME : CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
DMQDLTMP          PR          EXTPROC('MQDLTMP')
D* Connection handle
D HCONN           10I 0 VALUE
D* Message handle
D HMSG           20I 0 VALUE
D* Options that control the action 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
```

IBM i IBM i üzerinde MQGET (İletiyi al)

MQGET çağırısı, MQOPEN çağırısı kullanılarak açılan bir yerel kuyruktan ileti alır.

- [“Sözdizimi” sayfa 1275](#)
- [“Kullanım notları” sayfa 1275](#)
- [“Parametreler” sayfa 1278](#)
- [“RPG Bildirimi” sayfa 1283](#)

Sözdizimi

MQGet (*HCONN, HOBJ, MSGDSC, GMO, BUFLN, BUFFER, DATLEN, CMPCOD, REASON*)

Kullanım notları

1. Alınan ileti olağan durumda kuyruktan silinir. Bu silme, MQGET çağırısının bir parçası olarak ya da bir eşitleme noktasının parçası olarak gerçekleştirilebilir. **GMO** değiştirilmesinde bir GMBRWF ya

da GMBRWN seçeneği belirtilirse ileti silme işlemi gerçekleşmez ("IBM i üzerinde MQGMO (Get-message seçenekleri)" sayfa 1060 başlıklı konuda açıklanan *GMOPT* alanına bakın).

2. GMLK seçeneği göz atma seçeneklerinden biriyle belirtilirse, göz atılan ileti yalnızca bu tanıtıcı tarafından görülebilecek şekilde kilitletir.

GMUNLK seçeneği belirtilirse, önceden kilitletmiş bir iletinin kilidi açılır. Bu durumda ileti alınmaz ve **MSGDSC**, **BUFLN**, **BUFFER** ve **DATLEN** parametreleri denetlenmez ya da değiştirilmez.

3. MQGET çağrısı yayınlayan uygulama bir IBM MQ MQI clientolarak çalışıyorsa, MQGET çağrısının işlenmesi sırasında IBM MQ MQI client olağandışı bir şekilde sona ererse ya da istemci bağlantısı kesilirse, alınan ileti kaybedilebilir. Bunun nedeni, kuyruk yöneticisinin altyapısında çalışan vekil ve istemci adına MQGET çağrısının verilmesinin, vekil iletiyi istemciye döndürmek üzere oluncaya kadar istemcinin kaybını saptayamamasından kaynaklıdır; bu, ileti kuyruktan kaldırıldıktan sonra ortaya çıkar. Bu, hem kalıcı iletiler hem de kalıcı olmayan iletiler için ortaya çıkabilir.

Bu şekilde ileti kaybetme riski, iş birimleri içinde her zaman ileti alınmasıyla (yani, MQGET çağrısında GMSYP seçeneği belirtilerek ve ileti işlenirken iş birimini kesinleştirmek ya da geri almak için MQCMIT ya da MQBACK çağrıları kullanılarak) ortadan kaldırılabilir. GMSYP belirtilirse ve istemci olağandışı bir şekilde sonlandırılırsa ya da bağlantı kesilirse, vekil, kuyruk yöneticisindeki iş birimini geri alır ve ileti kuyrukta yeniden yürürlüğe konur.

Prensip olarak, kuyruk yöneticisinin altyapısında çalışan uygulamalarla aynı durum ortaya çıkabilir, ancak bu durumda bir iletinin kaybolabileceği pencere küçüktür. Ancak, IBM MQ MQI clients ' de olduğu gibi, iş birimi içindeki ileti alınarak risk ortadan kaldırılabilir.

4. Bir uygulama belirli bir iletiye bir dizi ileti koyarsakuyruğun tek bir iş birimi içinde olması ve daha sonra bu iş birimini başarıyla kesinleştirmesi, iletilerin aşağıdaki gibi alınabilmesine neden olur:

- Kuyruk *paylaşılmayan bir kuyruksa* (yerel kuyruk), iş birimindeki tüm iletiler aynı anda kullanılabilir olur.
- Kuyruk bir *paylaşılan kuyruksa*, iş birimi içindeki iletiler yerleştirildikleri sırayla kullanılabilir, ancak tümü aynı anda kullanılamaz. Sistem ağır biçimde yüklendiğinde, iş birimindeki ilk iletinin başarıyla alınması mümkündür; ancak, iş birimindeki ikinci ya da sonraki iletinin RC2033ile başarısız olması için MQGET çağrısı gerçekleştirilebilir. Bu durumda, uygulamanın kısa bir süre beklemesi ve işlemi yeniden denemesi gerekir.

5. Bir uygulama ileti gruplarını kullanmadan aynı kuyruğa ileti dizisi koyarsa, belirli koşullar yerine getirilirse, bu iletilerin sırası korunur. Ayrıntılar için MQPUT çağrısının tanımındaki kullanım notlarına bakın. Koşullar karşılanıyorsa, iletiler alıcı uygulamaya gönderildikleri sırayla sunulur. Bu durumda:

- Kuyruktan ileti alan tek bir günlük nesnesi var.

Kuyruktan ileti alan iki ya da daha fazla uygulama varsa, bir sıraya ait iletileri tanımlamak için kullanılacak mekanizmayı gönderen ile aynı fikirde olmaları gerekir. Örneğin, gönderen iletilerdeki tüm MDCID alanlarını bir sırayla, bu ileti sırası için benzersiz olan bir değere ayarlayabilir.

- Alıcı, örneğin belirli bir MDMID ya da MDCIDbelirterek, alma sırasını kasıtlı olarak değiştirmez.

Gönderen uygulama iletileri ileti grubu olarak yerleştirirse, alan uygulama MQGET çağrısında GMLOGO seçeneğini belirtiyorsa, iletiler alan uygulamaya doğru sırayla sunulur. İleti gruplarıyla ilgili ek bilgi için aşağıdaki konuya bakın:

- MQMD ' deki MDMFL alanı
- MQPMO içindeki PMLOGO seçeneği
- MQGMO ' da GMLOGO seçeneği

6. Uygulamalar, **MSGDSC** değiştirgesinin MDFB alanında FBQUIT geribildirim kodunu sınavabilirler. Bu değer bulunursa, uygulama sona erer. Ek bilgi için "IBM i üzerinde MQMD (İleti tanımlayıcı)" sayfa 1093 içinde açıklanan MDFB alanına bakın.

7. HOBJ ile tanıtilan kuyruk OOSAVA seçeneğiyle açıldıysa ve MQGET çağrısındaki tamamlanma kodu CCOK ya da CCWARN ise, HOBJ kuyruk tanıtıcısıyla ilişkili bağlam, alınan iletinin bağlamına ayarlanır (GMBRWF ya da GMBRWV seçeneği ayarlanmadıysa, bu durumda bağlam kullanılamıyor olarak işaretlenir). Bu bağlam, PMPASI ya da PMPASA seçenekleri belirtilerek sonraki bir MQPUT ya da

MQPUT1 çağrısında kullanılabilir. Bu, alınan iletinin bağlamının tamamen ya da kısmen başka bir iletiye aktarılmasını sağlar (örneğin, ileti başka bir kuyruğa iletildiğinde). İleti bağlamıyla ilgili daha fazla bilgi için İleti bağlamı ve Bağlam bilgilerini denetleme başlıklı konuya bakın.

8. **GMO** değiştirgesinde GMCONV seçeneği varsa, veriler **BUFFER** değiştirgesine yerleştirilmeden önce, uygulama iletisi verileri alıcı uygulama tarafından istenen gösterimine dönüştürülür:

- İletideki denetim bilgilerindeki MDFMT alanı, uygulama verilerinin yapısını tanımlar ve iletideki denetim bilgilerindeki MDCSI ve MDENC alanları, karakter kümesi tanıtıcısını ve kodlamasını belirtir.
- MQGET çağrısını yayınlayan uygulama, **MSGDSC** değiştirgesindeki MDCSI ve MDENC alanlarında, uygulama ileti verilerinin dönüştürülmesi gereken karakter kümesi tanıtıcısını ve kodlamasını belirtir.

İleti verilerinin dönüştürülmesi gerektiğinde, dönüştürme, iletideki denetim bilgisindeki MDFMT alanının değerine bağlı olarak, kuyruk yöneticisi tarafından ya da kullanıcı tarafından yazılan bir çıkış tarafından gerçekleştirilir:

- Aşağıdaki biçimler kuyruk yöneticisi tarafından otomatik olarak dönüştürülür; bu biçimlere "yerleşik" biçimler denir:

FMADMN	FMMDE
FMCICS	FMPCF
FMCM1	FMRMH
FMCM2	FMRFH
FMDLH	FMRFH2
FMDH	FMSTR
FMEVNT	FMTM
FMIMS	FMXQH
FMIMVS	

- FMNONE biçim adı, iletideki verilerin niteliğini tanımsız olarak gösteren özel bir değerdir. Sonuç olarak, kuyruk yöneticisi, ileti kuyruktan alındığında dönüştürme girişiminde bulunmaz.

Not: MQGET çağrısında FMNONE biçim adına sahip bir ileti için GMCONV belirtilirse ve iletinin karakter kümesi ya da kodlaması **MSGDSC** değiştirgesinde belirtilenden farklıysa, ileti **BUFFER** değiştirgesinde döndürülür (başka hata olmadığı varsayılarak), ancak çağrı tamamlanma kodu CCWARN ve neden kodu RC2110 ile tamamlanır.

FMNONE, ileti verilerinin niteliği, dönüştürme gerektirmediği anlamına geldiğinde ya da gönderen ve alan uygulamalar, ileti verilerinin gönderilmesi gereken formu kendi aralarında kararlaştırdığında kullanılabilir.

- Diğer tüm biçim adları, iletinin dönüştürme için kullanıcı tarafından yazılan bir çıkışa geçirilmesine neden olur. Çıkış, ortama özgü eklemeler dışında, biçimle aynı ada sahiptir. Kullanıcı tarafından belirtilen biçim adları "MQ" harfleriyle başlamamalıdır; bu tür adlar gelecekte desteklenen biçim adlarıyla çakışabilir.

İletideki kullanıcı verileri, desteklenen karakter kümeleri ve kodlamalar arasında dönüştürülebilir. Ancak, ileti bir ya da daha çok IBM MQ üstbilgi yapısı içeriyorsa, iletinin kuyruk adlarında geçerli olan herhangi bir karakter için çift baytlık ya da çok baytlık karakter içeren bir karakter kümesine dönüştürülemeyeceğini unutmayın. Neden kodu RC2111 ya da RC2115 denenirse ve ileti dönüştürülmezse döndürülür. Unicode karakter kümesi UTF-16, böyle bir karakter kümesinin bir örneğidir.

MQGET döndürülüşünde, aşağıdaki neden kodu iletinin başarıyla dönüştürüldüğünü gösterir:

- RCNONE

Aşağıdaki neden kodu, iletinin başarıyla dönüştürülmüş olabileceğini gösterir; uygulamanın öğrenmek için **MSGDSC** değiştirgesindeki MDCSI ve MDENC alanlarını denetlemesi gerekir:

- RC2079

Diğer tüm neden kodları iletinin dönüştürülmediğini gösterir.

Not: Bu örnekte açıklanan neden kodunun yorumlanması, yalnızca çıkış işleme yönergelerine uyduysa, kullanıcı tarafından yazılan çıkışlar tarafından gerçekleştirilen dönüştürmeler için geçerlidir.

9. Daha önce listelenen yerleşik biçimler için, kuyruk yöneticisi, GMCONV seçeneği belirtildiğinde iletideki karakter dizgilerinin varsayılan dönüşümünü gerçekleştirebilir. Varsayılan dönüştürme, kuyruk yöneticisinin dizgi verilerini dönüştürürken gerçek karakter kümesine yaklaşan, kuruluş tarafından belirtilen bir varsayılan karakter kümesi kullanmasını sağlar. Sonuç olarak, MQGET çağrısı CCWARN ve neden kodu RC2111 ya da RC2115 ile tamamlanmak yerine, tamamlanma kodu CCOK ile başarılı olabilir.

Not: Dizgi verilerini dönüştürmek için yaklaşık bir karakter kümesi kullanılmasının sonucu, bazı karakterlerin yanlış dönüştürülebilmesinden kaynaklanır. Dizgide yalnızca gerçek karakter kümesi ve varsayılan karakter kümesi için ortak olan karakterler kullanılarak bu önlenir.

Varsayılan dönüştürme hem uygulama iletisi verileri hem de MQMD ve MQMDE yapılarındaki karakter alanları için geçerlidir:

- Uygulama iletisi verilerinin varsayılan dönüşümü yalnızca aşağıdaki deyimlerin tümü true olduğunda gerçekleşir:
 - Uygulama GMCONV belirtiyor.
 - İleti, desteklenmeyen bir karakter kümesinden ya da karakter kümesine dönüştürülmesi gereken veriler içeriyor.
 - Kuyruk yöneticisi kurulduğunda ya da yeniden başlatıldığında varsayılan dönüştürme etkinleştirildi.
 - Kuyruk yöneticisi için varsayılan dönüştürme etkinleştirildiyse, MQMD ve MQMDE yapılarındaki karakter alanlarının varsayılan dönüşümü gerektiği gibi gerçekleşir. GMCONV seçeneği MQGET çağrısında uygulama tarafından belirtilmemiş olsa da dönüştürme gerçekleştirilir.
10. RPG programlama örneğinde gösterilen **BUFFER** parametresi bir dizgi olarak bildirildi; bu, parametrenin uzunluk üst sınırını 256 baytla sınırlar. Daha büyük bir arabellek gerekiyorsa, parametre bir yapı olarak ya da fiziksel bir dosyada bir alan olarak bildirilmelidir.
- Değiştirgenin yapı olarak bildirilmesi mümkün olan uzunluk üst sınırını 9999 bayta yükseltirken, değiştirgenin fiziksel bir dosyada alan olarak bildirilmesi olası uzunluk üst sınırını yaklaşık 32 KB 'ye yükseltir.

Parametreler

MQGET çağrısındaki değiştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. HCONN değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

HOBJ (10 basamaklı işaretli tamsayı)-giriş

Nesne tanıtıcısı.

Bu tanıtıcı, iletinin alınacağı kuyruğu gösterir. HOBJ değeri önceki bir MQOPEN çağrısı tarafından döndürüldü. Kuyruk aşağıdaki seçeneklerden biriyle ya da daha fazlasıyla açılmış olmalıdır (ayrıntılar için bkz. [“IBM i üzerinde MQOPEN \(nesneyi aç\)” sayfa 1299](#)):

- OOINPS
- OOINPX
- OOINPQ
- OOBROW

MSGDSC (MQMD)-giriş/çıkış

İleti tanımlayıcı.

Bu yapı, gerekli iletinin özniteliklerini ve alınan iletinin özniteliklerini açıklar. Ayrıntılar için bkz. [“IBM i üzerinde MQMD \(İleti tanımlayıcı\)” sayfa 1093.](#)

BUFLEN ileti uzunluğundan azsa, MSGDSC kuyruk yöneticisi tarafından, **GMO** parametresinde GMATM belirtilip belirtilmediğine bakılmaksızın girilir ([“IBM i üzerinde MQGMO \(Get-message seçenekleri\)” sayfa 1060](#) içinde açıklanan GMOPT alanına bakın).

Uygulama bir version-1 MQMD sağlarsa, döndürülen iletide uygulama ileti verilerinin başına bir MQMDE öneki eklenmiş, ancak yalnızca MQMDE 'deki alanlardan birinin ya da daha fazlasının varsayılan olmayan bir değeri varsa. MQMDE içindeki tüm alanların varsayılan değerleri varsa, MQMDE atlanır. MQMD 'deki MDFMT alanındaki FMMDE biçim adı, bir MQMDE' nin var olduğunu gösteriyor.

GMO (MQGMO)-giriş/çıkış

MQGET işlemini denetleyen seçenekler.

Ayrıntılar için bkz. [“IBM i üzerinde MQGMO \(Get-message seçenekleri\)” sayfa 1060.](#)

BUFLEN (10 basamaklı işaretli tamsayı)-giriş

BUFFER alanının bayt cinsinden uzunluğu.

Veri olmayan iletiler için sıfır belirtilebilir ya da ileti kuyruktan ve atılan veriler kaldırılacaksa (bu durumda GMATM belirtilmelidir).

Not: Kuyruktan okunabilecek en uzun iletinin uzunluğu **MaxMsgLength** kuyruk özniteliği tarafından verilir; bkz. [“Kuyruklara ilişkin öznitelikler” sayfa 1345.](#)

BUFFER (1 baytlık bit dizilimi x BUFLEN)-çıkış

İleti verilerini içerecek alan.

Arabellek, iletideki verilerin niteliğine uygun bir sınırdaki hizalanmalıdır. 4 baytlık hizalama, çoğu ileti için uygun olmalıdır (IBM MQ üstbilgi yapılarını içeren iletiler de içinde olmak üzere), ancak bazı iletiler daha sıkı hizalama gerektirebilir. Örneğin, 64 bitlik ikili tamsayı içeren bir ileti 8 baytlık hizalama gerektirebilir.

BUFLEN ileti uzunluğundan azsa, iletinin çoğu BUFFER içine taşınır; Bu, GMATM 'nin **GMO** parametresinde belirtilmiş olup olmadığını belirtir (daha fazla bilgi için bkz. GMOPT alanı [“IBM i üzerinde MQGMO \(Get-message seçenekleri\)” sayfa 1060](#) içinde açıklanmıştır).

BUFFER içindeki verilerin karakter kümesi ve kodlaması, **MSGDSC** parametresinde döndürülen MDCSI ve MDENC alanları tarafından verilir. Bu değerler alıcının gerektirdiği değerlerden farklıysa, alıcının uygulama iletisi verilerini gereken karakter kümesine ve kodlamaya dönüştürmesi gerekir. GMCONV seçeneği, ileti verilerinin dönüştürülmesini gerçekleştirmek için kullanıcı tarafından yazılan bir çıkışla kullanılabilir (bu seçeneğin ayrıntıları için bkz. [“IBM i üzerinde MQGMO \(Get-message seçenekleri\)” sayfa 1060](#)).

Not: MQGET çağrısındaki diğer tüm değiştirgeler, yerel kuyruk yöneticisinin karakter kümesi ve kodlamasında (**CodedCharSetId** kuyruk yöneticisi özniteliği ve ENNAT ile verilir) bulunur.

Arama başarısız olursa, arabelleğin içeriği değişmiş olabilir.

DATLEN (10 basamaklı işaretli tamsayı)-çıkış

İletinin uzunluğu.

Bu, iletideki uygulama verilerinin bayt cinsinden uzunluğudur. Bu ileti uzunluğu BUFLEN değerinden büyükse, **BUFFER** değiştirgesinde yalnızca BUFLEN byte döndürülür (iletisi kesilir). Değer sıfırsa, iletinin uygulama verisi içermediği anlamına gelir.

BUFLEN ileti uzunluğundan azsa, DATLEN kuyruk yöneticisi tarafından, **GMO** parametresinde GMATM belirtilip belirtilmediğine bakılmaksızın girilir (daha fazla bilgi için [“IBM i üzerinde MQGMO \(Get-message seçenekleri\)” sayfa 1060](#) içinde açıklanan GMOPT alanına bakın). Bu, uygulamanın ileti

verilerini barındırmak için gereken arabellek büyüklüğünü saptamasını sağlar ve daha sonra, çağrıyı uygun büyüklükte bir arabellekle yeniden yürütür.

Ancak, GMCONV seçeneği belirtilirse ve dönüştürülen ileti verileri BUFFERiçine sığmayacak kadar uzunsa, DATLEN için döndürülen değer:

- Kuyruk yöneticisi tanımlı biçimler için dönüştürülmemiş verilerin uzunluğu.

Bu durumda, verilerin türü dönüştürme sırasında genişletilmesine neden olursa, uygulamanın DATLENiçin kuyruk yöneticisi tarafından döndürülen değerden daha büyük bir arabellek ayırması gerekir.

- Uygulama tanımlı biçimler için, veri dönüştürme çıkışı tarafından döndürülen değer.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici CMPCOD.

Aşağıdaki neden kodları, kuyruk yöneticisinin **REASON** parametresi için döndürebileceği kodlardır. Uygulama GMCONV seçeneğini belirtiyorsa ve ileti verilerinin bazılarını ya da tümünü dönüştürmek için kullanıcı tarafından yazılan bir çıkış çağrılırsa, **REASON** parametresi için hangi değer döndürüleceğine karar veren çıkış olur. Sonuç olarak, bu kısımda daha sonra belgelenen değerlerden başka değerler de olabilir.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2120

(2120, X'848 ') Dönüştürülen veriler arabellek için çok büyük.

RC2190

(2190, X'88E') Dönüştürülen dizgi alan için çok büyük.

RC2150

(2150, X'866 ') DBCS dizgisi geçersiz.

RC2110

(2110, X'83E') İleti biçimi geçersiz.

RC2243

(2243, X'8C3') İleti bölümlerinin CCSID' leri farklı.

RC2244

(2244, X'8C4') İleti bölümlerinin farklı kodlamaları var.

RC2209

(2209, X'8A1') Kilitli ileti yok.

RC2119

(2119, X'847 ') İleti verileri dönüştürülmedi.

RC2272

(2272, X'8E0') İleti verileri kısmen dönüştürüldü.

RC2145

(2145, X'861 ') Kaynak arabellek deęiřtirgesi geersiz.

RC2111

(2111, X'83F') Kaynak kodlanmış karakter takımı tanıtıcısı geersiz.

RC2113

(2113, X'841 ') İletideki paketli ondalık kodlama tanınmıyor.

RC2114

(2114, X'842 ') İletide kayan nokta kodlaması tanınmıyor.

RC2112

(2112, X'840 ') Kaynak tamsayı kodlaması tanınmıyor.

RC2143

(2143, X'85F') Kaynak uzunluęu parametresi geersiz.

RC2146

(2146, X'862 ') Hedef arabellek deęiřtirgesi geersiz.

RC2115

(2115, X'843 ') Hedef kodlanmış karakter takımı tanıtıcısı geersiz.

RC2117

(2117, X'845 ') Alıcı tarafından belirtilen Paketlenmiş ondalık kodlama tanınmıyor.

RC2118

(2118, X'846 ') Alıcı tarafından belirtilen kayan nokta kodlaması tanınmıyor.

RC2116

(2116, X'844 ') Hedef tamsayı kodlaması tanınmıyor.

RC2079

(2079, X'81F') İzlenen ileti döndürüldü (işleme tamamlandı).

RC2080

(2080, X'820 ') İzlenen ileti döndürüldü (işlem tamamlanmadı).

CMPCOD CCFAIL ise:

RC2004

(2004, X'7D4') Arabellek deęiřtirgesi geersiz.

RC2005

(2005, X'7D5') Arabellek uzunluęu parametresi geersiz.

RC2219

(2219, X'8AB') MQI çağırısı önceki arama tamamlanmadan yeniden girildi.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2010

(2010, X'7DA') Veri uzunluęu parametresi geersiz.

RC2016

(2016, X'7E0') Kuyruk için engellendi.

RC2186

(2186, X'88A') Get-message seçenekleri yapısı geersiz.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geersiz.

RC2019

(2019, X'7E3') Nesne tanıtıcısı geçerli deęil.

RC2241

(2241, X'8C1') İleti grubu tamamlanmadı.

RC2242

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

- RC2259**
(2259, X'8D3') Tutarsız göz atma belirtimi.
- RC2245**
(2245, X'8C5') İş birimi belirtimi tutarsız.
- RC2246**
(2246, X'8C6') İmleç altındaki ileti alma için geçerli değil.
- RC2247**
(2247, X'8C7') Eşleştirme seçenekleri geçerli değil.
- RC2026**
(2026, X'7EA') İleti tanımlayıcısı geçersiz.
- RC2250**
(2250, X'8CA') İleti sıra numarası geçersiz.
- RC2033**
(2033, X'7F1') İleti yok.
- RC2034**
(2034, X'7F2') Göz at imleci iletide konumlandırılmadı.
- RC2036**
(2036, X'7F4') Kuyruk göz atmak için açık değil.
- RC2037**
(2037, X'7F5') Kuyruk giriş için açık değil.
- RC2041**
(2041, X'7F9') Nesne tanımı açıldıktan sonra değiştirildi.
- RC2101**
(2101, X'835 ') Nesne zarar gördü.
- RC2046**
(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.
- RC2052**
(2052, X'804 ') Kuyruk silindi.
- RC2058**
(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.
- RC2059**
(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.
- RC2161**
(2161, X'871 ') Kuyruk yöneticisi susturuyor.
- RC2162**
(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.
- RC2102**
(2102, X'836 ') Sistem kaynakları yetersiz.
- RC2071**
(2071, X'817 ') Kullanılabilir bellek yetersiz.
- RC2024**
(2024, X'7E8') Yürürlükteki iş birimi içinde başka ileti işlenmez.
- RC2072**
(2072, X'818 ') Syncpoint desteği kullanılamıyor.
- RC2195**
(2195, X'893 ') Beklenmeyen hata oluştu.
- RC2255**
(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.
- RC2090**
(2090, X'82A') MQGMO' daki bekleme aralığı geçersiz.

RC2256

(2256, X'8D0') Yanlıř MGMO srm saęlandı.

RC2257

(2257, X'8D1') Yanlıř MMD srm saęlandı.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQGET(HCONN : HOBJ : MSGDSC : GMO :
C          BUFLN : BUFFER : DATLEN :
C          CMPCOD : REASON)
```

aęrıya iliřkin prototip tanımlaması:

```
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

IBM i zerinde MIN (Nesne znitelikleri hakkında sorgu)

MIN aęırısı, bir nesnenin zniteliklerini ieren bir dizi tamsayı ve karakter dizisi dndrr.

Ařaęıdaki nesne tipleri geerlidir:

- Kuyruk
- Ad listesi
- Sre tanımlaması
- Kuyruk yneticisi
- [“Sdzimizi” sayfa 1283](#)
- [“Kullanım notları” sayfa 1283](#)
- [“Parametreler” sayfa 1285](#)
- [“RPG Bildirimi” sayfa 1291](#)

Sdzimizi

MIN (*HCONN*, *HOBJ*, *SELCNT*, *SELS*, *IACNT*, *INTATR*, *CALEN*, *CHRATR*, *CMPCOD*, *REASON*)

Kullanım notları

1. Dndrlen deęerler, seilen zniteliklerin anlık grntsdr. Uygulamanın dndrlen deęerlere gre hareket edebilmesi iin zniteliklerin deęiřtirilmedięine iliřkin bir garanti yoktur.
2. Bir model kuyruęunu atıęınızda, dinamik bir yerel kuyruk yaratılır. Bu, znitelikleri hakkında bilgi almak iin model kuyruęunu asanız da doęrudur.

Dinamik kuyruğun öznitelikleri (belirli kural dışı durumlar dışında), dinamik kuyruk yaratıldığında model kuyruğunun öznitelikleriyle aynıdır. Daha sonra bu kuyrukta MQINQ çağrısı kullanırsanız, kuyruk yöneticisi model kuyruğunun özniteliklerini değil, dinamik kuyruğun özniteliklerini döndürür. Dinamik kuyruk tarafından devralınan model kuyruğu özniteliklerine ilişkin ayrıntılar için bkz. [Tablo 1](#).

3. Sorulması gereken nesne bir diğer ad kuyruğuysa, MQINQ çağrısının döndürdüğü öznitelik değerleri, diğer adın çözüldüğü temel kuyruğun değil, diğer ad kuyruğunun değerleridir.

4. Sorulacak nesne bir küme kuyruğuysa, sorulabilecek öznitelikler kuyruğun nasıl açılmasına bağlıdır:

- Küme kuyruğu sorgunun yanı sıra bir ya da daha fazla giriş, göz atma ya da küme için açılırsa, açmanın başarılı olması için küme kuyruğunun yerel bir eşgörünümü olmalıdır. Bu durumda, sorulabilecek öznitelikler yerel kuyruklar için geçerlidir.
- Küme kuyruğu tek başına sorgu ya da sorgu ve çıkış için açılırsa, yalnızca aşağıdaki öznitelikler sorgulanabilir; **QType** özniteliği bu durumda QTCLUS değerine sahiptir:

- SSS
- SSS
- IADBND
- İADPER
- IADPRI
- IAIPUT
- IAQTYP

Küme kuyruğu değişmez bağ tanımı olmadan açılırsa (MQOPEN çağrısında belirtilen OOBNDN ya da **DefBind** özniteliği BNDNOT değerine sahip olduğunda belirtilen OOBNDQ), kuyruk için ardışık MQINQ çağrıları küme kuyruğunun farklı eşgörünümlerini sorgulayabilir, ancak genellikle tüm eşgörünümler aynı öznitelik değerlerine sahiptir.

Küme kuyruklarıyla ilgili ek bilgi için [Kuyruk yöneticisi kümesinin yapılandırılması](#) başlıklı konuya bakın.

5. Çok sayıda öznitelik sorulacaksa ve bunların bazıları MQSET çağrısı kullanılarak ayarlanırsa, seçilecek özniteliklerin başında konumlandırılması uygun olabilir; böylece, MQSET için aynı diziler (azaltılmış sayılarla) kullanılabilir.

6. Birden çok uyarı durumu ortaya çıkarsa (bkz. **CMPCOD** parametresi), döndürülen neden kodu, geçerli olan aşağıdaki listede *ilk* olur:

- a. RC2068
- b. RC2022
- c. RC2008

7. Nesne öznitelikleri ile ilgili ek bilgi için aşağıdaki konuya bakın:

- [“Kuyruklara ilişkin öznitelikler” sayfa 1345](#)
- [“Ad listesi öznitelikleri” sayfa 1373](#)
- [“IBM i üzerinde süreç tanımlamalarına ilişkin öznitelikler” sayfa 1374](#)
- [“IBM i üzerindeki kuyruk yöneticisine ilişkin öznitelikler” sayfa 1376](#)

8. Yeni bir yerel kuyruk SYSTEM.ADMIN.COMMAND.EVENT , komutlar her yayınlandığında oluşturulan iletileri kuyruğa almak için kullanılır. CMDEV kuyruk yöneticisi özniteliğinin nasıl ayarlandığına bağlı olarak, çoğu komut için iletiler bu kuyruğa yerleştirilir:

- ENABLED-komut olayı iletileri oluşturulur ve tüm başarılı komutlar için kuyruğa yerleştirilir.
- NODISPLAY-komut olayı iletileri oluşturulur ve DISPLAY (MQSC) komutu ve Inquire (PCF) komutu dışındaki tüm başarılı komutlar için kuyruğa yerleştirilir.
- DISABLED-Komut olayı iletileri oluşturulmaz (bu, kuyruk yöneticisinin ilk varsayılan değeridir).

Parametreler

MQINQ çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı iřaretli tamsayı)-giriř

Baęlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik baęlantıyı gösterir. HCONN deęeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

HOBJ (10 basamaklı iřaretli tamsayı)-giriř

Nesne tanıtıcısı.

Bu tanıtıcı, gerekli öznitelikleri olan nesneyi (herhangi bir tipte) gösterir. Tanıtıcı deęeri, OOINQ seçeneđini belirten önceki bir MQOPEN çağrısı tarafından döndürülmüř olmalıdır.

SELCNT (10 basamaklı iřaretli tamsayı)-giriř

Seçicilerin sayısı.

Bu, SELS dizisinde saęlanan seçicilerin sayısıdır. Döndürülecek özniteliklerin sayısıdır. Sıfır geçerli bir deęerdir. İzin verilen üst sınır 256 'dır.

SELS (10 basamaklı iřaretli tamsayı x SELCNT)-giriř

Öznitelik seçicilerin dizisi.

Bu bir SELCNT öznitelik seçicileri dizisidir; her seçici, gerekli bir deęere sahip bir özniteliđi (tamsayı ya da karakter) tanımlar.

Her seçici, HOBJ ' un temsil ettiđi nesne tipi için geçerli olmalıdır; tersi durumda, çağrı tamamlanma kodu CCFAIL ve neden kodu RC2067 ile başarısız olur.

Kuyrukların özel durumunda:

- Seçici herhangi bir tipindeki kuyruklar için geçerli deęilse, çağrı tamamlanma kodu CCFAIL ve neden kodu RC2067 ile başarısız olur.
- Seçici yalnızca nesne tipi ya da tipi dıřındaki kuyruklar için geçerliyse, çağrı tamamlanma kodu CCWARN ve neden kodu RC2068 ile başarılı olur.
- Sorulan kuyruk bir küme kuyruđuysa, geçerli seçiciler kuyruđun nasıl çözümlendiđine baęlıdır; daha fazla ayrıntı için kullanım notu 4 'e bakın.

Seçiciler herhangi bir sırada belirtilebilir. Tamsayı öznitelik seçicilere (IA* seçicileri) karşılık gelen öznitelik deęerleri, INTATR içinde bu seçicilerin SELS içinde gerçekteřtiđi sırayla döndürülür. Karakter öznitelik seçicilere (CA* seçicileri) karşılık gelen öznitelik deęerleri, CHRATR içinde bu seçicilerin oluřtuđu sırada döndürülür. IA* seçiciler CA* seçicileriyle serpiřtirilebilirler; yalnızca her tip içindeki görelü sıra önemlidir.

Not:

1. Tamsayı ve karakter öznitelik seçicileri iki farklı aralık içinde ayrılır; IA* seçicileri IAFRST-IALAST aralığında ve CA* seçicileri CAFRST-CALAST aralığında bulunur.

Her aralık için, IALSTU ve CALSTU deęiřmezleri, kuyruk yöneticisinin kabul ettiđi en yüksek deęeri tanımlar.

2. Önce tüm IA* seçicileri ortaya çıkarsa, SELS ve INTATR dizilerinde karşılık gelen öđeleri adreslemek için aynı öđe numaraları kullanılabilir.

Sorulabilecek öznitelikler ařađıdaki çizelgelerde listelenir. CAY* seçicileri için, CHRATR içinde elde edilen dizinin uzunluđunu bayt cinsinden tanımlayan deęiřmez parantez içinde verilir.

Çizelge 746. Kuyruklar için MQINQ öznitelik seçicileri		
Seçici	Açıklama	Not
CAALTD	En son deęiřiklik tarihi (LNDATE).	1

Çizelge 746. Kuyruklar için MQINQ öznelik seçicileri (devamı var)		
Seçici	Açıklama	Not
CAALTT.	En son değişiklik zamanı (LNTIME).	1
CABRQN	Aşırı geri dönüş adı (LNQN).	5
KABASQ	Diğer adın çözüldüğü kuyruğun adı (LNQN).	
CACFSN	Bağlaşım olanağı yapısı adı (LNCFSN).	3
CACLN	Küme adı (LNCLUN).	1
CACLNL	Küme ad listesi (LNNLN).	1
CACRTD	Kuyruk oluşturma tarihi (LNCRTD).	
CACRTT.	Kuyruk oluşturma süresi (LNCRTT).	
CAINIQ.	Başlatma kuyruğu adı (LNQN).	
CAPRON.	Süreç tanımlamasının adı (LNPRON).	
SSS	Kuyruk tanımlaması (LNQD).	
SSS	Kuyruk adı (LNQN).	
CARQMN	Uzak kuyruk yöneticisinin adı (LNQMN).	
CARQN	Uzak kuyruk yöneticisinde (LNQN) bilinen uzak kuyruğun adı.	
KATRGD	Tetikleme verileri (LNTRGD).	5
CAXQN	İletim kuyruğu adı (LNQN).	
IABTHR	Geriletme eşiği.	5
IACDEP	Kuyruktaki iletilerin sayısı.	
IADBND	Varsayılan bağ tanımı.	1
IADINP	Varsayılan giriş için aç seçeneği.	5
İADPER	Varsayılan ileti kalıcılığı.	
IADPRI	Varsayılan ileti önceliği.	5
IADEFT	Kuyruk tanımlaması tipi.	
İADIST	Dağıtım listesi desteği.	2
IAHGB	Geri sayımı sertleştirip sertleşmeyeceğini.	5
IAIGET	Alma işlemlerine izin verilip verilmediğini belirler.	
IAIPUT	Koyma işlemlerine izin verilip verilmediğini belirler.	
IAMLEN	İleti uzunluğu üst sınırı.	
IAMDEP	Kuyrukta izin verilen ileti sayısı üst sınırı.	
IAMDS ' LER	İleti önceliğinin ilgili olup olmadığını belirler.	5
IAİKÖ	Kuyruk giriş için açık olan MQOPEN çağrılarının sayısı.	
IAOOC.	Kuyruğun çıkış için açık olduğu MQOPEN çağrılarının sayısı.	
IAQDHE	Kuyruk derinliği yüksek olayları için denetim özneliği.	4, 5
IAQDHL	Kuyruk derinliği için üst sınır.	4, 5

Çizelge 746. Kuyruklar için MQINQ öznelik seçicileri (devamı var)		
Seçici	Açıklama	Not
IAQDLE	Kuyruk derinliği düşük olayları için denetim özneliği.	4, 5
IAQDLL	Kuyruk derinliği için alt sınır.	4, 5
IAQDME	Kuyruk derinliği olay sayısı üst sınırı için denetim özneliği.	4, 5
IAQSI	Kuyruk hizmeti aralığı sınırı.	4, 5
IAQSIE	Kuyruk hizmeti aralığı olayları için denetim özneliği.	4, 5
IAQTYP	Kuyruk tipi.	
IAQSGD	Kuyruk paylaşım grubu düzenleme.	3
İARINT	Kuyruk alıkoyma aralığı.	5
IASCOP.	Kuyruk tanımlaması kapsamı.	4, 5
IASHAR	Kuyruğun giriş için paylaşılıp paylaşılmayacağını belirler.	
IATRGC	Tetikleme kontrolü.	
IATRGD	Tetikleme derinliği.	5
IATRGP	Tetikleyiciler için eşik ileti önceliği.	5
IATRGT	Tetikleyici tipi.	
IAUSAG	Kullanım.	
CLWLUSEQ	Uzak kuyrukları kullan.	

Not:

1. Aşağıdaki altyapılarda desteklenir:


-  AIX
-  IBM i
-  Windows
-  z/OS


ve bu sistemlere bağlı IBM MQ MQI clients için.

2. Aşağıdaki altyapılarda desteklenir:

-  AIX
-  IBM i
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

3.  z/OS üzerinde desteklenir.

4.  z/OS üzerinde desteklenmez.

5. VSE/ESA üzerinde desteklenmez.

<i>Çizelge 747. Ad listesi için MQINQ öznitelik seçicileri</i>		
Seçici	Açıklama	Not
CAALTD	En son değişiklik tarihi (LNDATE)	1
CAALTT.	En son değişiklik zamanı (LNTIME)	1
KALSTD	Ad listesi tanımlaması (LNNLD)	1
KALSTN	Ad listesi nesnesinin adı (LNNLN)	1
YAMLAR	Ad listesindeki adlar (LNQN x <i>Listedeki ad sayısı</i>)	1
ANAMC	Ad listesindeki ad sayısı	1
IAQSGD	Kuyruk paylaşım grubu yok etme	3

<i>Çizelge 748. Süreç tanımlamaları için MQINQ öznitelik seçicileri</i>		
Seçici	Açıklama	Not
CAALTD	En son değişiklik tarihi (LNDATE)	1
CAALTT.	En son değişiklik zamanı (LNTIME)	1
CAAPPI	Uygulama tanıtıcısı (LNPROA)	5
CAENVD (SERMAYE)	Ortam verileri (LNPROE)	5
CAPROD (SERMAYE)	Süreç tanımlamasının açıklaması (LNPROD)	5
CAPRON.	Süreç tanımlamasının adı (LNPRON)	5
CAUSRD.	Kullanıcı verileri (LNPROU)	5
IAAPPT	Uygulama tipi	5
IAQSGD	Kuyruk paylaşım grubu yok etme	3

<i>Çizelge 749. Kuyruk yöneticisine ilişkin MQINQ öznitelik seçicileri</i>		
Seçici	Açıklama	Not
CAALTD	En son değişiklik tarihi (LNDATE)	1
CAALTT.	En son değişiklik zamanı (LNTIME)	1
KACADX	Otomatik kanal tanımlaması çıkış adı (LNEXN)	1
CACLWD	Küme iş yükü çıkışına (LNEXDA) geçirilen veriler	1
CACLWX	Küme iş yükü çıkışının adı (LNEXN)	1
CACMDQ	Sistem komutu giriş kuyruğu adı (LNQN)	5
KADLQ	Gönderilmeyen iletiler kuyruğunun adı (LNQN)	5
CADXQN	Varsayılan iletim kuyruğu adı (LNQN)	5
CAQMD	Kuyruk yöneticisi tanımlaması (LNQMD)	5
SAQMID	Kuyruk yöneticisi tanıtıcısı (LNQMID)	1
SAQMN	Yerel kuyruk yöneticisinin adı (LNQMN)	5
SAQSGN	Kuyruk paylaşım grubu adı (LNQSGN)	3

Çizelge 749. Kuyruk yöneticisine ilişkin MQINQ öznitelik seçicileri (devamı var)		
Seçici	Açıklama	Not
KARPN	Kuyruk yöneticisinin havuz hizmetleri sağladığı kümenin adı (LNQMN)	1
KARPNL	Kuyruk yöneticisinin havuz hizmetleri sağladığı kümelerin adlarını içeren ad listesi nesnesinin adı (LNNLN)	1
CMDEV	Komutlar yayınlandığında oluşturulan iletilerin bir kuyruğa konup konmayacağını belirleyen denetim özniteliği	8
IAAUTE	Yetki olayları için denetim özniteliği	4, 5
IACAD	Otomatik kanal tanımlaması için denetim özniteliği	2
İACADE	Otomatik kanal tanımlaması olayları için denetim özniteliği	2
IACLXQ	Varsayılan küme iletim kuyruğu tipi	4
IACLWL	Küme iş yükü uzunluğu	1
IACCSI	Kodlanmış karakter takımı tanıtıcısı	5
IACMDL	Kuyruk yöneticisi tarafından desteklenen komut düzeyi	5
İACFGE	Yapılandırma olayları için denetim özniteliği	3
İADIST	Dağıtım listesi desteği	2
IAINHE.	Olayları engelleme için denetim özniteliği	4, 5
IACLE	Yerel olaylar için denetim özniteliği	4, 5
IAMHND.	Tanıtıcı sayısı üst sınırı	5
IAMLEN	İleti uzunluğu üst sınırı	5
IAMPRI	Öncelik üst sınırı	5
IAMUNC	Bir iş birimindeki kesinleştirilmemiş ileti sayısı üst sınırı	5
IAPFME	Performans olayları için denetim özniteliği	4, 5
IAPLAT	Kuyruk yöneticisinin bulunduğu platform	5
ARMTE	Uzak olaylar için denetim özniteliği	4, 5
IASSE.	Durdurma başlatma olayları için denetim özniteliği	4, 5
IASYNC	Eşitleme noktası kullanılabilirliği	5
IATRLFT	Kullanılmayan yönetimle ilgili olmayan konuların geçerlik süresi	
IATRGI	Tetikleme aralığı	5

IACNT (10 basamaklı işaretli tamsayı)-giriş

Tamsayı özniteliklerinin sayısı.

Bu, INTATR dizisindeki öğelerin sayısıdır. Sıfır geçerli bir değerdir.

Bu, **SELS** parametresindeki en az IA* seçicilerinin sayıya, istenen tüm tamsayı öznitelikleri döndürülür.

INTATR (10 basamaklı işaretli tamsayı x IACNT)-çıkış

Tamsayı öznitelikleri dizisi.

Bu, IACNT tamsayı öznitelik değerlerinin bir dizisidir.

Tamsayı öznitelik değerleri, **SELS** parametresindeki IA* seçicileri ile aynı sırada döndürülür. Dizi, IA* seçicilerinin sayısından daha fazla öge içeriyorsa, fazla ögeler değişmez.

HOBJS bir kuyruğu gösteriyorsa, ancak o kuyruk tipine bir öznitelik seçici uygulanamıyorsa, INTATTR dizisindeki ilgili öge için IAVNA özel değeri döndürülür.

CALEN (10 basamaklı işaretli tamsayı)-giriş

Karakter öznitelikleri arabelleğinin uzunluğu.

Bu, **CHRATR** parametresinin bayt cinsinden uzunluğudur.

Bu, en azından istenen karakter özniteliklerinin uzunluklarının toplamı olmalıdır (bkz. SELS). Sıfır geçerli bir değerdir.

CHRATR (1 baytlık karakter dizisi x CALEN)-çıkış

Karakter öznitelikleri.

Bu, karakter özniteliklerinin birlikte döndürüldüğü ve birleştirildiği arabellektir. Arabelleğin uzunluğu **CALEN** değeriyle verilir.

Karakter öznitelikleri, **SELS** parametresindeki CA* seçicileri ile aynı sırada döndürülür. Her öznitelik dizisinin uzunluğu her öznitelik için sabitlenir (bkz. SELS) ve içindeki değer, gerekirse boşluklarla sağa doğru doldurulur. Arabellek, istenen tüm karakter özniteliklerini (doldurma da içinde olmak üzere) içermesi gerekenden büyükse, döndürülen son öznitelik değerinin ötesindeki baytlar değişmez.

HOBJS bir kuyruğu gösteriyorsa, ancak o kuyruk tipi için bir öznitelik seçici geçerli değilse, CHRATR içinde bu öznitelik değeri olarak tamamen yıldız işaretlerinden (*) oluşan bir karakter dizisi döndürülür.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici CMPCOD.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2008

(2008, X'7D8') Karakter öznitelikleri için yeterli yer yok.

RC2022

(2022, X'7E6') Tamsayı öznitelikleri için yeterli yer yok.

RC2068

(2068, X'814 ') Seçici kuyruk tipi için geçerli değil.

CMPCOD CCFail ise:

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2006

(2006, X'7D6') Karakter özniteliklerinin uzunluğu geçersiz.

RC2007

(2007, X'7D7') Karakter öznitelikleri dizgisi geçersiz.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2019

(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

RC2021

(2021, X'7E5') Tamsayı özniteliklerinin sayısı geçersiz.

RC2023

(2023, X'7E7') Tamsayı öznitelikleri dizisi geçerli değil.

RC2038

(2038, X'7F6') Kuyruk sorma için açık değil.

RC2041

(2041, X'7F9') Nesne tanımı açıldıktan sonra değiştirildi.

RC2101

(2101, X'835 ') Nesne zarar gördü.

RC2052

(2052, X'804 ') Kuyruk silindi.

RC2058

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

RC2059

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2065

(2065, X'811 ') Seçici sayısı geçerli değil.

RC2067

(2067, X'813 ') Öznitelik seçici geçersiz.

RC2066

(2066, X'812 ') Seçicilerin sayısı çok büyük.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```
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)
```

Çağrıya ilişkin prototip tanımlaması:

```

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

IBM i üzerinde MQINQMP (Inquire message özelliği)

MQINQMP çağrısı, bir iletinin özelliğinin değerini döndürür.

- “Sözdizimi” sayfa 1292
- “Parametreler” sayfa 1292
- “RPG Bildirimi” sayfa 1296

Sözdizimi

MQINQMP (*Hconn*, *Hmsg*, *InqPropOpts*, *Name*, *PropDesc*, *Type*, *ValueLength*, *Value*, *DataLength*, *CompCode*, *Reason*)

Parametreler

MQINQMP çağrısındaki değişirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri, **Hmsg** değiştirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eşleşmelidir.

İleti tanıtıcısı HCUNAS kullanılarak yaratıldıysa, ileti tanıtıcısının bir özelliğini soran iş parçacığında geçerli bir bağlantı kurulmalıdır; tersi durumda, çağrı RC2009ile başarısız olur.

HMSG (20 basamaklı işaretli tamsayı)-giriş

Bu, sorulacak ileti tanıtıcısıdır. Değer, önceki bir **MQCRTMH** çağrısı tarafından döndürüldü.

INQOPT (MQIMPO)-giriş

Ayrıntılar için [MQIMPO](#) veri tipine bakın.

PRNAME (MQCHARV)-giriş

Bu, sorgunun yapılacağı özelliğin adını açıklar.

Bu ada sahip bir özellik bulunamazsa, arama RC2471nedeniyle başarısız olur.

Özellik adının sonunda yüzde işareti (%) karakterini kullanabilirsiniz. Joker karakter, nokta (.) karakteri de içinde olmak üzere sıfır ya da daha fazla karakterle eşleşir. Bu, bir uygulamanın birçok özelliğinin değerini sorgulamasına olanak sağlar. İlk eşleşen özelliği almak için IPINQF seçeneğiyle MQINQMP ' yi ve sonraki eşleşen özelliği almak için IPINQN seçeneğini yeniden çağırın. Başka eşleşen özellik yoksa, arama RC2471ile başarısız olur. InqPropOpts yapısının *ReturnedName* alanı, özelliğinin döndürülen adı için bir adres ya da görelî konumla kullanıma hazırlandıysa, bu, eşleştirilen özelliğinin adıyla MQINQMP

' den dönüldükten sonra tamamlanır. InqPropiçindeki *ReturnedName* alanının *VSBufSize* alanı döndürülen özellik adının uzunluğundan azsa, tamamlama kodu RC2465nedeniyle CCFAIL olarak ayarlanır.

Bilinen eşanlamlıları olan özellikler aşağıdaki gibi döndürülür:

1. "mqps" öneğine sahip özellikler. IBM MQ özellik adıyla döndürülür. Örneğin, "MQTopicString", "mqps.Top" yerine döndürülen addır.
2. "jms" öneğine sahip özellikler. Ya da "McD." JMS üstbilgi alanı adı olarak döndürülür. Örneğin, "JMSExpiration", "jms.Exp" yerine döndürülen addır.
3. "usr." öneğine sahip özellikler bu önek olmadan döndürülür. Örneğin, "usr.Color" yerine "Color" döndürülür.

Eşanlamlıları olan özellikler yalnızca bir kez döndürülür.

RPG programlama dilinde, "usr" ile başlayan tüm özellikleri ve özellikleri sorgulamak için aşağıdaki makro değişkenleri tanımlanır:

GİRİŞ

İletinin tüm özelliklerini sorgulamanızı sağlar.

INQUSR

"usr." ile başlayan iletinin tüm özelliklerini sorgulamanızı sağlar. Döndürülen ad "usr" olmadan döndürülür. önek.

IPINQN belirtilirse, ancak Ad önceki çağrıdan sonra değiştiyse ya da bu ilk çağrıysa, IPINQF örtük olarak belirtilir.

Özellik adlarının kullanımıyla ilgili ek bilgi için [Özellik adları](#) ve [Özellik adı kısıtlamaları](#) konusuna bakın.

PRPDSC (MQPD)-çıkış

Bu yapı, özellik desteklenmiyorsa ne olacağı, özelliğin hangi ileti bağlamına ait olduğu ve özelliğin hangi iletilere kopyalanması gerektiği de içinde olmak üzere, bir özelliğin özniteliklerini tanımlamak için kullanılır. Bu yapının ayrıntıları için bkz. [MQPD](#).

TYPE (10 basamaklı işaretli tamsayı)-giriş/çıkış

MQINQMP çağrısından dönüş için bu parametre *Değerveri* tipine ayarlanır. Veri tipi aşağıdakilerden biri olabilir:

TYPBOL

Bir boole.

TYPBST

bir bayt dizilimi.

TYPI8

8 bitlik işaretli bir tamsayı.

TYPI16

16 bit işaretli tamsayı.

TYPI32

32 bit işaretli tamsayı.

TYPI64

64 bitlik imzalı bir tamsayı.

TYPF32

32 bit kayan noktalı sayı.

TYPF64

64 bit kayan noktalı sayı.

TYPSTR

Bir karakter dizgisi.

TYPNUL

Özellik var, ancak boş değerli.

Özellik değerinin veri tipi tanınmazsa, TYPSTR döndürülür ve değer dizgi gösterimi *Değer* alanına yerleştirilir. *IPOPT* değiştirgesinin *IPTYP* alanında, veri tipinin dizgi gösterimini bulabilirsiniz. RC2467nedeniyle bir uyarı tamamlanma kodu döndürüldü.

Ayrıca, IPCTYP seçeneği belirtilirse, özellik değerinin dönüştürülmesi istenecektir. Özelliğin döndürülmesini istediğiniz veri tipini belirtmek için giriş olarak *Tip* ögesini kullanın. Veri tipi dönüştürmeye ilişkin ayrıntılar için "IBM i üzerinde MQIMPO (İleti özelliği seçeneklerini sorgularken sor)" sayfa 1086 IPCTYP seçeneğinin açıklamasına bakın.

Tip dönüştürme isteğinde bulunmazsanız, girişte aşağıdaki değeri kullanabilirsiniz:

TYPAST

Özelliğin değeri, veri tipi dönüştürülmeden döndürülür.

VALLEN (10 basamaklı işaretli tamsayı)-giriş

Değer alanının bayt cinsinden uzunluğu.

Değer döndürülmesini gerektirmediğiniz özellikler için sıfır belirtin. Bunlar, bir uygulama tarafından boş değere ya da boş bir dizgiye sahip olacak şekilde tasarlanmış özellikler olabilir. IPQLEN seçeneği belirlendiyse sıfır değerini de belirleyin; bu durumda değer döndürülmez.

VALUE (1 baytlık bit stringxVALLEN)-çıkış

Bu, sorulmuş özellik değerini içerecek alandır. Arabellek, döndürülmekte olan değer için uygun bir sınıra hizalanmalıdır. Bunun yapılmaması, değere daha sonra erişildiğinde bir hatayla sonuçlanabilir.

VALLEN özellik değerinin uzunluğundan azsa, özellik değerinin mümkün olduğunca çoğu *VALUE* değerine taşınır ve çağrı tamamlanma kodu CCFAIL ve neden RC2469ile başarısız olur.

VALUE içindeki verilerin karakter takımı, INQOPT parametresindeki IPRETCSE alanı tarafından verilir. *VALUE* içindeki verilerin kodlaması, INQOPT parametresindeki IİDC alanı tarafından verilir.

VALLEN parametresi sıfırsa, *VALUE* değerine başvurulur.

DATLEN (10 basamaklı işaretli tamsayı)-çıkış

Bu, *Değer* alanında döndürülen gerçek özellik değerinin bayt cinsinden uzunluğudur.

DataLength özellik değeri uzunluğundan azsa, MQINQMP çağrısından dönüşte *DataLength* değeri girilir. Bu, uygulamanın özellik değerini karşılamak için gereken arabellek büyüklüğünü belirlemesini sağlar ve daha sonra, çağrıyı uygun büyüklükte bir arabellekle yeniden yürütün.

Aşağıdaki değerler de döndürülebilir.

Tip parametresi TYPSTR ya da TYPBST olarak ayarlanırsa:

VLEMP

Özellik var, ancak karakter ya da byte içermiyor.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu; aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CompCode*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode CCWARN ise:

RC2492

(2492, X'09BC') Döndürülen özellik adı dönüştürülmedi.

RC2466

(2466, X'09A2') Özellik değeri dönüştürülmedi.

RC2467

(2467, X'09A3') Özellik veri tipi desteklenmiyor.

RC2421

(2421, X'0975 ') Özellikleri içeren bir MQRFH2 klasörü ayrıştırılmadı.

CMPCOD CCFAIL ise:

RC2204

(2204, X'089C') Bağdaştırıcı kullanılmıyor.

RC2130

(2130, X'0852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

RC2157

(2157, X'086D') Birincil ve ana ASID' ler farklıdır.

RC2004

(2004, X'07D4') Değer parametresi geçersiz.

RC2005

(2005, X'07D5') Değer uzunluğu parametresi geçersiz.

RC2219

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

RC2009

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2010

(2010, X'07DA') Veri uzunluğu parametresi geçersiz.

RC2464

(2464, X'09A0') İleti sorgusuna ilişkin özellik seçenekleri yapısı geçersiz.

RC2460

(2460, X'099C') İleti tanıtıcısı geçersiz.

RC2499

(2499, X'09C3') İleti tanıtıcısı kullanımda.

RC2064

(2046, X'07F8') Seçenekler geçerli değil ya da tutarlı değil.

RC2482

(2482, X'09B2') Özellik tanımlayıcı yapısı geçersiz.

RC2470

(2470, X'09A6') Gerçek veri tipinden istenen veri tipine dönüştürme desteklenmiyor.

RC2442

(2442, X'098A') Özellik adı geçersiz.

RC2465

(2465, X'09A1') Özellik adı, döndürülen ad arabelleği için çok büyük.

RC2471

(2471, X'09A7) Özellik yok.

RC2469

(2469, X'09A5') Özellik değeri, Değer alanı için çok büyük.

RC2472

(2472, X'09A8') Değer verilerinde sayı biçimi hatası saptandı.

RC2473

(2473, X'09A9') İstenen özellik tipi geçersiz.

RC2111

(2111, X'083F') Özellik adı kodlanmış karakter takımı tanıtıcısı geçersiz.

RC2071

(2071, X'0871 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'0893 ') Beklenmeyen hata oluştu.

Bu kodlara ilişkin ayrıntılı bilgi için aşağıdaki adrese bakın:

- [IBM MQ for z/OS iletileri, tamamlama ve neden kodları - IBM MQ for z/OS](#)
- Diğer tüm IBM MQ platformları için [İletiler ve neden kodları](#)

RPG Bildirimi

```

C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQINQMP(HCONN : HMSG : INQOPT :
                          PRNAME : PRPDSC : TYPE :
                          VALLEN : VALUE : DATLEN :
                          CMPCOD : REASON)

```

Çağrıya ilişkin prototip tanımlaması:

```

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**IBM i üzerinde MQMHBUF (İleti tanıtıcısını arabelleğe dönüştür)**

MQMHBUF, bir ileti tanıtıcısını arabelleğe dönüştürür ve MQBUFMH çağrısının tersidir.

- [“Sözdizimi” sayfa 1296](#)
- [“Kullanım notları” sayfa 1297](#)
- [“Parametreler” sayfa 1297](#)
- [“RPG Bildirimi” sayfa 1299](#)

Sözdizimi

MQMHBUF (*Hconn, Hmsg, MsgHBufOpts, Name, MsgDesc, BufferLength, Buffer, DataLength, CompCode, Reason*)

Kullanım notları

MQMHBUFF, ileti tanıtıcısını arabelleğe dönüştürür.

İleti özelliği API ' lerini kullanarak belirli özelliklere erişmek için MQGET API çıkımıyla bunu kullanabilir ve daha sonra, bu özellikleri ileti tanıtıcıları yerine MQRFH2 üstbilgilerini kullanmak üzere tasarlanmış bir uygulamaya geri aktarabilirsiniz.

Bu çağrı, ileti özelliklerini arabellekten ileti tanıtıcısına ayırtırmak için kullanabileceğiniz MQBUFMH çağrısının tersidir.

Parametreler

MQMHBUFF çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı iřaretli tamsayı)-giriř

Bu tanıtıcı, kuyruk yöneticisine yönelik baęlantıyı gösterir.

HCONN deęeri, HMSG deęiřtirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan baęlantı tanıtıcısıyla eřleřmelidir.

İleti tanıtıcısı HCUNAS kullanılarak yaratıldıysa, ileti tanıtıcısını silen iř parçacıęında geęerli bir baęlantı kurulmalıdır. Geęerli bir baęlantı kurulmazsa, arama RC2009ile başarısız olur.

HMSG (20 basamaklı iřaretli tamsayı)-giriř

Bu tanıtıcı, arabellek gerektiren ileti tanıtıcısıdır.

Deęer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

MHBOPT (MQMHBO)-giriř

MQMHBO yapısı, uygulamaların, arabelleklerin ileti tanıtıcılarından nasıl üretildięini denetleyen seęenekleri belirtmesine olanak saęlar.

Ayrıntılar için bkz. ["IBM i üzerinde MQBMHO \(Arabellek-ileti iřleme seęenekleri\)" sayfa 1002.](#)

PRNAME (MQCHARV)-giriř

Arabelleğe konacak özellięin ya da özelliklerin adı.

Adla eřleřen bir özellik bulunamazsa, arama RC2471ile başarısız olur.

jokerler

Arabelleğe birden çok özellik koymak için joker karakter kullanabilirsiniz. Bunu yapmak için özellik adının sonundaki yüzde iřaretini (%) kullanın. Bu genel arama karakteri, nokta (.) karakteri de içinde olmak üzere sıfır ya da daha fazla karakterle eřleřir.

Özellik adlarının kullanımıyla ilgili ek bilgi için [Özellik adları](#) ve [Özellik adı kısıtlamaları](#) konusuna bakın.

MSGDSC (MQMD)-giriř/çıkıř

MSGDSC yapısı, arabellek alanının içerięini açıklar.

Çıkıřta, *Encoding*, *CodedCharSetId* ve *Format* alanları, çağrı tarafından yazıldıęı gibi arabellek alanındaki verilerin kodlamasını, karakter kümesi tanıtıcısını ve biçimini doęru řekilde tanımlamak için ayarlanır.

Bu yapıdaki veriler, uygulamanın karakter kümesinde ve kodlamasında yer almaktadır.

BUFLEN (10 basamaklı iřaretli tamsayı)-giriř

BUFLEN , Arabellek alanının bayt cinsinden uzunluęudur.

BUFFER (1 baytlık bit dizilimi x BUFLEN)-giriř/çıkıř

BUFFER , ileti arabelleęini içeren alanı tanımlar. Çoęu veri için, arabelleęi 4 baytlık bir sınırdaki hizalayın.

BUFFER karakter ya da sayısal veri içeriyorsa, **MSGDSC** parametresindeki *CodedCharSetId* ve *Encoding* alanlarını veriler için uygun değerlere ayarlayın; bu, gerekirse verilerin dönüştürülmesini sağlar.

İleti arabelleğinde özellikler bulunursa, isteğe bağlı olarak bunlar kaldırılır; bunlar daha sonra çağrıdan döndüklerinde ileti tanıtıcısından kullanılabilir olur.

C programlama dilinde parametre, herhangi bir veri tipinin adresinin parametre olarak belirtileceği anlamına gelen bir işaretçi-to-void olarak bildirilir.

BUFLEN parametresi sıfırsa, *BUFFER* parametresine başvuruda bulunulmaz. Bu durumda, C ya da System/390 çevirici dilinde yazılmış programlardan geçirilen parametre adresi boş olabilir.

DATLEN (10 basamaklı işaretli tamsayı)-çıkış

DATLEN, arabellekte döndürülen özelliklerin bayt cinsinden uzunluğudur. Değer sıfırsa, hiçbir özellik *PRNAME* içinde belirtilen değerle eşleşmez ve çağrı RC2471neden koduyla başarısız olur.

BUFLEN, özellikleri arabellekte saklamak için gerekli uzunluktan azsa, MQMHBUF çağrısı RC2469ile başarısız olur, ancak *DATLEN*içine yine de bir değer girilir. Bu, uygulamanın özellikleri yerleştirmek için gereken arabellek büyüklüğünü saptamasını sağlar ve daha sonra, gerekli *BUFLEN*ile çağrıyı yeniden yürütür.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu; aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCFail ise:

RC2204

(2204, X'089C') Bağdaştırıcı kullanılmıyor.

RC2130

(2130, X'852 ') Bağdaştırıcı hizmet modülü yüklenemiyor.

RC2157

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

RC2501

(2501, X'095C') Arabellek seçenekleri yapısına ilişkin ileti tanıtıcısı geçersiz.

RC2004

(2004, X'07D4') Arabellek değiştirgesi geçersiz.

RC2005

(2005, X'07D5') Arabellek uzunluğu değiştirgesi geçersiz.

RC2219

(2219, X'08AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

RC2009

(2009, X'07D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2010

(2010, X'07DA') Veri uzunluğu parametresi geçersiz.

RC2460

(2460, X'099C') İleti tanıtıcısı geçersiz.

RC2026

(2026, X'07EA') İleti tanımlayıcısı geçersiz.

RC2499

(2499, X'09C3') İleti tanıtıcısı kullanımda.

RC2046

(2046, X'07FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2442

(2442, X'098A') Özellik adı geçerli değil.

RC2471

(2471, X'09A7') Özellik kullanılmıyor.

RC2469

(2469, X'09A5') BufferLength değeri, belirtilen özellikleri içermeyecek kadar küçük.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQMHBUF(HCONN : HMSG : MHOPT :
                          PRNAME : MSGDSC : BUFLen :
                          BUFFER : DATLEN :
                          CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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 MHOPT          12A
D* Property name
D PRNAME          32A
D* Message descriptor
D MSGDSC          364A
D* Length in bytes of the Buffer area
D BUFLen          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 IBM i üzerinde MQOPEN (nesneyi aç)

MQOPEN çağrısı bir nesneye erişim sağlar.

Aşağıdaki nesne tipleri geçerlidir:

- Kuyruk (dağıtım listeleri dahil)
- Ad listesi
- Süreç tanımlaması
- Kuyruk yöneticisi
- Konu

Dizin

- “Sözdizimi” sayfa 1300
- “Kullanım notları” sayfa 1300
- “Parametreler” sayfa 1304
- “RPG Bildirimi” sayfa 1310

Sözdizimi

MQOPEN (HCONN, OBJDSC, OPTS, HOBJ, CMPCOD, REASON)

Kullanım notları

1. Açılan nesne aşağıdakilerden biridir:

- Aşağıdaki işlemleri yapmak için bir kuyruk:
 - İletileri al ya da göz at (MQGET çağrısı kullanarak)
 - Koyma iletileri (MQPUT çağrılarını kullanarak)
 - Kuyruğun öznitelikleri hakkında bilgi alma (MQINQ çağrısının kullanılması)
 - Kuyruğun özniteliklerinin ayarlanması (MQSET çağrısının kullanılması)

Adı belirtilen kuyruk bir model kuyruğuysa, dinamik bir yerel kuyruk yaratılır.

Dağıtım listesi, kuyruk listesini içeren özel bir kuyruk nesnesi tipidir. İletileri koymak için açılabilir, ancak iletileri almak ya da göz atmak ya da öznitelikleri sorgulamak ya da ayarlamak için açılmaz. Daha fazla ayrıntı için kullanım notu 8'e bakın.

QSGDISP (GROUP) içeren bir kuyruk, MQOPEN ya da MQPUT1 çağrılarıyla kullanılmayacak özel bir kuyruk tanımlaması tipidir.

- Bir namelist, aşağıdakileri yapmak için:
 - Listedeki kuyrukların adlarını (MQINQ çağrılarını kullanarak) sorgulayabilirsiniz.
 - Aşağıdakileri yapmak için bir süreç tanımlaması:
 - Süreç özniteliklerini (MQINQ çağrısı kullanarak) sorgulamanızı sağlar.
 - Aşağıdaki işlemleri yapmak için kuyruk yöneticisi:
 - Yerel kuyruk yöneticisinin özniteliklerini (MQINQ çağrısının kullanılması) sorgulamanızı sağlar.
2. Bir uygulamanın aynı nesneyi bir kereden fazla açması geçerlidir. Her açık için farklı bir nesne tanıtıcısı döndürülür. Döndürülen her tanıtıcı, ilgili açma işlemi gerçekleştirildiği işlevler için kullanılabilir.
3. Açılmakta olan nesne bir kuyruksa, ancak bir küme kuyruğuysa, MQOPEN çağrısı sırasında yerel kuyruk yöneticisi içindeki tüm ad çözümlemesi gerçekleşir. Bu, belirli bir MQOPEN çağrısı için aşağıdakilerden birini ya da birkaçını içerebilir:
- Temel kuyruk adına diğer ad çözümlemesi
 - Uzak kuyruk yöneticisinin adına uzak kuyruk tanımının yerel tanımının çözülmesi ve kuyruğun uzak kuyruk yöneticisinde tanındığı ad
 - Uzak kuyruk yöneticisi adının yerel iletim kuyruğu adına çözülmesi

Ancak, tanıtıcı için sonraki MQINQ ya da MQSET çağrılarının yalnızca açılmış olan adla ilişkili olduğunu ve ad çözüldükten sonra sonuçlanan nesneyle değil. Örneğin, açılan nesne bir diğer adsa, MQINQ çağrısının döndürdüğü öznitelikler, diğer adın çözüldüğü temel kuyruğun öznitelikleri değil, diğer adın öznitelikleridir. Ad çözme denetimi, ilgili MQOPEN ' de **OPTS** değiştirilmesi için ne belirtildiğine bakılmaksızın devam etmektedir.

Açılmakta olan nesne bir küme kuyruğuysa, MQOPEN çağrısı sırasında ad çözme gerçekleşebilir ya da daha sonraya ertelenebilir. Çözümün oluştuğu nokta, MQOPEN çağrısında belirtilen OOBND* seçenekleriyle denetlenir:

- UUUUIM
- OOBNDN
- OOBNDQ

Küme kuyruklarına ilişkin ad çözümlemesi hakkında ek bilgi için [Ad çözünürlüğü](#) başlıklı konuya bakın.

4. Bir uygulamanın nesnesi açıkken nesnenin öznitelikleri değişebilir. Birçok durumda, uygulama bunu fark etmez, ancak belirli öznitelikler için kuyruk yöneticisi tanıtıcıyı artık geçerli değil olarak işaretler. Bunlar:

- Nesnenin ad çözünürlüğünü etkileyen herhangi bir öznitelik. Bu, kullanılan açık seçeneklerden bağımsız olarak geçerlidir ve aşağıdakileri içerir:
 - Açık olan bir diğer ad kuyruğunun **BaseQName** özniteliğinde yapılan bir değişiklik.
 - **RemoteQName** ya da **RemoteQMgrName** kuyruk özniteliklerinde, bu kuyruk için açık olan herhangi bir tanıtıcıya ya da kuyruk yöneticisi diğer adı olarak bu tanımlamayla çözülen bir kuyruğa ilişkin değişiklik.
 - Uzak bir kuyruğa ilişkin şu anda açık olan bir tutamacın farklı bir *iletim* kuyruğuna çözümlenmesine ya da bir tanesine çözülememesine neden olan herhangi bir değişiklik. Örneğin, bu şunları içerebilir:
 - Uzak bir kuyruğun yerel tanımlamasının **XmitQName** özniteliğinde, tanımlamanın bir kuyruk için mi, yoksa kuyruk yöneticisi diğer adı için mi kullanıldığına ilişkin bir değişiklik.

Bunun bir istisnası vardır, yani yeni bir iletim kuyruğunun yaratılması. Tanıtıcı açıldığında bu kuyruğa çözülmüş olan, ancak varsayılan iletim kuyruğuna çözülmüş olan bir tanıtıcı geçersiz kılmaz.
 - **DefXmitQName** kuyruk yöneticisi özniteliğinde değişiklik. Bu durumda, önceden adlandırılan kuyruğa çözülen (yalnızca varsayılan iletim kuyruğu olduğu için çözülen) tüm açık tanıtıcıları geçersiz olarak işaretlenir. Başka nedenlerle bu kuyruğa çözülen tanıtıcıları etkilenmez.
- Bu kuyruk için ya da bu kuyruğa çözülen bir kuyruk için OOINPS erişimi sağlayan iki ya da daha fazla tanıtıcı varsa, **Shareability** kuyruk özniteliği. Bu durumda, bu kuyruk için ya da bu kuyruğa çözülen bir kuyruk için açık olan *tüm* tanıtıcıları, açık seçeneklerden bağımsız olarak geçersiz olarak işaretlenir.
- Açık seçeneklerden bağımsız olarak, bu kuyruk için ya da bu kuyruğa çözülen bir kuyruk için açık olan tüm tanıtıcıları içeren **Usage** kuyruk özniteliği.

Bir tanıtıcı geçersiz olarak işaretlendiğinde, bu tanıtıcıyı kullanan sonraki tüm çağrılar (MQCLOSE dışında) RC2041; uygulama bir MQCLOSE çağrısı yayınlamalı (özgün tanıtıcıyı kullanarak) ve kuyruğu yeniden açmalıdır. Önceki başarılı çağrılarının eski tanıtıcısıyla ilgili kesinleştirilmemiş güncellemeler, uygulama mantığının gerektirdiği şekilde, yine de kesinleştirilebilir ya da geriletebilir.

Bir özniteliğin değiştirilmesi bunun gerçekleşmesine neden olarsa, komutun özel bir "force" sürümü kullanılmalıdır.

5. Kuyruk yöneticisi, bir MQOPEN çağrısı yayınlandığında, uygulamanın çalıştırdığı kullanıcı kimliğinin erişime izin verilmeden önce uygun yetki düzeyine sahip olduğunu doğrulamak için güvenlik denetimleri gerçekleştirir. Yetki denetimi, bir ad çözüldükten sonra ortaya çıkan, adı ya da adları değil, açılmakta olan nesnenin adı üzerinde yapılır.

Açılmakta olan nesne bir model kuyruğuysa, kuyruk yöneticisi hem model kuyruğunun adı, hem de yaratılan dinamik kuyruğun adı için tam güvenlik denetimi gerçekleştirir. Sonuçta ortaya çıkan dinamik kuyruk belirtik olarak açılırsa, dinamik kuyruğun adına göre başka bir kaynak güvenliği denetimi gerçekleştirilir.

6. Uzak kuyruk, bu çağrıya ilişkin **OBJDSC** değiştirgesinde iki yoldan biriyle belirtilebilir (bkz. "[IBM i üzerinde MQOD \(Nesne tanımlayıcı\)](#)" sayfa 1141 içinde açıklanan *ODON* ve *ODMN* alanları):

- *ODON* için uzak kuyruğun yerel tanımının adını belirleyerek. Bu durumda, *ODMN* yerel kuyruk yöneticisini belirtir ve boşluk olarak belirtilebilir.

Yerel kuyruk yöneticisi tarafından gerçekleştirilen güvenlik doğrulaması, kullanıcının uzak kuyruğun yerel tanımını açma yetkisinin olduğunu doğrular.

- Uzak kuyruk yöneticisi tarafından bilinen uzak kuyruğun adını *ODON* olarak belirterek. Bu durumda, *ODMN* uzak kuyruk yöneticisinin adıdır.

Yerel kuyruk yöneticisi tarafından gerçekleştirilen güvenlik doğrulaması, kullanıcının ad çözme işleminden kaynaklanan iletim kuyruğuna ileti gönderme yetkisi olduğunu doğrular.

Her iki durumda da:

- Kullanıcının kuyruğa ileti koyma yetkisi olup olmadığını denetlemek için, yerel kuyruk yöneticisi tarafından uzak kuyruk yöneticisine ileti gönderilmez.
 - Uzak kuyruk yöneticisine bir ileti geldiğinde, uzak kuyruk yöneticisi iletiyi oluşturan kullanıcının yetkisi olmadığı için iletiyi reddedebilir.
7. OOB_{RW} seçeneğiyle bir MQOPEN çağrısı, nesne tanıtıcısını ve göz atma seçeneklerinden birini belirten MQGET çağrılarını kullanarak kullanılmak üzere bir göz atma geçici çizelgesi oluşturur. Bu, kuyruğun içeriğini değiştirmeden taranmasını sağlar. Göz atma yoluyla bulunan bir ileti daha sonra GMMUC seçeneği kullanılarak kuyruktan kaldırılabilir.

Aynı kuyruk için birden çok MQOPEN isteği yayınlanarak, tek bir uygulama için birden çok göz atma işlemleri etkin olabilir.

8. Aşağıdaki notlar dağıtım listelerinin kullanımı için geçerlidir.

- Bir dağıtım listesi açılırken MQOD yapısındaki alanlar aşağıdaki gibi ayarlanmalıdır:
 - *ODVER* *ODVER2* ya da daha büyük olmalıdır.
 - *ODOT* *OTQ* olmalıdır.
 - *ODON* boş değerli dizgi ya da boş değerli dizgi olmalıdır.
 - *ODMN* boş değerli dizgi ya da boş değerli dizgi olmalıdır.
 - *ODREC* sıfırdan büyük olmalıdır.
 - *ODORO* ve *ODORP* değerlerinden biri sıfır, diğeri sıfır olmayan olmalıdır.
 - *ODRRO* ve *ODRRP* değerlerinden en fazla biri sıfır dışında olabilir.
 - *ODORO* ya da *ODORP* tarafından adreslenen *ODREC* nesne kayıtları olmalıdır. Nesne kayıtları, açılacak hedef kuyrukların adlarına ayarlanmalıdır.
 - *ODRRO* ve *ODRRP* değerlerinden biri sıfır değilse, *ODREC* yanıt kaydı olmalıdır. Çağrı RC2136neden koduyla tamamlanırsa kuyruk yöneticisi tarafından ayarlanır.

version-2 MQOD, *ODREC* ' in sıfır olduğundan emin olarak, dağıtım listesinde olmayan tek bir kuyruğu açmak için de kullanılabilir.

- **OPTS** değiştirilmesinde yalnızca aşağıdaki açma seçenekleri geçerlidir:
 - OOOOUT!
 - OOPAS*
 - OOOSET*
 - OOALTU
 - OOFIQ
- Dağıtım listesindeki hedef kuyruklar yerel, diğer ad ya da uzak kuyruklar olabilir, ancak bunlar model kuyrukları olamaz. Bir model kuyruğu belirtilirse, o kuyruk RC2057neden koduyla açılmaz. Ancak bu, listedeki diğer kuyrukların başarıyla açılmasını önlemez.
- Tamamlanma kodu ve neden kodu değiştirilmesi aşağıdaki gibi ayarlanır:
 - Dağıtım listesindeki kuyruklara ilişkin açma işlemleri aynı şekilde başarılı ya da başarısız olursa, tamamlanma kodu ve neden kodu parametreleri ortak sonucu tanımlayacak şekilde ayarlanır. Bu durumda MQRR yanıt kayıtları (uygulama tarafından sağlandıysa) ayarlanmaz.
Örneğin, her açık başarılı olursa, tamamlama kodu CCOK olarak ayarlanır ve neden kodu RCNONE olur; kuyrukların hiçbiri var olmadığı için her açık başarısız olursa, parametreler CCFAIL ve RC2085olarak ayarlanır.

- Dağıtım listesindeki kuyruklara ilişkin açma işlemleri aynı şekilde başarılı olmazsa ya da başarısız olursa:
 - En az bir açma başarılı olursa, tamamlanma kodu parametresi CCWARN olarak ve tümü başarısız olursa CCFAIL olarak ayarlanır.
 - Neden kodu parametresi RC2136olarak ayarlandı.
 - Yanıt kayıtları (uygulama tarafından sağlandıysa), dağıtım listesindeki kuyruklara ilişkin tek tek tamamlanma kodlarına ve neden kodlarına ayarlanır.

- Bir dağıtım listesi başarıyla açıldığında, çağrı tarafından döndürülen *HOBJ* tanıtıcısı, dağıtım listesindeki kuyruklara ileti koymak için sonraki MQPUT çağrılarında ve dağıtım listesine erişimi bırakmaya ilişkin bir MQCLOSE çağrısında kullanılabilir. Dağıtım listesi için geçerli olan tek kapanış seçeneği CONONE.

MQPUT1 çağrısı bir iletiyi dağıtım listesine koymak için de kullanılabilir; listedeki kuyrukları tanımlayan MQOD yapısı, o çağrıda bir değiştirge olarak belirtilir.

- Dağıtım listesinde başarıyla açılan her hedef, uygulamanın izin verilen tanıtıcı sayısı üst sınırını aşmış olduğunu denetlerken *ayrı* bir tanıtıcı olarak sayılır (**MaxHandles** kuyruk yöneticisi özniteliğine bakın). Bu, dağıtım listesindeki iki ya da daha fazla hedef aynı fiziksel kuyruğa çözüldüğünde bile geçerlidir. Bir dağıtım listesine ilişkin MQOPEN ya da MQPUT1 çağrısı, uygulama tarafından kullanılan tanıtıcı sayısının *MaxHandles*değerini aşmasına neden olursa, çağrı başarısız olur; neden kodu RC2017.
- Başarıyla açılan her hedefin **OpenOutputCount** özniteliğinin değeri bir artırıldı. Dağıtım listesindeki hedeflerden ikisi ya da daha fazlası gerçekte aynı fiziksel kuyruğa çözümlenirse, o kuyruğun **OpenOutputCount** özniteliği, dağıtım listesindeki o kuyruğa çözülen hedef sayısı ile artırılır.
- Kuyruk tanımlamalarında, kuyruklar ayrı ayrı açıldığında (örneğin, çözüm yolundaki bir değişiklik), tanıtıcı geçersiz duruma gelmesine neden olacak herhangi bir değişiklik, dağıtım listesi tanıtıcılarının geçersiz olmasına neden olmaz. Ancak, dağıtım listesi tanıtıcısı sonraki bir MQPUT çağrısında kullanıldığında, o kuyruk için bir hata oluşmasına neden olur.
- Bir dağıtım listesinin tek bir hedef içermesi geçerlidir.

9. Aşağıdaki notlar küme kuyruklarının kullanımı için geçerlidir.

- Bir küme kuyruğu ilk kez açıldığında ve yerel kuyruk yöneticisi tam havuz kuyruğu yöneticisi olmadığında, yerel kuyruk yöneticisi küme kuyruğuna ilişkin bilgileri tam havuz kuyruk yöneticisinden alır. Ağ meşgul olduğunda, yerel kuyruk yöneticisinin havuz kuyruk yöneticisinden gereken bilgileri alması birkaç saniye sürebilir. Sonuç olarak, MQOPEN çağrısı yayınlayan uygulamanın, denetim MQOPEN çağrısından dönmeden önce en çok 10 saniye beklemesi gerekebilir. Yerel kuyruk yöneticisi bu süre içinde küme kuyruğuna ilişkin gerekli bilgileri almazsa, çağrı RC2189neden koduyla başarısız olur.
- Bir küme kuyruğu açıldığında ve kümede kuyruğun birden çok eşgörünümü olduğunda, gerçekte açılan eşgörünüm MQOPEN çağrısında belirtilen seçeneklere bağlıdır:

- Belirtilen seçenekler aşağıdakilerden herhangi birini içerirse:

- OOBW
- OOINPQ
- OOINPX
- OOINPS
- OOSSET!

Açılan küme kuyruğunun yönetim ortamının yerel yönetim ortamı olması gerekir. Kuyruğun yerel eşgörünümü yoksa, MQOPEN çağrısı başarısız olur.

- Belirtilen seçenekler yukarıdakilerin hiçbirini içermezse, ancak aşağıdakilerden birini ya da her ikisini de içerirse:
 - OOINQ
 - OOOOUT!

Açılan yönetim ortamı, varsa yerel yönetim ortamıyla, tersi durumda uzak yönetim ortamıyla olur. Ancak, kuyruk yöneticisi tarafından seçilen yönetim ortamı bir küme iş yükü çıkışıyla (varsa) değiştirilebilir.

Küme kuyruklarıyla ilgili ek bilgi için [Küme kuyrukları](#) başlıklı konuya bakın.

10. Bir tetikleyici izleme programı tarafından başlatılan uygulamalara, uygulama başlatıldığında uygulamayla ilişkili kuyruğun adı geçirilir. Bu kuyruk adı, kuyruğu açmak için **OBJDSC** parametresinde belirtilebilir. Daha fazla ayrıntı için MQTMC yapısının tanımına bakın.
11. OORLOQ seçeneği kullanılırken, yerel kuyruk ya da model kuyruğu açıldığında önceden döndürülür; ancak, örneğin, uzak bir kuyruk ya da yerel olmayan bir küme kuyruğu açıldığında bu durum olmaz; ResolvedQName ve ResolvedQMgradı RemoteQName ve RemoteQMgradı uzak kuyruk tanımlamasında ya da seçilen uzak küme kuyruğuyla benzer şekilde girilir. Açılırken OORLOQ belirtilirse, örneğin bir uzak kuyruk, ResolvedQName iletilerin yerleştirileceği iletim kuyruğu olur. ResolvedQMgradı, iletim kuyruğunu barındıran yerel kuyruk yöneticisinin adıyla girilir. Bir kullanıcının bir kuyrukta göz atma, giriş ya da çıkış için yetkisi varsa, MQOPEN çağrısında bu işareti belirtmek için gerekli yetkiye sahiptir. Özel bir yetkiye gerek yoktur.

Parametreler

MQOPEN çağrısındaki değiştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. HCONN değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

OBJDSC (MQOD)-giriş/çıkış

Nesne tanımlayıcı.

Bu, açılacak nesneyi tanımlayan bir yapıdır; ayrıntılar için bkz. [“IBM i üzerinde MQOD \(Nesne tanımlayıcı\)” sayfa 1141](#).

OBJDSC parametresindeki *ODON* alanı bir model kuyruğunun adıyla, dinamik bir yerel kuyruk model kuyruğunun öznitelikleriyle yaratılır; bu, **OPTS** değiştirgesiyle belirtilen açma seçeneklerinden bağımsız olarak gerçekleşir. MQOPEN çağrısıyla döndürülen *HOBJ* kullanılarak yapılan sonraki işlemler, model kuyruğunda değil, yeni dinamik kuyrukta gerçekleştirilir. Bu, MQINQ ve MQSET çağrıları için bile geçerlidir. **OBJDSC** değiştirgesindeki model kuyruğunun adı, yaratılan dinamik kuyruğun adıyla değiştirilir. Dinamik kuyruğun tipi, model kuyruğunun **DefinitionType** özniteliklerinin değerine göre belirlenir (bkz. [“Kuyruklara ilişkin öznitelikler” sayfa 1345](#)). Dinamik kuyruklar için geçerli kapatma seçenekleriyle ilgili bilgi için MQCLOSE çağrısının tanımına bakın.

OPTS (10 basamaklı işaretli tamsayı)-giriş

MQOPEN işlemini denetleyen seçenekler.

Aşağıdaki seçeneklerden en az biri belirtilmelidir:

- OOBW
- OOINP* (bunlardan yalnızca biri)
- OOINQ
- OOOOUT!
- OOSSET!
- OORLQ

Diğer seçenekler gerekli olarak belirtilebilir. Birden çok seçenek gerekiyorsa, değerler eklenebilir (aynı değişmezi bir kereden fazla eklemeyin). Geçerli olmayan birleşimler belirtilir; diğer tüm birleşimler geçerlidir. Yalnızca, *OBJDSC* ile belirtilen nesne tipi için geçerli olan seçeneklere izin verilir ([Her kuyruk tipi için geçerli MQOPEN seçenekleri](#) konusuna bakın).

Erişim seçenekleri: Aşağıdaki seçenekler, nesne üzerinde gerçekleştirilebilecek işlemlerin tipini denetler:

OOINPQ

Kuyruk tanımlı varsayılan değeri kullanarak ileti almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Erişim tipi, **DefInputOpenOption** kuyruk özneliğinin değerine bağlı olarak paylaşılır ya da dışlayıcıdır; ayrıntılar için bkz. [“Kuyruklara ilişkin öznelikler” sayfa 1345](#).

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir.

OOINPS

Paylaşılan erişimi olan iletileri almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Kuyruk bu ya da OOINPS içeren başka bir uygulama tarafından açıksa, ancak kuyruk şu anda OOINPX ile açıksa RC2042 neden koduyla başarısız olursa çağrı başarılı olabilir.

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir.

OOINPX

Dışlayıcı erişimi olan iletileri almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Kuyruk şu anda bu ya da başka bir uygulama tarafından herhangi bir tip giriş (OOINPS ya da OOINPX) için açıksa, çağrı RC2042 neden koduyla başarısız olur.

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir.

Aşağıdaki notlar bu seçenekler için geçerlidir:

- Bu seçeneklerden yalnızca biri belirtilebilir.
- **InhibitGet** kuyruk özneliği QAGETI olarak ayarlanmış olsa da (öznelik bu değere ayarlandığında sonraki MQGET çağrıları başarısız olur), bu seçeneklerden birine sahip bir MQOPEN çağrısı başarılı olabilir.
- Kuyruk paylaşılabilir olarak tanımlanmazsa (yani, **Shareability** kuyruk özneliği QANSHR değerine sahiptir), kuyruğu paylaşılabilir erişim için açma girişimleri, kuyruğu dışlayıcı erişimle açma girişimleri olarak kabul edilir.
- Bu seçeneklerden biriyle bir diğer ad kuyruğu açılırsa, dışlayıcı kullanım sınavı (ya da başka bir uygulamanın özel kullanımı olup olmadığı), diğer adın çözümleneceği temel kuyruğa karşı olur.
- **ODMN** bir kuyruk yöneticisi diğer adının adıysa bu seçenekler geçerli değildir; kuyruk yöneticisi diğer adı için kullanılan uzak kuyruğun yerel tanımında **RemoteQMgrName** özneliğinin değeri yerel kuyruk yöneticisinin adı olsa bile bu geçerlidir.

OBRW

İletilere göz atmak için kuyruğu açın.

Kuyruk, aşağıdaki seçeneklerden biriyle sonraki MQGET çağrılarıyla kullanılmak üzere açılır:

- GMBRWF
- GMBRWN
- GMBRWC

Kuyruk şu anda OOINPX için açık olsa da buna izin verilir. OBRW seçeneğiyle MQOPEN çağrısı bir göz atma imlecini oluşturur ve mantıksal olarak kuyruktaki ilk iletiden önce konumlandırır; ek bilgi için [“IBM i üzerinde MQGMO \(Get-message seçenekleri\)” sayfa 1060](#) içinde açıklanan **GMOPT** alanına bakın.

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir. **ODMN** bir kuyruk yöneticisi diğer adının

adıysa da geçerli değildir; uzak kuyruk yöneticisi diğer adı için kullanılan uzak kuyruk yöneticisinin yerel tanımındaki **RemoteQMgrName** özniteliğinin değeri yerel kuyruk yöneticisinin adı olsa da bu geçerlidir.

OOOOUT!

İletileri koymak için kuyruğu açın ya da iletileri yayınlamak için bir konu ya da konu dizesini açın.

Kuyruk, sonraki MQPUT çağrılarıyla kullanılmak üzere açılır.

Bu seçenekle bir MQOPEN çağrısı, **InhibitPut** kuyruk özniteliği QAPUTI olarak ayarlansa bile başarılı olabilir (öznitelik bu değere ayarlandığında sonraki MQPUT çağrıları başarısız olur).

Bu seçenek, dağıtım listeleri ve konular da içinde olmak üzere, tüm kuyruk tipleri için geçerlidir.

OOINQ

Öznitelikleri sorgulamak için nesneyi açın.

Kuyruk, ad listesi, süreç tanımlaması ya da kuyruk yöneticisi sonraki MQINQ çağrılarıyla kullanılmak üzere açılır.

Bu seçenek, dağıtım listeleri dışındaki tüm nesne tipleri için geçerlidir. *ODMN* bir kuyruk yöneticisi diğer adının adıysa geçerli değildir; kuyruk yöneticisi için kullanılan uzak kuyruğun yerel tanımında **RemoteQMgrName** özniteliğinin değeri yerel kuyruk yöneticisinin adı olsa da bu geçerlidir.

OOSSET!

Öznitelikleri ayarlamak için kuyruğu açın.

Kuyruk, sonraki MQSET çağrılarıyla kullanılmak üzere açılır.

Bu seçenek, dağıtım listeleri dışındaki tüm kuyruk tipleri için geçerlidir. *ODMN* uzak bir kuyruğun yerel tanımının adıdır; kuyruk yöneticisi için kullanılan uzak kuyruğun yerel tanımında **RemoteQMgrName** özniteliğinin değeri yerel kuyruk yöneticisinin adı olsa bile bu geçerlidir.

Bağ tanımlama seçenekleri: Açılmakta olan nesne bir küme kuyruğuyken aşağıdaki seçenekler geçerlidir; bu seçenekler, kuyruk tanıtıcısının küme kuyruğunun bir eşgörünümüne bağlanmasını denetler:

UUUUUIM

Kuyruk açıldığında hedef için bağ tanımlama tanıtıcısı.

Bu, kuyruk açıldığında, yerel kuyruk yöneticisinin kuyruk tanıtıcısını hedef kuyruğun bir eşgörünümüne bağlamasına neden olur. Sonuç olarak, bu tanıtıcı kullanılarak konan tüm iletiler hedef kuyruğun aynı eşgörünümüne ve aynı rotaya gönderilir.

Bu seçenek yalnızca kuyruklar için geçerlidir ve yalnızca küme kuyruklarını etkiler. Küme kuyruğu olmayan bir kuyruk için belirtildiyse, seçenek yoksayılr.

OOBNDN

Belirli bir hedefe bağlanmayın.

Bu, yerel kuyruk yöneticisinin kuyruk tanıtıcısını hedef kuyruğun bir eşgörünümüne bağlamasını durdurur. Sonuç olarak, bu tanıtıcıyı kullanan ardışık MQPUT çağrıları, iletilerin hedef kuyruğun *farklı* eşgörünümlerine gönderilmesine ya da aynı yönetim ortamına, ancak farklı rotalarla gönderilmesine neden olabilir. Ayrıca, seçilen yönetim ortamının daha sonra yerel kuyruk yöneticisi, uzak kuyruk yöneticisi ya da bir ileti kanalı aracısı (MCA) tarafından ağ koşullarına göre değiştirilmesini sağlar.

Not: Bir hareketi tamamlamak için *dizi* ileti göndermesi gereken istemci ve sunucu uygulamaları, *DefBind* BNDNOT değerine sahip olduğunda OOBNDN (ya da OOBNDQ) kullanmamalıdır; çünkü dizideki ardışık iletiler sunucu uygulamasının farklı örneklerine gönderilebilir.

Bir küme kuyruğu için OOBRW ya da OOINP* seçeneklerinden biri belirtilirse, kuyruk yöneticisi küme kuyruğunun yerel yönetim ortamını seçmeye zorlanır. Sonuç olarak, OOBNDN belirlenmiş olsa bile, kuyruk tanıtıcısının bağ tanımı düzeltilmiştir.

OOBNDN ile OOINQ belirtilirse, bu tanıtıcıyı kullanan ardışık MQINQ çağruları küme kuyruğunun farklı eşgörünümlerini sorgulayabilir, ancak genellikle tüm eşgörünümler aynı öznitelik değerlerine sahiptir.

OOBNDN yalnızca kuyruklar için geçerlidir ve yalnızca küme kuyruklarını etkiler. Küme kuyruğu olmayan bir kuyruk için belirtildiyse, seçenek yoksayılr.

OOBNDQ

Kuyruk için varsayılan bağ tanımını kullan.

Bu, yerel kuyruk yöneticisinin kuyruk tanıtıcısını **DefBind** kuyruk özniteliği tarafından tanımlanan şekilde bağlamasına neden olur. Bu özniteliğin değeri BNDOPN ya da BNDNOT.

OOBNUDO ve OOBNDN belirtilmezse, varsayılan değer OOBNDQ olur.

OOBNDQ, program belgelerine yardımcı olmak için tanımlanır. Bu seçeneğin diğer iki bağ tanımlama seçeneğinden biriyle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

Bağlam seçenekleri: Aşağıdaki seçenekler ileti bağlamının işlenmesini denetler:

OOSAVA.

İleti alındığında bağlamı kaydet.

Bağlam bilgileri bu kuyruk tanıtıcısıyla ilişkilendirilir. Bu bilgi, bu tanıtıcı kullanılarak alınan tüm iletilerin bağlamından ayarlanır. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

Bu bağlam bilgileri, daha sonra MQPUT ya da MQPUT1 çağruları kullanılarak kuyruğa konan bir iletiye iletilebilir. [“IBM i üzerinde MQPMO \(Put-message options\)” sayfa 1154](#) içinde açıklanan PMPASI ve PMPASA seçeneklerine bakın.

Bir ileti başarıyla alınıncaya kadar, bağlam kuyruğa konan bir iletiye geçirilemez.

GMBRW* göz atma seçeneklerinden biri kullanılarak alınan bir iletinin bağlam bilgileri saklanmaz (ancak, **MSGDSC** değiştirgesindeki bağlam alanları göz atma işleminden sonra ayarlanır).

Bu seçenek yalnızca yerel, diğer ad ve model kuyrukları için geçerlidir; uzak kuyruklar, dağıtım listeleri ve kuyruk olmayan nesnelere için geçerli değildir. OOINP* seçeneklerinden biri belirtilmelidir.

OOPASI

Kimlik bağlamının iletilmesine izin ver.

Bu, kuyruğa bir ileti bulunduğu **PMO** parametresinde PMPASI seçeneğinin belirlenmesini sağlar; bu, iletiye OOSAVA seçeneğiyle açılan bir giriş kuyruğundan kimlik bağlamı bilgilerini verir. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

OOOUT seçeneği belirtilmelidir.

Bu seçenek, dağıtım listeleri de içinde olmak üzere, tüm kuyruk tipleri için geçerlidir.

OOPASA

Tüm bağlamın iletilmesine izin ver

Bu, kuyruğa bir ileti bulunduğu PMPASA seçeneğinin **PMO** parametresinde belirtilmesini sağlar; bu, iletiye OOSAVA seçeneğiyle açılan bir giriş kuyruğundan kimlik ve kaynak bağlam bilgilerini verir. İleti bağlamıyla ilgili daha fazla bilgi için [İleti bağlamı](#) ve [Bağlam bilgilerini denetleme](#) başlıklı konuya bakın.

Bu seçenek, belirtilmesi gerekmeyen OOPASI seçeneğini belirtir. OOOUT seçeneği belirtilmelidir.

Bu seçenek, dağıtım listeleri de içinde olmak üzere, tüm kuyruk tipleri için geçerlidir.

OOSSETI

Kimlik bağlamının ayarlanmasına izin ver.

Bu, kuyruğa bir ileti konduğunda **PMO** deęiřtirgesinde PMSETI seeneęinin belirtilmesini saęlar; bu, iletiye MQPUT ya da MQPUT1 aęrısında belirtilen **MSGDSC** deęiřtirgesinde bulunan kimlik baęlamı bilgilerini verir. İleti baęlamıyla ilgili daha fazla bilgi için İleti baęlamı ve Baęlam bilgilerini denetlemebařlıklı konuya bakın.

Bu seenek, belirtilmesi gerekmeyen OOPASI seeneęini belirtir. OOOOT seeneęi belirtilmelidir.

Bu seenek, daęıtım listeleri de iinde olmak üzere, tüm kuyruk tipleri iin geerlidir.

OOSETA.

Tüm baęlamın ayarlanmasına izin ver.

Bu, kuyruğa bir ileti konduğunda **PMO** deęiřtirgesinde PMSETA seeneęinin belirtilmesini saęlar; bu, iletiye MQPUT ya da MQPUT1 aęrısında belirtilen **MSGDSC** deęiřtirgesinde bulunan kimlik ve kaynak baęlam bilgilerini verir. İleti baęlamıyla ilgili daha fazla bilgi için İleti baęlamı ve Baęlam bilgilerini denetlemebařlıklı konuya bakın.

Bu seenek, belirtilmesi gerekmeyen ařaęıdaki seenekleri belirtir:

- OOPASI
- OOPASA
- OOSETI

OOOOUT seeneęi belirtilmelidir.

Bu seenek, daęıtım listeleri de iinde olmak üzere, tüm kuyruk tipleri iin geerlidir.

Dięer seenekler: Ařaęıdaki seenekler yetki denetimini denetler ve kuyruk yneticisi susturulduğunda ne olur:

OOALTU

Belirtilen kullanıcı kimlięiyle doęrulayın.

Bu, **OBJDSC** deęiřtirgesindeki *ODAU* alanının, bu MQOPEN aęrısının geerlilięini denetlemek iin kullanılacak bir kullanıcı kimlięi ierdięini gsterir. aęrı, yalnızca bu *ODAU* ' in belirtilen eriřim seenekleriyle nesneyi ama yetkisi varsa, uygulamanın alıřtıęı kullanıcı kimlięinin bunu yapma yetkisi olup olmadıęına bakılmaksızın bařarılı olabilir. Bu, belirtilen baęlam seenekleri iin geerli deęildir; ancak, her zaman uygulamanın alıřtıęı kullanıcı kimlięiyle karřılařtırılarak denetlenir.

Bu seenek, tüm nesne tipleri iin geerlidir.

OOFIQ

Kuyruk yneticisi susturulursa bařarısız olur.

Bu seenek, kuyruk yneticisi susturma durumundaysa MQOPEN aęrıyla bařarısız etmeye zorlar.

Bu seenek, tüm nesne tipleri iin geerlidir.

OORLQ

Aılan yerel kuyruęun adını girin.

Bu seenek, MQOD yapısındaki (varsa) ResolvedQName deęerinin, aılan yerel kuyruęun adıyla girilmesi gerektięini belirtir. ResolvedQMGrAdı, yerel kuyruęu barındıran yerel kuyruk yneticisinin adıyla benzer Őekilde girilir.

<i>izelge 750. Her kuyruk tipi iin geerli MQOPEN seenekleri</i>						
Seenek	Dięer Ad ("1" sayfa 1309)	Yerel ve Model	Uzak	Yerel Olmayan Küme	Daęıtım listesi	Konu
OOINPQ	✓	✓	-	-	-	-
OOINPS	✓	✓	-	-	-	-

Çizelge 750. Her kuyruk tipi için geçerli MÇOPEN seçenekleri (devamı var)						
Seçenek	Diğer Ad (<u>"1" sayfa 1309</u>)	Yerel ve Model	Uzak	Yerel Olmayan Küme	Dağıtım listesi	Konu
OOINPX	✓	✓	-	-	-	-
OBRW	✓	✓	-	-	-	-
OOOUT!	✓	✓	✓	✓	✓	✓
OOINÇ	✓	✓	<u>"2" sayfa 1309</u>	✓	-	-
OSET!	✓	✓	<u>"2" sayfa 1309</u>	-	-	-
OUBUDU (<u>"3" sayfa 1309</u>)	✓	✓	✓	✓	✓	-
OBNNDN (<u>"3" sayfa 1309</u>)	✓	✓	✓	✓	✓	-
OBNNDÇ (<u>"3" sayfa 1309</u>)	✓	✓	✓	✓	✓	-
OOSAVA.	✓	✓	-	-	-	-
OOPASI	✓	✓	✓	✓	✓	<u>"5" sayfa 1309</u>
OOPASA	✓	✓	✓	✓	✓	<u>"5" sayfa 1309</u>
OSETI	✓	✓	✓	✓	✓	<u>"5" sayfa 1309</u>
OSETA.	✓	✓	✓	✓	✓	<u>"5" sayfa 1309</u>
OALTU	✓	✓	✓	✓	✓	✓
OOFIÇ	✓	✓	✓	✓	✓	✓
OORLÇ	✓	✓	✓	✓	-	-

Notlar:

1. Diğer adlara ilişkin seçeneklerin geçerliliği, diğer adın çözümleneceği kuyruğa ilişkin seçeneğin geçerliliğine bağlıdır.
2. Bu seçenek yalnızca uzak kuyruğun yerel tanımlaması için geçerlidir.
3. Bu seçenek herhangi bir kuyruk tipi için belirlenebilir, ancak kuyruk bir küme kuyruğu değilse yoksayılır.
4. Bu öznitelik bir konu için yoksayıldı.
5. Bu öznitelikler bir konuyla birlikte kullanılabilir, ancak herhangi bir aboneye gönderilen bağlam alanlarını değil, yalnızca alıkonan ileti için ayarlanan bağlamı etkiler.

HOBJ (10 basamaklı işaretli tamsayı)-çıkış

Nesne tanıtıcısı.

Bu tanıtıcı, nesne için oluşturulan erişimi gösterir. Nesne üzerinde işlem yapan sonraki ileti kuyruklama çağrılarında belirtilmelidir. MQCLOSE çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde geçerli olmaktan çıkar.

Tutamacın kapsamı, Uygulamanın çalıştığı platform tarafından desteklenen koşut işleme; tanıtıcı, MQOPEN çağrısının verildiği koşut işleme biriminin dışında geçerli değil:

- IBM i' de, tanıtıcı kapsamı, çağrıyı yayınlayan iştir.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQOPEN(HCONN : OBJDSC : OPTS :
C                               HOBJ : CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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 üzerinde MQPUT (Put message)

MQPUT çağrısı bir iletiyi bir kuyruğa, dağıtım listesine ya da bir konuya koyar. Kuyruk, dağıtım listesi ya da konu açık olmalıdır.

- [“Sözdizimi” sayfa 1311](#)
- [“Kullanım notları” sayfa 1311](#)
 - [“Konular” sayfa 1311](#)
 - [“MQPUT ve MQPUT1” sayfa 1311](#)
 - [“Hedef kuyruklar” sayfa 1311](#)
 - [“Dağıtım listeleri” sayfa 1312](#)
 - [“Üst Bilgiler” sayfa 1314](#)
 - [“Arabellek” sayfa 1314](#)
- [“Parametreler” sayfa 1314](#)

- “RPG Bildirimi” sayfa 1319

Sözdizimi

MQPUT (HCONN, HOBJ, MSGDSC, PMO, BUFLN, BUFFER, CMPCOD, REASON)

Kullanım notları

Konular

Aşağıdaki notlar konuların kullanımı için geçerlidir:

1. Bir konuda ileti yayınlamak için MQPUT kullanılırken, abone kuyruğuyla ilgili bir sorun (örneğin, dolu) nedeniyle o konuya ilişkin bir ya da daha çok aboneye yayın verilemediğinde, MQPUT çağrısına döndürülen neden kodu ve teslim davranışı, TOPIC üzerindeki PMSGDLV ya da NPMSGDLV özniteliklerinin ayarına bağlıdır. RODLQ belirtildiğinde bir yayının teslim edilmesinin ya da RODISC belirtildiğinde iletinin atılmasının iletinin başarılı bir şekilde teslim edilmesi olarak kabul edileceğini unutmayın. Yayınların hiçbiri teslim edilmezse, MQPUT RC2502 ile döndürülür. Bu, aşağıdaki durumlarda gerçekleşebilir:
 - Bir ileti, PMSGDLV ya da NPMSGDLV (iletinin kalıcı olmasına bağlı olarak) ALL değerine ayarlanmış olarak bir TOPIC içinde yayınlanır ve herhangi bir aboneliğin (sürekli ya da değil) yayını alamayacağı bir kuyruğu vardır.
 - Bir ileti, ALLDUR olarak ayarlanmış PMSGDLV ya da NPMSGDLV (iletinin kalıcı olmasına bağlı olarak) ile bir TOPIC ' e yayınlanır ve sürekli abonelik, yayını alamayan bir kuyruğa sahiptir.Yayınlar aşağıdaki durumlarda bazı abonelere teslim edilemese de MQPUT RCNONE ile dönebilir:
 - Bir ileti, PMSGDLV ya da NPMSGDLV (iletinin kalıcı olmasına bağlı olarak) AllaVAIL olarak ayarlanmış bir TOPIC ' e yayınlanır ve kalıcı ya da kalıcı olmayan herhangi bir abonelik, yayını alamayan bir kuyruğa sahiptir.
 - Bir ileti, ALLDUR olarak ayarlanmış PMSGDLV ya da NPMSGDLV ile bir TOPIC içinde yayınlanır ve kalıcı olmayan bir abonelik, yayını alamayan bir kuyruğa sahiptir.
2. Kullanılmakta olan konuya abone yoksa, yayınlanan ileti herhangi bir kuyruğa gönderilmez ve atılır. Bu iletinin kalıcı ya da kalıcı olmaması ya da sınırsız süre bitimi ya da küçük bir süre sonu olması fark etmez; abone yoksa, yine de atılır. Bunun istisnası, ileti alıkonacaksa, bu durumda ileti herhangi bir abonenin kuyruğuna gönderilmese de, yeni aboneliklere teslim edilecek konuya ya da MQSUBRQ kullanılarak alıkonan yayınları soran abonelere karşı depolanır.

MQPUT ve MQPUT1

İletileri kuyruğa koymak için hem MQPUT hem de MQPUT1 çağrıları kullanılabilir; kullanılacak çağrı duruma bağlıdır

- Birden çok ileti aynı kuyruğuna yerleştirilirken MQPUT çağrısı kullanılmalıdır.

Önce OOOOUT seçeneğini belirten bir MQOPEN çağrısı yayınlanır, ardından kuyruğa ileti eklemek için bir ya da daha çok MQPUT isteği gelir; son olarak kuyruk bir MQCLOSE çağrısıyla kapatılır. Bu, MQPUT1 çağrısının yinelenen kullanımından daha iyi performans sağlar.

- MQPUT1 çağrısı, bir kuyruğa yalnızca bir iletisi yerleştirildiğinde kullanılmalıdır.

Bu çağrı, MQOPEN, MQPUT ve MQCLOSE çağrılarını tek bir çağrıya sararak, verilmesi gereken çağrı sayısını en aza indirir.

Hedef kuyruklar

Bir uygulama ileti gruplarını kullanmadan aynı kuyruğa ileti dizisi koyarsa, Aşağıdaki koşullar yerine getirilirse, bu iletilerin sırası korunur. Bazı koşullar hem yerel, hem de uzak hedef kuyruklar için geçerlidir; diğer koşullar yalnızca uzak hedef kuyruklar için geçerlidir.

Yerel ve uzak hedef kuyruklarına ilişkin koşullar

- Tüm MQPUT çağrılarını aynı iş birimi içinde ya da hiçbiri bir iş birimi içinde değil.
İletiler tek bir iş birimi içindeki belirli bir kuyruğa yerleştirildiğinde, diğer uygulamalardan gelen iletiler kuyruktaki ileti sırasıyla birlikte serpiştirilir.
- Tüm MQPUT çağrılarını aynı HOBJ nesne tanıtıcısı kullanılarak yapılır.
Bazı ortamlarda, çağrılar aynı uygulamadan yapıldığında, farklı nesne tanıtıcıları kullanıldığında da ileti sırası korunur. "Aynı uygulamanın" anlamı çevre tarafından belirlenir:
 - IBM i' de, iş uygulama.
- Tüm iletilerin önceliği aynı.

Uzak hedef kuyruklarına ilişkin ek koşullar

- Gönderen kuyruk yöneticisinden hedef kuyruk yöneticisine tek bir yol vardır.
Dizideki bazı iletilerin farklı bir yola (örneğin, yeniden yapılandırma, trafik dengeleme ya da ileti büyüklüğüne dayalı yol seçimi nedeniyle) gitme olasılığı varsa, hedef kuyruk yöneticisindeki iletilerin sırası garanti edilemez.
- İletiler, gönderen, ara ya da hedef kuyruk yöneticilerindeki gitmeyen iletiler kuyruklarına geçici olarak yerleştirilmez.
İletilerden biri ya da birkaçı geçici olarak gitmeyen iletiler kuyruğuna konursa (örneğin, bir iletim kuyruğu ya da hedef kuyruk geçici olarak dolu olduğu için), iletiler hedef kuyruğa sıra dışı olarak gelebilir.
- İletilerin tümü kalıcı ya da kalıcı değil.

Gönderen ve hedef kuyruk yöneticileri arasındaki rotadaki bir kanalda **CDNPM** özneliği NPFAS olarak ayarlanmışsa, kalıcı olmayan iletiler kalıcı iletilerin önüne geçebilir ve kalıcı olmayan iletilerin korunmamasına göre kalıcı iletilerin sırası ortaya çıkmaktadır. Ancak, birbiriyle görece kalıcı iletilerin ve birbirleriyle görece kalıcı olmayan iletilerin sırası korunur.

Bu koşullar karşılanmazsa, ileti grupları ileti sırasını korumak için kullanılabilir, ancak bunun hem gönderen hem de alan uygulamaların ileti gruplaması desteğini kullanmasını gerektirdiğini unutmayın. İleti gruplarıyla ilgili ek bilgi için aşağıdaki konuya bakın:

- MQMD ' deki *MDMFL* alanı
- MQPMO içindeki *PMLOGO* seçeneği
- MQGMO ' da *GMLOGO* seçeneği

Dağıtım listeleri

Aşağıdaki notlar dağıtım listelerinin kullanımı için geçerlidir.

1. İletiler, version-1 ya da version-2 MQPMO kullanılarak bir dağıtım listesine konabilir. version-1 MQPMO kullanılıyorsa (ya da *PMREC* sıfıra eşit bir version-2 MQPMO kullanılıyorsa), uygulama tarafından bir koyma ileti kaydı ya da yanıt kaydı sağlanamaz. Başka bir deyişle, ileti dağıtım listesindeki bazı kuyruklara değil, bazı kuyruklara başarıyla gönderilirse, hatalarla karşılaşan kuyruklar belirlenemez.

Uygulama tarafından konan ileti kayıtları ya da yanıt kayıtları sağlanırsa, *PMVER* alanı *PMVER2* olarak ayarlanmalıdır.

version-2 MQPMO, *PMREC* ' in sıfır olduğundan emin olarak dağıtım listesinde olmayan tek bir kuyruğa ileti göndermek için de kullanılabilir.

2. Tamamlanma kodu ve neden kodu değiştiricileri aşağıdaki gibi ayarlanır:

- Dağıtım listesindeki kuyruklara yapılan girişler aynı şekilde başarılı ya da başarısız olursa, tamamlanma kodu ve neden kodu parametreleri ortak sonucu tanımlayacak şekilde ayarlanır. Bu durumda MQRR yanıt kayıtları (uygulama tarafından sağlandıysa) ayarlanmaz.

Örneğin, her koyma başarılı olursa, tamamlama kodu CCOK olarak ayarlanır ve neden kodu RCNONE olur; girişler için tüm kuyruklar engellendiği için her koyma başarısız olursa, parametreler CCFAIL ve RC2051 olarak ayarlanır.

- Dağıtım listesindeki kuyruklara konan konutların tümü aynı şekilde başarılı ya da başarısız olmazsa:
 - En az bir koyma başarılı olursa, tamamlama kodu parametresi CCWARN olarak ve tümü başarısız olursa CCFAIL olarak ayarlanır.
 - Neden kodu parametresi RC2136 olarak ayarlandı.
 - Yanıt kayıtları (uygulama tarafından sağlandıysa), dağıtım listesindeki kuyruklara ilişkin tek tek tamamlama kodlarına ve neden kodlarına ayarlanır.

Hedef için açma başarısız olduğu için hedefe koyma başarısız olursa, yanıt kaydındaki alanlar CCFAIL ve RC2137; olarak ayarlanır ve bu hedef *PMIDC* içinde yer alır.

3. Dağıtım listesindeki bir hedef yerel kuyruğa çözüldüğünde, ileti o kuyruğa olağan biçimde yerleştirilir (dağıtım listesi iletisi olarak değil). Birden çok hedef aynı yerel kuyruğa çözümlenirse, bu tür her hedef için kuyruğa bir ileti yerleştirilir.

Dağıtım listesindeki bir hedef uzak kuyruğa çözüldüğünde, ileti uygun iletim kuyruğuna yerleştirilir. Birden çok hedefin aynı iletim kuyruğuna çözülmesi durumunda, bu hedefleri içeren tek bir dağıtım listesi iletisi, uygulama tarafından sağlanan hedefler listesinde bitişik olmasa da iletim kuyruğuna yerleştirilebilir. Ancak, bu yalnızca iletim kuyruğu dağıtım listesi iletilerini destekliyorsa yapılabilir (bkz. [“Kuyruklara ilişkin öznitelikler” sayfa 1345](#) içinde açıklanan **DistLists** kuyruk özniteliğine bakın).

İletim kuyruğu dağıtım listelerini desteklemiyorsa, iletinin olağan biçimde bir kopyası, bu iletim kuyruğunu kullanan her hedef için iletim kuyruğuna yerleştirilir.

Uygulama ileti verilerini içeren bir dağıtım listesi iletim kuyruğu için çok büyükse, dağıtım listesi iletisi, her biri daha az hedef içeren daha küçük dağıtım listesi iletilerine bölünür. Uygulama ileti verileri yalnızca kuyruğa sığarsa, dağıtım listesi iletileri hiç kullanılmaz ve kuyruk yöneticisi, iletim kuyruğunu kullanan her hedef için normal biçimde iletinin bir kopyasını oluşturur.

Farklı hedeflerin ileti önceliği ya da ileti kalıcılığı farklıysa (bu, uygulama PRQDEF ya da PEQDEF değerini belirlediğinde ortaya çıkabilir), iletiler aynı dağıtım listesi iletisinde tutulmaz. Kuyruk yöneticisi, farklı öncelik ve kalıcılık değerlerini barındırmak için gereken sayıda dağıtım listesi iletisi oluşturur.

4. Dağıtım listesine konan bir not aşağıdakilere neden olabilir:

- Tek bir dağıtım listesi iletisi ya da
- Daha küçük dağıtım listesi iletileri ya da
- Dağıtım listesi iletilerinin ve normal iletilerin karışımı ya da
- Yalnızca normal iletiler.

Önceki oluşlardan hangisinin gerçekleştiği aşağıdakilere bağlıdır:

- Listedeki hedefler yerel, uzak ya da karışık.
- Hedefler aynı ileti önceliğine ve ileti kalıcılığına sahip.
- İletim kuyrukları dağıtım listesi iletilerini tutabilir.
- İletim kuyruklarının ileti uzunluğu üst sınırı, iletiyi dağıtım listesi biçiminde barındıracak kadar büyüktür.

Ancak, yukarıdakilerden hangisinin oluştuğunu dikkate almaksızın, sonuçta ortaya çıkan her *fiziksel* ileti (yani, her normal ileti ya da dağıtım listesi iletisi), aşağıdaki durumlarda yalnızca *tek* ileti olarak sayılır:

- Uygulamanın bir iş biriminde izin verilen ileti sayısı üst sınırını aşmış denetleniyor (**MaxUncommittedMsgs** kuyruk yöneticisi özniteliğine bakın).
- Tetikleme koşullarının karşılanıp karşılanmadığı denetleniyor.
- Kuyruk derinlikleri artırılıyor ve kuyrukların kuyruk derinliği üst sınırının aşılıp aşılmayacağı denetleniyor.

5. Kuyruk tanımlamalarında, kuyruklar ayrı ayrı açıldığında (örneğin, çözüm yolundaki bir değişiklik), tanıtıcı geçersiz duruma gelmesine neden olacak herhangi bir değişiklik, dağıtım listesi tanıtıcılarının geçersiz olmasına neden olmaz. Ancak, dağıtım listesi tanıtıcısı sonraki bir MQPUT çağrısında kullanıldığında, o kuyruk için bir hata oluşmasına neden olur.

Üst Bilgiler

Uygulama ileti verilerinin başına bir ya da daha çok IBM MQ üstbilgi yapısıyla bir ileti konursa, kuyruk yöneticisi bunların geçerli olduğunu doğrulamak için üstbilgi yapılarında belirli denetimler gerçekleştirir. Kuyruk yöneticisi bir hata saptarsa, çağrı uygun bir neden koduyla başarısız olur. Gerçekleştirilen denetimler, var olan belirli yapılara göre değişir. Ayrıca, denetimler yalnızca MQPUT ya da MQPUT1 çağrısında bir version-2 ya da üstü MQMD kullanılıyorsa gerçekleştirilir; uygulama ileti verilerinin başlangıcında bir MQMDE varsa bile, bu denetimler gerçekleştirilmez.

Şu IBM MQ üstbilgi yapılarının geçerliliği kuyruk yöneticisi tarafından tam olarak denetleniyor: MQDH, MQMDE.

Diğer IBM MQ üstbilgi yapıları için, kuyruk yöneticisi bazı geçerlilik denetimi gerçekleştirir, ancak her alanı denetlemez. Yerel kuyruk yöneticisi tarafından desteklenmeyen yapıların ve iletideki ilk MQDLH 'yi izleyen yapıların geçerliliği denetlenmez.

IBM MQ yapılarındaki alanlarda genel denetimlere ek olarak aşağıdaki koşullar da karşılanmalıdır:

- Bir IBM MQ yapısı iki ya da daha fazla segmente bölünmemelidir; yapının tamamı tek bir kesim içinde bulunmalıdır.
- Bir PCF iletisindeki yapıların uzunluklarının toplamı, MQPUT ya da MQPUT1 çağrısında **BUFLN** değiştirilmesiyle belirtilen uzunluğa eşit olmalıdır. PCF iletisi, aşağıdaki biçim adlarından birine sahip bir iletidir:
 - FMADMN
 - FMEVNT
 - FMPCF
- IBM MQ yapıları, kesilmiş yapılara izin verilen aşağıdaki durumlar dışında kesilmemelidir:
 - Rapor iletileri olan iletiler.
 - PCF iletileri.
 - MQDLH yapısı içeren iletiler. (İlk MQDLH 'yi izleyen yapılar kesilebilir; MQDLH' den önceki yapılar kesilemiyor.)

Arabellek

RPG programlama örneğinde gösterilen **BUFFER** parametresi bir dizgi olarak bildirildi; bu, parametrenin uzunluk üst sınırını 256 baytla sınırlar. Daha büyük bir arabellek gerekiyorsa, parametre bir yapı olarak ya da fiziksel bir dosyada bir alan olarak bildirilmelidir. Bu, mümkün olan uzunluk üst sınırını yaklaşık 32 KB 'ye yükseltir.

Parametreler

MQPUT çağrısındaki değiştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. HCONN değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

HOBJ (10 basamaklı işaretli tamsayı)-giriş

Nesne tanıtıcısı.

Bu tanıtıcı, iletinin eklendiği kuyruğu ya da iletinin yayınlandığı konuyu gösterir. *HOB*J değeri, *OOOUT* seçeneğini belirten önceki bir *MQOPEN* çağrısı tarafından döndürüldü.

MSGDSC (MQMD)-giriş/çıkış

İleti tanımlayıcı.

Bu yapı, gönderilmekte olan iletinin özniteliklerini tanımlar ve koyma isteği tamamlandıktan sonra iletiyle ilgili bilgileri alır. Ayrıntılar için bkz. "IBM i üzerinde MQMD (İleti tanımlayıcı)" sayfa 1093.

Uygulama bir version-1 MQMD 'si sağlarsa, version-2 MQMD 'de bulunan, ancak version-1' de bulunmayan alanlara ilişkin değerleri belirtmek için ileti verilerinin başına bir MQMDE yapısı konabilir. MQMD 'deki *MDFMT* alanı, bir MQMDE' nin var olduğunu göstermek için FMMDE olarak ayarlanmalıdır. Daha fazla ayrıntı için bkz. "IBM i üzerinde MQMDE (İleti tanımlayıcı uzantısı)" sayfa 1134 .

PMO (MQPMO)-giriş/çıkış

MQPUT işlemini denetleyen seçenekler.

Ayrıntılar için bkz. "IBM i üzerinde MQPMO (Put-message options)" sayfa 1154.

BUFLEN (10 basamaklı işaretli tamsayı)-giriş

*BUFFER*içindeki iletinin uzunluğu.

Sıfır geçerlidir ve iletinin uygulama verisi içermediğini gösterir. *BUFLEN* için üst sınır çeşitli etmenlere bağlıdır:

- Hedef kuyruk paylaşılan bir kuyruksa, üst sınır 63 KB (64 512 bayt) olur.
- Hedef yerel bir kuyruksa ya da yerel bir kuyruğa çözümlendiye (ancak paylaşılan bir kuyruk değilse), üst sınır aşağıdakilerden biri olup olmadığına bağlıdır:
 - Yerel kuyruk yöneticisi bölümlenmeyi destekler.
 - Gönderen uygulama, kuyruk yöneticisinin iletiyi kesmesini sağlayan işareti belirtir. Bu işaret *MFSEGA* 'dır ve version-2 MQMD 'de ya da version-1 MQMD ile kullanılan bir MQMDE' de belirtilebilir.

Bu koşulların her ikisi de karşılandıysa, *BUFLEN* , MQMD 'deki *MDOFF* alanının değerinden 999 999 999 999 değerini aşamaz. Bu nedenle, konabilecek en uzun mantıksal ileti 999 999 999 999 bayttır (*MDOFF* sıfır olduğunda). Ancak, uygulamanın çalıştığı işletim sistemi ya da ortam tarafından uygulanan kaynak kısıtlamaları daha düşük bir sınırla sonuçlanabilir.

Daha önce açıklanan koşullardan biri ya da her ikisi karşılanmazsa, *BUFLEN* , kuyruğun **MaxMsgLength** özneliği ve kuyruk yöneticisinin **MaxMsgLength** özneliğinden daha küçük olanı aşamaz.

- Hedef uzak bir kuyruksa ya da uzak bir kuyruğa çözüldüyse, yerel kuyruklara ilişkin koşullar geçerli olur , *ancak iletinin hedef kuyruğa ulaşmak için geçmesi gereken her kuyruk yöneticisinde* ; Özellikle:
 1. İletiyi geçici olarak yerel kuyruk yöneticisinde saklamak için kullanılan yerel iletim kuyruğu
 2. İletiyi yerel ve hedef kuyruk yöneticileri arasındaki rotada kuyruk yöneticilerine saklamak için kullanılan ara iletim kuyrukları (varsa)
 3. Hedef kuyruk yöneticisindeki hedef kuyruk

Bu nedenle, konabilecek en uzun ileti, bu kuyrukların ve kuyruk yöneticilerinin en kısıtlayıcı olanına göre yönetilir.

Bir ileti iletim kuyruğundayken, ek bilgiler ileti verileriyle birlikte bulunur ve bu, taşınabilecek uygulama verileri miktarını azaltır. Bu durumda, *BUFLEN*'sınıırı belirlenirken iletim kuyruklarının *MaxMsgLength* değerlerinden LNMHD baytların çıkarılması önerilir.

Not: İleti konduğunda, yalnızca koşul 1 'e uyulmaması (RC2030 ya da RC2031neden koduyla) zamanuyumlu olarak tanılanabilir. 2 ya da 3 numaralı koşullar karşılanmazsa, ileti, bir ara kuyruk yöneticisinde ya da hedef kuyruk yöneticisinde bir teslim edilmeyen ileti kuyruğuna yeniden yönlendirilir. Bu durumda, gönderen tarafından istendiyse bir rapor iletisi oluşturulur.

BUFFER (1 baytlık bit dizgisi x BUFLen)-giriş

İleti verileri.

Bu, gönderilecek uygulama verilerini içeren bir arabellektir. Arabellek, iletideki verilerin niteliğine uygun bir sınırla hizalanmalıdır. 4 baytlık hizalama çoğu ileti için uygun olmalıdır (MQ üstbilgi yapılarını içeren iletiler de içinde olmak üzere), ancak bazı iletiler daha sıkı hizalama gerektirebilir. Örneğin, 64 bitlik ikili tamsayı içeren bir ileti 8 baytlık hizalama gerektirebilir.

BUFFER, karakter verileri, sayısal veriler ya da her ikisini içeriyorsa, **MSGDSC** parametresindeki *MDCSI* ve *MDENC* alanları, veriler için uygun değerlere ayarlanmalıdır; bu, iletinin alıcısının verileri (gerekliyse) alıcı tarafından kullanılan karakter kümesine ve kodlamaya dönüştürmesini sağlar.

Not: MQPUT çağrısındaki diğer tüm değiştirgeler, **CodedCharSetId** kuyruk yöneticisi özniteliği tarafından belirtilen karakter kümesinde ve ENNAT tarafından verilen yerel kuyruk yöneticisinin kodlamasında olmalıdır.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2104

(2104, X'838 ') İleti tanımlayıcıdaki Rapor seçeneği tanınmıyor.

RC2136

(2136, X'858 ') Birden çok neden kodu döndürüldü.

CMPCOD CCFail ise:

RC2004

(2004, X'7D4') Arabellek değıştirgesi geçersiz.

RC2005

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2013

(2013, X'7DD') Sona erme süresi geçerli değil.

RC2014

(2014, X'7DE') Geribildirim kodu geçersiz.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2019

(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

RC2024

(2024, X'7E8') Yürürlükteki iş birimi içinde başka ileti işlenmez.

RC2026

(2026, X'7EA') İleti tanımlayıcısı geçersiz.

RC2027

(2027, X'7EB') Yanıt kuyruğu eksik.

RC2029

(2029, X'7ED') İleti tanımlayıcıdaki ileti tipi geçersiz.

RC2030

(2030, X'7EE') İleti uzunluğu kuyruk için üst sınırdan fazla.

RC2031

(2031, X'7EF') Kuyruk yöneticisi için ileti uzunluğu üst sınırından fazla.

RC2039

(2039, X'7F7') Kuyruk çıkış için açık değil.

RC2041

(2041, X'7F9') Nesne tanımı açıldıktan sonra değiştirildi.

RC2046

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2047

(2047, X'7FF') Kalıcılık geçersiz.

RC2048

(2048, X'800 ') Kuyruk kalıcı iletileri desteklemiyor.

RC2050

(2050, X'802 ') İleti önceliği geçersiz.

RC2051

(2051, X'803 ') Kuyruk için çağrılarını engelleme.

RC2052

(2052, X'804 ') Kuyruk silindi.

RC2053

(2053, X'805 ') Kuyruk zaten ileti sayısı üst sınırını içeriyor.

RC2056

(2056, X'808 ') Diskte kuyruk için kullanılabilir alan yok.

RC2058

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

RC2059

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.

RC2061

(2061, X'80D') İleti tanımlayıcıdaki rapor seçenekleri geçersiz.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2072

(2072, X'818 ') Syncpoint desteği kullanılmıyor.

RC2093

(2093, X'82D') Kuyruk tüm bağlamı geçirmek için açık değil.

RC2094

(2094, X'82E') Kuyruk kimlik bağlamı için açık değil.

RC2095

(2095, X'82F') Kuyruk tüm bağlamı ayarlamak için açık değil.

RC2096

(2096, X'830 ') Kimlik bağlamını ayarlamak için kuyruk açık değil.

RC2097

(2097, X'831 ') Bu gönderide bulunulan kuyruk tanıtıcısı bağlamı saklamıyor.

RC2098

(2098, X'832 ') Bu bağlam, gönderme yapılan kuyruk tanıtıcısı için kullanılamaz.

RC2101

(2101, X'835 ') Nesne zarar gördü.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2135

(2135, X'857 ') Dağıtım üstbilgisi yapısı geçersiz.

RC2136

(2136, X'858 ') Birden çok neden kodu döndürüldü.

RC2137

(2137, X'859 ') Nesne başarıyla açılmadı.

RC2149

(2149, X'865 ') PCF yapıları geçersiz.

RC2154

(2154, X'86A') Mevcut kayıt sayısı geçerli değil.

RC2156

(2156, X'86C') Yanıt kayıtları geçerli değil.

RC2158

(2158, X'86E') Put ileti kaydı işaretleri geçerli değil.

RC2159

(2159, X'86F') Put ileti kayıtları geçerli değil.

RC2161

(2161, X'871 ') Kuyruk yöneticisi susturuyor.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2173

(2173, X'87D') Put-message seçenek yapısı geçerli değil.

RC2185

(2185, X'889 ') Tutarsız kalıcılık belirtimi.

RC2188

(2188, X'88C') Çağrı küme iş yükü çıkışı tarafından reddedildi.

RC2189

(2189, X'88D') Küme adı çözümlemesi başarısız oldu.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2241

(2241, X'8C1') İleti grubu tamamlanmadı.

RC2242

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

RC2245

(2245, X'8C5') İş birimi belirtimi tutarsız.

RC2248

(2248, X'8C8') İleti tanımlayıcı uzantısı geçersiz.

RC2249

(2249, X'8C9') İleti işaretleri geçerli değil.

RC2250

(2250, X'8CA') İleti sıra numarası geçersiz.

RC2251

(2251, X'8CB') İleti kesimi görelî konumu geçersiz.

RC2252

(2252, X'8CC') Özgün uzunluk geçersiz.

RC2253

(2253, X'8CD') İleti bölümündeki verilerin uzunluğu sıfır.

RC2255

(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.

RC2257

(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

RC2258

(2258, X'8D2') Grup tanıtıcısı geçersiz.

RC2266

(2266, X'8DA') Küme iş yükü çıkışı başarısız oldu.

RC2269

(2269, X'8DD') Küme kaynağı hatası.

RC2270

(2270, X'8DE') Kullanılabilir hedef kuyruk yok.

RC2420

(2420) Bir MQPUT çağrısı yayınlandı, ancak ileti verileri geçersiz bir MQEPH yapısı içeriyor.

RC2479

(2479, X'9AF') Yayın alıkonamadı.

RC2480

(2480, X'9B0') Hedef tipi değişti: Diğer ad kuyruğu bir kuyruğa gönderme yaptı, ancak şimdi bir konuya gönderme yapıyor.

RC2502

(2502, X'9C6') Yayın başarısız oldu ve yayın hiçbir aboneye teslim edilmedi

RC2551

(2551, X'9F7') Belirtilen seçim dizgisi kullanılmıyor.

RC2554

(2554, X'9FA') İletinin genişletilmiş ileti seçici ile bir aboneye teslim edilip edilmeyeceğini belirlemek için ileti içeriği ayrıştırılmadı.

RPG Bildirimi

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQPUT(HCONN : HOBJ : MSGDSC : PMO :
C          BUFLLEN : BUFFER : CMPCOD :
C          REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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 (Bir ileti koy) IBM i

MQPUT1 çağrısı, bir iletiyi bir kuyruğa ya da dağıtım listesine ya da bir konuya koyar. Kuyruk, dağıtım listesi ya da konunun açık olması gerekmez.

- [“Sözdizimi” sayfa 1320](#)
- [“Kullanım notları” sayfa 1320](#)
- [“Parametreler” sayfa 1321](#)
- [“RPG Bildirimi” sayfa 1325](#)

Sözdizimi

MQPUT1 (*HCONN, OBJDSC, MSGDSC, PMO, BUFLen, BUFFER, CMPCOD, REASON*)

Kullanım notları

1. Bir kuyruğa ileti koymak için hem MQPUT hem de MQPUT1 çağrıları kullanılabilir; kullanılacak çağrı aşağıdaki koşullara bağlıdır:
 - Birden çok ileti *aynı* kuyruğuna yerleştirilirken MQPUT çağrısı kullanılmalıdır. Önce OOOOUT seçeneğini belirten bir MQOPEN çağrısı yayınlanır, ardından kuyruğa ileti eklemek için bir ya da daha çok MQPUT isteği gelir; son olarak kuyruk bir MQCLOSE çağrısıyla kapatılır. Bu, MQPUT1 çağrısının yinelenen kullanımından daha iyi performans sağlar.
 - MQPUT1 çağrısı, bir kuyruğa yalnızca *bir* iletisi yerleştirildiğinde kullanılmalıdır. Bu çağrı, MQOPEN, MQPUT ve MQCLOSE çağrılarını tek bir çağrıya sararak, verilmesi gereken çağrı sayısını en aza indirir.
2. Bir uygulama ileti gruplarını kullanmadan aynı kuyruğa ileti dizisi koyarsa, belirli koşullar yerine getirilirse, bu iletilerin sırası korunur. Ancak, çoğu ortamda MQPUT1 çağrısı bu koşulları karşılamaz ve bu nedenle ileti sırasını korumaz. MQPUT çağrısı bu ortamlarda kullanılmalıdır. Ayrıntılar için MQPUT çağrısının tanımındaki kullanım notlarına bakın.
3. MQPUT1 çağrısı, iletileri dağıtım listelerine koymak için kullanılabilir. Bununla ilgili genel bilgi için MQOPEN ve MQPUT çağrılarına ilişkin kullanım notlarına bakın. MQPUT1 çağrısı kullanılırken aşağıdaki farklılıklar söz konusudur:
 - a. MQRR yanıt kayıtları uygulama tarafından sağlandıysa, bunların MQOD yapısı kullanılarak sağlanması gerekir; bunlar MQPMO yapısı kullanılarak sağlanamaz.
 - b. Neden kodu RC2137 yanıt kayıtlarında MQPUT1 tarafından hiçbir zaman döndürülmez; bir kuyruk açılmazsa, o kuyruğa ilişkin yanıt kaydı açma işleminden kaynaklanan gerçek neden kodunu içerir. Bir kuyruğa ilişkin açma işlemi CCWARN tamamlanma koduyla başarılı olursa, o kuyruğa ilişkin yanıt kaydındaki tamamlanma kodu ve neden kodu, koyma işleminden kaynaklanan tamamlanma ve neden kodlarıyla değiştirilir. MQOPEN ve MQPUT çağrılarında olduğu gibi, kuyruk yöneticisi yanıt kayıtlarını (varsa) yalnızca, çağrı sonucu dağıtım listesindeki tüm kuyruklar için aynı olmadığında ayarlar; bu, çağrı tamamlanırken RC2136neden koduyla gösterilir.
4. MQPUT1 çağrısı bir iletiyi küme kuyruğuna koymak için kullanılırsa, çağrı MQOPEN çağrısında OOBNDN belirtilmiş gibi davranır.

5. Uygulama ileti verilerinin başına bir ya da daha çok IBM MQ üstbilgi yapısıyla bir ileti konursa, kuyruk yöneticisi bunların geçerli olduğunu doğrulamak için üstbilgi yapılarında belirli denetimler gerçekleştirir. Bununla ilgili daha fazla bilgi için MQPUT çağrısına ilişkin kullanım notlarına bakın.
6. Birden çok uyarı durumu ortaya çıkarsa (bkz. **CMPCOD** parametresi), döndürülen neden kodu, geçerli olan aşağıdaki listede *ilk* olur:
 - a. RC2136
 - b. RC2242
 - c. RC2241
 - d. RC2049 ya da RC2104
7. RPG programlama örneğinde gösterilen **BUFFER** parametresi bir dizgi olarak bildirildi; bu, parametrenin uzunluk üst sınırını 256 baytla sınırlar. Daha büyük bir arabellek gerekiyorsa, parametre bir yapı olarak ya da fiziksel bir dosyada bir alan olarak bildirilmelidir. Bu, mümkün olan uzunluk üst sınırını yaklaşık 32 KB 'ye yükseltir.

Parametreler

MQPUT1 çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı iřaretli tamsayı)-giriř

Baęlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik baęlantıyı gösterir. *HCONN* deęeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

OBJDSC (MQOD)-giriř/çıkıř

Nesne tanımlayıcı.

Bu, iletinin eklendięi kuyruęu tanımlayan bir yapıdır. Ayrıntılar için bkz. "IBM i üzerinde MQOD (Nesne tanımlayıcı)" sayfa 1141.

Kullanıcının çıkıř kuyruęunu açma yetkisi olmalıdır. Kuyruk bir model kuyruęu **olmamalıdır** .

MSGDSC (MQMD)-giriř/çıkıř

İleti tanımlayıcı.

Bu yapı, gönderilmekte olan iletinin özniteliklerini tanımlar ve koyma isteęi tamamlandıktan sonra geribildirim bilgilerini alır. Ayrıntılar için bkz. "IBM i üzerinde MQMD (İleti tanımlayıcı)" sayfa 1093.

Uygulama bir version-1 MQMD 'si saęlarsa, version-2 MQMD 'de bulunan, ancak version-1' de bulunmayan alanlara iliřkin deęerleri belirtmek için ileti verilerinin başına bir MQMDE yapısı konabilir. MQMD 'deki *MDFMT* alanı, bir MQMDE' nin var olduęunu göstermek için FMMDE olarak ayarlanmalıdır. Daha fazla ayrıntı için bkz. "IBM i üzerinde MQMDE (İleti tanımlayıcı uzantısı)" sayfa 1134 .

PMO (MQPMO)-giriř/çıkıř

MQPUT1iřlemine denetleyen seęenekler.

Ayrıntılar için bkz. "IBM i üzerinde MQPMO (Put-message options)" sayfa 1154.

BUFLEN (10 basamaklı iřaretli tamsayı)-giriř

*BUFFER*içindeki iletinin uzunluęu.

Sıfır geçerlidir ve iletinin uygulama verisi içermedięini gösterir. Üst sınır çeřitli etmenlere baęlıdır; ek ayrıntılar için MQPUT çağrısının **BUFLEN** deęiřtirgesinin tanımına bakın.

BUFFER (1 baytlık bit dizgisi x BUFLEN)-giriř

İleti verileri.

Bu, gönderilecek uygulama iletisi verilerini içeren bir arabellektir. Arabellek, iletideki verilerin nitelięine uygun bir sınırla hizalanmalıdır. 4 baytlık hizalama, çoęu ileti için uygun olmalıdır (IBM

MQ üstbilgi yapılarını içeren iletiler de içinde olmak üzere), ancak bazı iletiler daha sıkı hizalama gerektirebilir. Örneğin, 64 bitlik ikili tamsayı içeren bir ileti 8 baytlık hizalama gerektirebilir.

BUFFER, karakter verileri, sayısal veriler ya da her ikisini içeriyorsa, **MSGDSC** parametresindeki *MDCSI* ve *MDENC* alanları, veriler için uygun değerlere ayarlanmalıdır; bu, iletinin alıcısının verileri (gerekliyse) alıcı tarafından kullanılan karakter kümesine ve kodlamaya dönüştürmesini sağlar.

Not: MQPUT1 çağrısındaki diğer tüm değiştiriciler, **CodedCharSetId** kuyruk yöneticisi özneliği ve ENNAT tarafından belirtilen yerel kuyruk yöneticisinin kodlaması tarafından belirtilen karakter kümesinde olmalıdır.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2104

(2104, X'838 ') İleti tanımlayıcıdaki Rapor seçeneği tanınmıyor.

RC2136

(2136, X'858 ') Birden çok neden kodu döndürüldü.

RC2049

(2049, X'801 ') İleti Önceliği, desteklenen değer üst sınırını aşıyor.

RC2241

(2241, X'8C1') İleti grubu tamamlanmadı.

RC2242

(2242, X'8C2') Mantıksal ileti tamamlanmadı.

CMPCOD CCFAIL ise:

RC2001

(2001, X'7D1') Diğer ad temel kuyruğu geçerli bir tip değil.

RC2004

(2004, X'7D4') Arabellek değiştirgesi geçersiz.

RC2005

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2013

(2013, X'7DD') Sona erme süresi geçerli değil.

RC2014

(2014, X'7DE') Geribildirim kodu geçersiz.

- RC2017**
(2017, X'7E1') Başka tanıtıcı yok.
- RC2018**
(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.
- RC2024**
(2024, X'7E8') Yürürlükteki iş birimi içinde başka ileti işlenmez.
- RC2026**
(2026, X'7EA') İleti tanımlayıcısı geçersiz.
- RC2027**
(2027, X'7EB') Yanıt kuyruğu eksik.
- RC2029**
(2029, X'7ED') İleti tanımlayıcıdaki ileti tipi geçersiz.
- RC2030**
(2030, X'7EE') İleti uzunluğu kuyruk için üst sınırdan fazla.
- RC2031**
(2031, X'7EF') Kuyruk yöneticisi için ileti uzunluğu üst sınırından fazla.
- RC2035**
(2035, X'7F3') Erişim için yetkili değil.
- RC2042**
(2042, X'7FA') Nesne zaten çakışan seçeneklerle açık.
- RC2043**
(2043, X'7FB') Nesne tipi geçersiz.
- RC2044**
(2044, X'7FC') Nesne tanımlayıcı yapısı geçerli değil.
- RC2046**
(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.
- RC2047**
(2047, X'7FF') Kalıcılık geçersiz.
- RC2048**
(2048, X'800 ') Kuyruk kalıcı iletileri desteklemiyor.
- RC2050**
(2050, X'802 ') İleti önceliği geçersiz.
- RC2051**
(2051, X'803 ') Kuyruk için çağrılarını engelleme.
- RC2052**
(2052, X'804 ') Kuyruk silindi.
- RC2053**
(2053, X'805 ') Kuyruk zaten ileti sayısı üst sınırını içeriyor.
- RC2056**
(2056, X'808 ') Diskte kuyruk için kullanılabilir alan yok.
- RC2057**
(2057, X'809 ') Kuyruk tipi geçersiz.
- RC2058**
(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.
- RC2059**
(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılmıyor.
- RC2061**
(2061, X'80D') İleti tanımlayıcıdaki rapor seçenekleri geçersiz.
- RC2063**
(2063, X'80F') Güvenlik hatası oluştu.

- RC2071**
(2071, X'817 ') Kullanılabilir bellek yetersiz.
- RC2072**
(2072, X'818 ') Syncpoint desteęi kullanılmıyor.
- RC2082**
(2082, X'822 ') Bilinmeyen dięer ad temel kuyruęu.
- RC2085**
(2085, X'825 ') Bilinmeyen nesne adı.
- RC2086**
(2086, X'826 ') Bilinmeyen nesne kuyruk yöneticisi.
- RC2087**
(2087, X'827 ') Bilinmeyen uzak kuyruk yöneticisi.
- RC2091**
(2091, X'82B') İletim kuyruęu yerel deęil.
- RC2092**
(2092, X'82C') İletim kuyruęu yanlış kullanımla.
- RC2097**
(2097, X'831 ') Bu gönderide bulunan kuyruk tanıtıcısı bağlamı saklamıyor.
- RC2098**
(2098, X'832 ') Bu bağlam, gönderme yapılan kuyruk tanıtıcısı için kullanılamaz.
- RC2101**
(2101, X'835 ') Nesne zarar gördü.
- RC2102**
(2102, X'836 ') Sistem kaynakları yetersiz.
- RC2135**
(2135, X'857 ') Daęıtım üstbilgisi yapısı geçersiz.
- RC2136**
(2136, X'858 ') Birden çok neden kodu döndürüldü.
- RC2149**
(2149, X'865 ') PCF yapıları geçersiz.
- RC2154**
(2154, X'86A') Mevcut kayıt sayısı geçerli deęil.
- RC2155**
(2155, X'86B') Nesne kayıtları geçerli deęil.
- RC2156**
(2156, X'86C') Yanıt kayıtları geçerli deęil.
- RC2158**
(2158, X'86E') Put ileti kaydı işaretleri geçerli deęil.
- RC2159**
(2159, X'86F') Put ileti kayıtları geçerli deęil.
- RC2161**
(2161, X'871 ') Kuyruk yöneticisi susturuyor.
- RC2162**
(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.
- RC2173**
(2173, X'87D') Put-message seçenek yapısı geçerli deęil.
- RC2184**
(2184, X'888 ') Uzak kuyruk adı geçersiz.
- RC2188**
(2188, X'88C') Çaęrı küme iş yükü çıkışı tarafından reddedildi.

RC2189

(2189, X'88D') Küme adı çözümlemesi başarısız oldu.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RC2196

(2196, X'894 ') Bilinmeyen iletim kuyruğu.

RC2197

(2197, X'895 ') Bilinmeyen varsayılan iletim kuyruğu.

RC2198

(2198, X'896 ') Varsayılan iletim kuyruğu yerel değil.

RC2199

(2199, X'897 ') Varsayılan iletim kuyruğu kullanım hatası.

RC2258

(2258, X'8D2') Grup tanıtıcısı geçersiz.

RC2248

(2248, X'8C8') İleti tanımlayıcı uzantısı geçersiz.

RC2219

(2219, X'8AB') MQI çağrısı önceki arama tamamlanmadan yeniden girildi.

RC2249

(2249, X'8C9') İleti işaretleri geçerli değil.

RC2250

(2250, X'8CA') İleti sıra numarası geçersiz.

RC2251

(2251, X'8CB') İleti kesimi görelî konumu geçersiz.

RC2252

(2252, X'8CC') Özgün uzunluk geçersiz.

RC2253

(2253, X'8CD') İleti bölümündeki verilerin uzunluğu sıfır.

RC2255

(2255, X'8CF') Kuyruk yöneticisinin kullanabileceği iş birimi yok.

RC2257

(2257, X'8D1') Yanlış MQMD sürümü sağlandı.

RC2266

(2266, X'8DA') Küme iş yükü çıkışı başarısız oldu.

RC2269

(2269, X'8DD') Küme kaynağı hatası.

RC2270

(2270, X'8DE') Kullanılabilir hedef kuyruk yok.

RC2420

(2420) Bir MQPUT1 çağrısı yayınlandı, ancak ileti verileri geçersiz bir MQEPH yapısı içeriyor.

RC2551

(2551, X'9F7') Belirtilen seçim dizgisi kullanılamıyor.

RC2554

(2554, X'9FA') İletinin genişletilmiş ileti seçici ile bir aboneye teslim edilip edilmeyeceğini belirlemek için ileti içeriği ayrıştırılmadı.

RPG Bildirimi

C*..1.....2.....3.....4.....5.....6.....7..
C CALLP MQPUT1(HCONN : OBJDSC : MSGDSC :

C
C

PMO : BUFLN : BUFFER :
CMPCOD : REASON)

Çağrıya ilişkin prototip tanımlaması:

```
D*..1.....2.....3.....4.....5.....6.....7..  
DMQPOT1 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

IBM i üzerinde MQSET (Nesne özniteliklerini ayarla)

MQSET çağrısı, bir tanıtıcının gösterdiği bir nesnenin özniteliklerini değiştirmek için kullanılır. Nesne bir kuyruk olmalıdır.

- “Sözdizimi” sayfa 1326
- “Kullanım notları” sayfa 1326
- “Parametreler” sayfa 1327
- “RPG Bildirimi” sayfa 1330

Sözdizimi

MQSET (*HCONN*, *HOBJ*, *SELCNT*, *SELS*, *IACNT*, *INTATR*, *CALEN*, *CHRATR*, *CMPCOD*, *REASON*)

Kullanım notları

1. Bu çağrıyı kullanarak uygulama, bir tamsayı öznitelikleri dizisi ya da bir karakter özniteliği dizileri derlemi ya da her ikisini birden belirtebilir. Hata ortaya çıkmazsa, belirtilen özniteliklerin tümü aynı anda ayarlanır. Bir hata oluşursa (örneğin, bir seçici geçerli değilse ya da bir özniteliği geçerli olmayan bir değere ayarlama girişiminde bulunulursa), çağrı başarısız olur ve öznitelik ayarlanmaz.
2. Özniteliklerin değerleri MQINQ çağrısı kullanılarak belirlenebilir; Ayrıntılar için bkz. [“IBM i üzerinde MQINQ \(Nesne öznitelikleri hakkında sorgu\)” sayfa 1283](#).
- Not:** MQINQ çağrısı kullanılarak sorulabilecek değerleri olan tüm özniteliklerin değerleri MQSET çağrısı kullanılarak değiştirilemez. Örneğin, bu çağrıyla hiçbir işlem nesnesi ya da kuyruk yöneticisi özniteliği ayarlanamaz.
3. Öznitelik değişiklikleri, kuyruk yöneticisinin yeniden başlatılmasından sonra korunur (kuyruk yöneticisinin yeniden başlatılmasından sonra devam etmeyen geçici dinamik kuyruklarda yapılan değişiklikler dışında).
4. MQSET çağrısının kullanıldığı bir model kuyruğunun özniteliklerini değiştiremezsiniz. Ancak, MQOPEN çağrısıyla MQOO_SET seçeneğini kullanarak bir model kuyruğu açarsanız, MQOPEN çağrısıyla yaratılan dinamik yerel kuyruğun özniteliklerini ayarlamak için MQSET çağrısı kullanabilirsiniz.
5. Ayarlanmakta olan nesne bir küme kuyruğuysa, açmanın başarılı olması için küme kuyruğunun yerel bir eşgörünümü olmalıdır.

Nesne öznitelikleriyle ilgili ek bilgi için aşağıdaki konuya bakın:

- [“Kuyruklara ilişkin öznitelikler” sayfa 1345](#)
- [“Ad listesi öznitelikleri” sayfa 1373](#)
- [“IBM i üzerinde süreç tanımlamalarına ilişkin öznitelikler” sayfa 1374](#)
- [“IBM i üzerindeki kuyruk yöneticisine ilişkin öznitelikler” sayfa 1376](#)

Parametreler

MQSET çağrısındaki deęiřtirgeler řunlardır:

HCONN (10 basamaklı iřaretli tamsayı)-giriř

Baęlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik baęlantıyı gösterir. HCONN deęeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

HOBJ (10 basamaklı iřaretli tamsayı)-giriř

Nesne tanıtıcısı.

Bu tanıtıcı, ayarlanacak öznitelikleri olan kuyruk nesnesini gösterir. Tanıtıcı deęeri, OOSSET seçeneęini belirten önceki bir MQOPEN çağrısı tarafından döndürüldü.

SELCNT (10 basamaklı iřaretli tamsayı)-giriř

Seçicilerin sayısı.

Bu, SELS dizisinde saęlanan seçicilerin sayısıdır. Ayarlanacak özniteliklerin sayısıdır. Sıfır geçerli bir deęerdir. İzin verilen üst sınır 256 'dır.

SELS (10 basamaklı iřaretli tamsayı x SELCNT)-giriř

Öznitelik seçicilerin dizisi.

Bu bir **SELCNT** öznitelik seçicileri dizisidir; her seçici, ayarlanacak bir deęere sahip bir öznitelięi (tamsayı ya da karakter) tanımlar.

Her seçici, HOBJ ' in temsil ettięi kuyruk tipi için geçerli olmalıdır. Yalnızca belirli IA* ve CA* deęerlerine izin verilir; bu deęerler bu bölümde daha sonra listelenir.

Seçiciler herhangi bir sırada belirtilebilir. Tamsayı öznitelik seçicilere (IA* seçicileri) karşılık gelen öznitelik deęerleri, INTATR içinde bu seçicilerin SELS içinde gerçekteřtięi sırayla belirtilmelidir. Karakter öznitelięi seçicilere (CA* seçicileri) karşılık gelen öznitelik deęerleri, CHRATR içinde bu seçicilerin oluřtuęu sırayla belirtilmelidir. IA* seçiciler CA* seçicileriyle serpiřtirilebilirler; yalnızca her tip içindeki görelisi sıra önemlidir.

Aynı seçicinin bir kereden fazla belirtilmesi bir hata deęildir; bu yapıldıysa, belirli bir seçici için belirtilen son deęer geçerli olur.

Not:

1. Tamsayı ve karakter öznitelik seçicileri iki farklı aralık içinde ayrılır; IA* seçicileri IAFRST-IALAST aralığında ve CA* seçicileri CAFRST-CALAST aralığında bulunur.

Her bir aralık için IALSTU ve CALSTU deęiřmezleri, kuyruk yöneticisinin kabul edeceęi en yüksek deęeri tanımlar.

2. Önce tüm IA* seçicileri ortaya çıkarsa, SELS ve INTATR dizilerinde karşılık gelen öęeleri adreslemek için aynı öęe numaraları kullanılabilir.

Ayarlanabilen öznitelikler ařaęıdaki tabloda listelenir. Bu çağrı kullanılarak başka bir öznitelik ayarlanamaz. CA* öznitelik seçicileri için, CHRATR içinde gerekli olan dizginin uzunluęunu bayt cinsinden tanımlayan deęiřmez parantez içinde saęlanır.

Çizelge 751. Kuyruklar için MQSET öznitelik seçicileri		
Seçici	Açıklama	Not
KATRGD	Tetikleme verileri (LNTRGD).	<u>"2" sayfa 1328</u>
İADIST	Dağıtım listesi desteği.	<u>"1" sayfa 1328</u>
IAIGET	Alma işlemlerine izin verilip verilmediğini belirler.	
IAIPUT	Koyma işlemlerine izin verilip verilmediğini belirler.	
IATRGC	Tetikleme kontrolü.	<u>"2" sayfa 1328</u>
IATRGD	Tetikleme derinliği.	<u>"2" sayfa 1328</u>
IATRGP	Tetikleyiciler için eşik ileti önceliği.	<u>"2" sayfa 1328</u>
IATRGT	Tetikleyici tipi.	<u>"2" sayfa 1328</u>

Notlar:

1. Yalnızca aşağıdaki altyapılarda desteklenir:

-  AIX
-  IBM i
-  Windows

ve bu sistemlere bağlı IBM MQ istemcileri için.

2. VSE/ESA üzerinde desteklenmez.

IACNT (10 basamaklı işaretli tamsayı)-giriş

Tamsayı özniteliklerinin sayısı.

Bu, INTATR dizisindeki öğelerin sayısıdır ve en az **SELS** parametresindeki IA* seçicilerinin sayısı olmalıdır. Sıfır, yoksa geçerli bir değerdir.

INTATR (10 basamaklı imzalı tamsayı x rxIACNT)-giriş

Tamsayı öznitelikleri dizisi.

Bu, IACNT tamsayı öznitelik değerlerinin bir dizisidir. Bu öznitelik değerleri, SELS dizisindeki IA* seçicileri ile aynı sırada olmalıdır.

CALEN (10 basamaklı işaretli tamsayı)-giriş

Karakter öznitelikleri arabelleğinin uzunluğu.

Bu, **CHRATR** parametresinin bayt cinsinden uzunluğudur ve en azından SELS dizisinde belirtilen karakter özniteliklerinin uzunluklarının toplamı olmalıdır. SELS içinde CA* seçicisi yoksa sıfır geçerli bir değerdir.

CHRATR (1 baytlık karakter dizisi x CALEN)-giriş

Karakter öznitelikleri.

Bu, karakter özniteliği değerlerini içeren, birbirine bitleştirilmiş arabellektir. Arabelleğin uzunluğu **CALEN** değıştirgesiyle verilir.

Karakter öznitelikleri, SELS dizisindeki CA* seçicileri ile aynı sırada belirtilmelidir. Her karakter özniteliğinin uzunluğu sabittir (bkz. SELS). Bir öznitelik için ayarlanacak değer, özniteliğin tanımlı uzunluğundan daha az boşluk olmayan karakter içeriyorsa, öznitelik değerinin özniteliğin tanımlı uzunluğuyla eşleşmesi için CHRATR içindeki değerın sağa doğru boşluklarla doldurması gerekir.

CMPCOD (10 basamaklı işaretili tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretili tamsayı)-çıkış

Neden kodu niteleyici CMPCOD.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCFAIL ise:

RC2219

(2219, X'8AB') MQI çağırısı önceki arama tamamlanmadan yeniden girildi.

RC2006

(2006, X'7D6') Karakter özniteliklerinin uzunluğu geçersiz.

RC2007

(2007, X'7D7') Karakter öznitelikleri dizisi geçersiz.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2019

(2019, X'7E3') Nesne tanıtıcısı geçerli değil.

RC2020

(2020, X'7E4') İnhibe-alma ya da engelleme-koyma kuyruğu özniteliği değeri geçerli değil.

RC2021

(2021, X'7E5') Tamsayı özniteliklerinin sayısı geçersiz.

RC2023

(2023, X'7E7') Tamsayı öznitelikleri dizisi geçerli değil.

RC2040

(2040, X'7F8') Kuyruk ayarlanmak üzere açılmadı.

RC2041

(2041, X'7F9') Nesne tanımı açıldıktan sonra değıştirildi.

RC2101

(2101, X'835 ') Nesne zarar gördü.

RC2052

(2052, X'804 ') Kuyruk silindi.

RC2058

(2058, X'80A') Kuyruk yöneticisi adı geçerli değil ya da bilinmiyor.

RC2059

(2059, X'80B') Kuyruk yöneticisi bağlantı için kullanılamıyor.

RC2162

(2162, X'872 ') Kuyruk yöneticisi kapatılıyor.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2065

(2065, X'811 ') Seçici sayısı geçerli değil.

RC2067

(2067, X'813 ') Öznitelik seçici geçersiz.

RC2066

(2066, X'812 ') Seçicilerin sayısı çok büyük.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2075

(2075, X'81B') Tetikleme denetimi özniteliğine ilişkin değer geçersiz.

RC2076

(2076, X'81C') Tetikleme derinliği özniteliğine ilişkin değer geçersiz.

RC2077

(2077, X'81D') Tetikleyici ileti önceliği özniteliğine ilişkin değer geçersiz.

RC2078

(2078, X'81E') Tetikleyici tipi özniteliğine ilişkin değer geçersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RPG Bildirimi

```

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)

```

Çağrıya ilişkin prototip tanımlaması:

```

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 IBM i üzerinde MQSETMP (İleti tanıtıcı özelliğini ayarla)

MQSETMP çağırısı, ileti tanıtıcısının bir özelliğini ayarlar ya da değiştirir.

- “Sözdizimi” sayfa 1331
- “Kullanım notları” sayfa 1331
- “Parametreler” sayfa 1332
- “RPG Bildirimi” sayfa 1335

Sözdizimi

MQSETMP (*Hconn, Hmsg, SetPropOpts, Name, PropDesc, Type, ValueLength, Value, CompCode, Reason*)

Kullanım notları

- Bu çağırışı yalnızca kuyruk yöneticisinin kendisi iş birimini koordine ettiğinde kullanabilirsiniz. Bu şöyle olabilir:
 - Değişikliklerin yalnızca IBM MQ kaynaklarını etkilediği yerel bir iş birimi.
 - Değişikliklerin diğer kaynak yöneticilerine ait kaynakları etkileyebileceği ve IBM MQ kaynaklarını etkileyebileceği genel bir iş birimi.
- Yerel ve genel iş birimleriyle ilgili daha fazla ayrıntı için bkz. “MQBEGIN (İş birimini başlat)- IBM i” sayfa 1233.
- Kuyruk yöneticisinin iş birimini koordine etmediği ortamlarda, MQBACK yerine uygun geri alma çağrılarını kullanın. Ortam, uygulamanın olağandışı sonlanmasından kaynaklanan örtük bir geri dönüşü de destekleyebilir.
 - z/OS' da aşağıdaki çağrılarını kullanın:
 - İş birimi yalnızca IBM MQ kaynaklarını etkiliyorsa, toplu iş programları (IMS toplu DL/I programları da içinde olmak üzere) MQBACK çağrılarını kullanabilir. Ancak, iş birimi hem IBM MQ kaynaklarını, hem de diğer kaynak yöneticilerine ait kaynakları (örneğin, Db2) etkiliyorsa, z/OS Recoverable Resource Service (RRS) tarafından sağlanan SRRBACK çağrısına bakın. SRRBACK çağrısı, RRS koordinasyonu için etkinleştirilen kaynak yöneticilerine ait kaynaklarda yapılan değişiklikleri geri çevirir.
 - CICS uygulamaları, iş birimini geri almak için EXEC CICS SYNCPOINT ROLLBACK komutunu kullanmalıdır. CICS uygulamaları için MQBACK çağrılarını kullanmayın.
 - IMS uygulamaları (toplu DL/I programlarından başka), iş birimini geri almak için ROLB gibi IMS çağrılarını kullanmalıdır. IMS uygulamaları (toplu DL/I programları dışında) için MQBACK çağrılarını kullanmayın.
 - IBM işletim sistemi üzerinde, kuyruk yöneticisi tarafından eşgüdümlü yerel iş birimleri için bu çağırışı kullanın. Bu, iş düzeyinde bir kesinleştirme tanımının var olmaması gerektiği, yani iş için **CMTSCOPE (*JOB)** parametresini içeren STRCMTCTL komutunun yayınlanmamış olması gerektiği anlamına gelir.
- Bir uygulama, bir iş biriminde kesinleştirilmemiş değişikliklerle sona ererse, bu değişikliklerin atılması uygulamanın olağan şekilde mi, yoksa olağandışı olarak mı bittiğine bağlıdır. Daha fazla ayrıntı için “IBM i üzerinde MQDISC (Kuyruk yöneticisinin bağlantısını kes)” sayfa 1268 içindeki kullanım notlarına bakın.
- Bir uygulama, mantıksal ileti gruplarına ya da bölümlerine ileti girdiğinde ya da aldığı anda, kuyruk yöneticisi, son başarılı MQPUT ve MQGET çağrılarında ilişkin ileti grubu ve mantıksal iletiyle ilgili bilgileri saklar. Bu bilgiler kuyruk tanıtıcısıyla ilişkilidir ve aşağıdaki gibi şeyleri içerir:

- MQMD 'deki *GroupId*, *MsgSeqNumber*, *Offset* ve *MsgFlags* alanlarının değerleri.
- İletinin bir iş biriminin parçası olup olmadığı.
- MQPUT çağrısı için: İletinin kalıcı mı, kalıcı mı olduğu.

Kuyruk yöneticisi, aşağıdakilerin her biri için bir küme olmak üzere üç grup ve bölüm bilgisi kümesini tutar:

- Son başarılı MQPUT çağrısı (bu, bir iş biriminin parçası olabilir).
- Kuyruktan bir iletiyi kaldıran son başarılı MQGET çağrısı (bu, bir iş biriminin parçası olabilir).
- Kuyruktaki bir iletiye göz atan son başarılı MQGET çağrısı (bu, bir iş biriminin parçası olamaz).

Uygulama iletileri bir iş biriminin parçası olarak yerleştirir ya da alırsa ve uygulama iş birimini geri çevirmeye karar verirse, grup ve bölüm bilgileri daha önce sahip olduğu değere geri yüklenir:

- MQPUT çağrısıyla ilişkili bilgiler, yürürlükteki iş biriminde o kuyruk tanıtıcısı için başarılı olan ilk MQPUT çağrıdan önce sahip olduğu değere geri yüklenir.
- MQGET çağrısıyla ilişkili bilgiler, yürürlükteki iş biriminde o kuyruk tanıtıcısı için ilk başarılı MQGET çağrıdan önce sahip olduğu değere geri yüklenir.

İş birimi başlatıldıktan sonra uygulama tarafından güncellenen, ancak iş birimi kapsamı dışında kalan kuyruklarda, iş birimi geriletildiyse, grup ve bölüm bilgileri geri yüklenmez.

Bir iş birimi geriletildiğinde grup ve bölüm bilgilerinin önceki değerine geri yüklenmesi, uygulamanın çeşitli iş birimlerine birçok kesim içeren büyük bir ileti grubu ya da büyük bir mantıksal ileti yaymasına ve iş birimlerinden birinin başarısız olması durumunda ileti grubunda ya da mantıksal iletide doğru noktada yeniden başlamasına olanak sağlar.

Yerel kuyruk yöneticisinde yalnızca sınırlı kuyruk saklama alanı varsa, birkaç iş birimi kullanılması avantajlı olabilir. Ancak, bir sistem arızası ortaya çıkarsa, uygulamanın iletileri doğru noktada yeniden koymayı ya da almayı yeniden başlatabilmesi için yeterli bilgiyi koruması gerekir.

Bir sistem hatasından sonra doğru noktada nasıl yeniden başlatılacağına ilişkin ayrıntılar için [PMOPT \(10 basamaklı imzalı tamsayı\)](#) içinde açıklanan [PMLOGO](#) seçeneğine ve [GMOPT \(10 basamaklı işaretli tamsayı\)](#) içinde açıklanan [GMLOGO](#) seçeneğine bakın.

Geri kalan kullanım notları yalnızca kuyruk yöneticisi iş birimlerini koordine ettiğinde geçerlidir:

- Bir iş birimi, bağlantı tanıtıcısıyla aynı kapsama sahiptir. Belirli bir iş birimini etkileyen tüm IBM MQ çağrılarının aynı bağlantı tanıtıcısı kullanılarak gerçekleştirilmesi gerekir. Farklı bir bağlantı tanıtıcısı kullanılarak verilen çağrılar (örneğin, başka bir uygulama tarafından verilen çağrılar) farklı bir iş birimini etkiler. Bağlantı tanıtıcılarının kapsamına ilişkin bilgi için [HCONN \(10 basamaklı işaretli tamsayı\)-output](#) başlıklı konuya bakın.
- Yalnızca yürürlükteki iş biriminin bir parçası olarak konan ya da alınan iletiler bu çağrıdan etkilenir.
- Bir iş birimi içinde MQGET, MQPUT ya da MQPUT1 çağrılarını veren, ancak hiçbir zaman kesinleştirme ya da geriletme çağrısı yapmayan uzun süreli bir uygulama, kuyrukları diğer uygulamalar tarafından kullanılmayan iletilerle doldurabilir. Bu olasılığa karşı koruma sağlamak için, denetiminin **MaxUncommittedMsgs** kuyruk yöneticisi özneteliğini, kuyrukları dolduran, ancak beklenen ileti sistemi uygulamalarının düzgün çalışmasını sağlayacak kadar yüksek olan kaçak uygulamaları önleyecek kadar düşük bir değere ayarlaması gerekir.

Parametreler

MQSETMP çağrısındaki değişirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir.

Değer, **HMSG** değiştirgesinde belirtilen ileti tanıtıcısını yaratmak için kullanılan bağlantı tanıtıcısıyla eşleşmelidir.

İleti tanıtıcısı HCUNAS kullanılarak yaratıldıysa, ileti tanıtıcısının bir özelliğini ayarlarken iş parçacığında geçerli bir bağlantı kurulmalıdır; tersi durumda, çağrı RC2009 neden koduyla başarısız olur.

HMSG (20 basamaklı işaretli tamsayı)-giriş

Bu, değiştirilecek ileti tanıtıcısıdır. Değer, önceki bir MQCRTMH çağrısı tarafından döndürüldü.

SETOPT (MQSMPO)-giriş

İleti özelliklerinin nasıl ayarlandığını denetleyin.

Bu yapı, uygulamaların ileti özelliklerinin nasıl ayarlanacağını denetleyen seçenekleri belirtmesine olanak sağlar. Yapı, MQSETMP çağrısındaki bir giriş değiştirgesidir. Daha fazla bilgi için bkz. [MQSMPO](#).

PRNAME (MQCHARV)-giriş

Bu, ayarlanacak özelliğin adıdır.

Özellik adlarının kullanımıyla ilgili ek bilgi için [Özellik adları](#) ve [Özellik adı kısıtlamaları](#) konusuna bakın.

PRPDSC (MQPD)-giriş/çıkış

Bu yapı, aşağıdakiler de içinde olmak üzere bir özelliğin özniteliklerini tanımlamak için kullanılır:

- özellik desteklenmiyorsa ne olur
- özelliğin ait olduğu ileti bağlamı
- Akarken özelliğin hangi iletilere kopyalandığı

Bu yapıyla ilgili ek bilgi için [MQPD](#) başlıklı konuya bakın.

TYPE (10 basamaklı işaretli tamsayı)-giriş

Ayarlanmakta olan özelliğin veri tipi. Aşağıdakilerden biri olabilir:

TYPBOL

Bir boole. *ValueLength* 4 olmalıdır.

TYPBST

Bir bayt dizilimi. *ValueLength* sıfır ya da daha büyük olmalıdır.

TYPI8

8 bit işaretli tamsayı. *ValueLength* 1 olmalıdır.

TYPI16

16 bit işaretli tamsayı. *ValueLength* 2 olmalıdır.

TYPI32

32 bit işaretli tamsayı. *ValueLength* 4 olmalıdır.

TYPI64

64 bit işaretli tamsayı. *ValueLength* 8 olmalıdır.

TYPF32

32 bit kayan noktalı sayı. *ValueLength* 4 olmalıdır.

TYPF64

64 bit kayan noktalı sayı. *ValueLength* 8 olmalıdır.

TYPSTR

Bir karakter dizisi. *ValueLength* sıfır ya da daha büyük ya da VLNULL özel değeri olmalıdır.

TYPNUL

Özellik var, ancak boş değerli. *ValueLength* sıfır olmalıdır.

VALLEN (10 basamaklı işaretli tamsayı)-giriş

Değer parametresindeki özellik değerinin bayt cinsinden uzunluğu.

Sıfır yalnızca boş değerler, dizgiler ya da byte dizgileri için geçerlidir. Sıfır, özelliğin var olduğunu, ancak değer karakter ya da bayt içermediğini gösterir.

Tip deęiřtirgesinde TYPSTR ayarlandıysa, deęer sıfırdan byk ya da sıfıra eřit ya da ařaęıdaki zel deęer olmalıdır:

VLNULL (boř deęerli)

Deęer, dizgide karřılařılan ilk boř deęerle sınırlandı. Boř deęer, dizginin bir parçası olarak ięerilmez. TYPSTR de ayarlanmamıřsa, bu deęer geęersizdir.

Not: VLNULL ayarlanırsa, bir dizgiyi sonlandırmak ięin kullanılan boř karakter, Deęerin karakter takımındaki boř deęerdir.

VALUE (1 baytlık bit dizgi x VALLEN)-giriř

Ayarlanacak zellięin deęeri. Arabellek, deęerdeki verilerin nitelięine uygun bir sınırdaki hizalanmalıdır.

C programlama dilinde, parametre bir gsterge-to-void olarak bildirilir; herhangi bir veri tipinin adresi parametre olarak belirtilebilir.

ValueLength sıfır ise, *Deęer* seęeneęine bařvurmaz. Bu durumda, C ya da System/390 ęevirici dilinde yazılmıř programlardan geęirilen parametre adresi boř olabilir.

CMPCOD (10 basamaklı iřaretli tamsayı)-çıkıř

Tamamlanma kodu; ařaęıdakilerden biridir:

CCOK

Bařarıyla tamamlandı.

CCFAIL

Arama bařarısız oldu.

REASON (10 basamaklı iřaretli tamsayı)-çıkıř

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak ięin bir neden yok.

CMPCOD CCWARN ise:

RC2421

(2421, X'0975 ') zellikleri ięeren bir MQRFH2 klasr ayrıřtırılmadı.

CMPCOD CCFAIL ise:

RC2204

(2204, X'089C') Baędařtırıcı kullanılmıyor.

RC2130

(2130, X'852 ') Baędařtırıcı hizmet modl yklenemiyor.

RC2157

(2157, X'86D') Birincil ve ana ASID' ler farklıdır.

RC2004

(2004, X'07D4') Deęer parametresi geęersiz.

RC2005

(2005, X'07D5') Deęer uzunluęu parametresi geęersiz.

RC2219

(2219, X'08AB') nceki arama tamamlanmadan nce girilen MQI ęaęırısı.

RC2460

(2460, X'099C') İleti tanıtıcısı gstergesi geęersiz.

RC2499

(2499, X'09C3') İleti tanıtıcısı kullanımda.

RC2046

(2046, X'07FE') Seęenekler geęerli deęil ya da tutarlı deęil.

RC2482

(2482, X'09B2') Özellik tanımlayıcı yapısı geçersiz.

RC2442

(2442, X'098A') Özellik adı geçersiz.

RC2473

(2473, X'09A9') Özellik veri tipi geçersiz.

RC2472

(2472, X'09A8') Değer verilerinde sayı biçimi hatası saptandı.

RC2463

(2463, X'099F') İleti özellik seçenekleri yapısını ayarla geçerli değil.

RC2111

(2111, X'083F') Özellik adı kodlanmış karakter takımı tanıtıcısı geçersiz.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

Daha fazla ayrıntı için bkz. [“IBM i \(ILE RPG\) için dönüş kodları” sayfa 1402](#) .

RPG Bildirimi

```

C*..1.....2.....3.....4.....5.....6.....7..
C                               CALLP      MQSETMP(HCONN : HMSG : SETOPT :
                               PRNAME : PRPDSC :
                               TYPE : VALLEN : VALUE :
                               CMPCOD : REASON)

```

Çağrıya ilişkin prototip tanımlaması:

```

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 üzerinde MQSTAT (Durum bilgilerini al)**

Durum bilgilerini almak için MQSTAT çağrısına bakın. Döndürülen durum bilgisi tipi, çağrıda belirtilen STYPE değerine göre belirlenir.

- [“Sözdizimi” sayfa 1336](#)
- [“Kullanım notları” sayfa 1336](#)
- [“Parametreler” sayfa 1336](#)
- [“RPG Bildirimi” sayfa 1337](#)

Sözdizimi

MQSTAT (*HCONN*, *STYPE*, *STAT*, *CMPCOD*, *REASON*)

Kullanım notları

1. Bir STATAPT tipi belirten MQSTAT çağrısı, önceki zamanuyumsuz MQPUT ve MQPUT1 işlemleriyle ilgili bilgi döndürür. Çağrıya geçirilen MQSTAT yapısı, o bağlantıya ilişkin kaydedilen ilk zamanuyumsuz uyarı ya da hata bilgileriyle tamamlanır. Daha fazla hata ya da uyarı ilkinin izlerse, normalde bu değerleri değiştirmezler. Ancak, CCWARN tamamlanma koduyla bir hata oluşursa, bunun yerine CCFAIL tamamlanma koduyla bir hata döndürülür.
2. Bağlantı kurulduğundan bu yana ya da MQSTAT ' a son çağrıldıktan sonra bir CCOK CMPCOD ve RCNONE nedeni döndürüldü.
3. Bağlantı tanıtıcısı altında işlenen zamanuyumsuz çağrı sayısı üç sayaç kullanılarak döndürülür; STSPSC, STSPWC ve STSPFC. Bu sayaçlar, zamanuyumsuz bir işlem her başarıyla işlendiğinde, bir uyarı geldiğinde ya da başarısız olduğunda kuyruk yöneticisi tarafından artırılır (muhasabe amacıyla bir dağıtım listesine konan sayıların, dağıtım listesi başına bir kereden fazla değil, hedef kuyruk başına bir kez sayıldığını unutmayın).
4. MQSTAT ' ye yapılan başarılı bir çağrı, önceki hata bilgilerinin ya da sayımları sıfırlanmasıyla sonuçlanır.

Parametreler

MQSTAT çağrısındaki deęiřtirgeler řunlardır:

Hconn (MQHCONN)-giriř

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* deęeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

STYPE (10 basamaklı iřaretli tamsayı)-giriř

İstenen durum bilgilerinin tipi. Geçerli olan tek deęer:

DURUM

Önceki zamanuyumsuz koyma işlemlerine ilişkin bilgileri döndürür.

STS (MQSTS)-giriř/çıkıř

Durum bilgisi yapısı. Ayrıntılar için bkz. [“IBM i üzerinde MQSTS \(Durum raporlama yapısı\)” sayfa 1211.](#)

CMPCOD (10 basamaklı iřaretli tamsayı)-çıkıř

Tamamlanma kodu; ařaęıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı iřaretli tamsayı)-çıkıř

Neden kodu niteleyici *CMPCOD*.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCFAIL ise:

RC2374

(2374, X' 946 ') API çıkıřı başarısız oldu

RC2183

(2183, X'887 ') API çıkıřı yüklenemiyor.

RC2219

(2219, X'8AB') Önceki arama tamamlanmadan önce girilen MQI çağrısı.

RC2009

(2009, X'7D9') Kuyruk yöneticisine bağlantı kaybedildi.

RC2203

(2203, X'89B') Bağlantı kapatılıyor.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2162

(2162, X'872 ') Kuyruk yöneticisi durduruluyor

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2430

(2430, X'97E') MQSTAT tipiyle ilgili hata.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2424

(2424, X' 978 ') MQSTS yapısıyla ilgili hata

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

RC2298

(2298, X'8FA') İstenen işlev yürürlükteki ortamda kullanılamıyor.

Bu kodlara ilişkin ayrıntılı bilgi için aşağıdaki adrese bakın:

- [İletiler ve neden kodları](#)

RPG Bildirimi

```
C*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
C          CALLP          MQSTAT(HCONN : ETYPE : ERR :
C                                CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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 üzerinde MQSUB (Aboneliği Kaydet)

MQSUB çağrısı, belirli bir konuya ilişkin uygulama aboneliğini kaydeder.

- [“Sözdizimi” sayfa 1338](#)
- [“Kullanım notları” sayfa 1338](#)
- [“Parametreler” sayfa 1339](#)
- [“RPG Bildirimi” sayfa 1342](#)

Sözdizimi

MQSUB (HCONN, SUBDSC, HOBJ, HSUB, CMPCOD, REASON)

Kullanım notları

- Abonelik, önceden tanımlanmış bir konu nesnesinin kısa adı, konu dizgisinin tam adı kullanılarak ya da Konu dizgilerinin birleştirilmesibaşlıklı konuda açıklandığı gibi, iki kısmın birleştirilmesiyle oluşturulur.
 - Kuyruk yöneticisi, bir MQSUB çağrısı yayınlandığında, uygulamanın çalıştırıldığı kullanıcı kimliğinin erişime izin verilmeden önce uygun yetki düzeyine sahip olduğunu doğrulamak için güvenlik denetimleri gerçekleştirir. Uygun konu nesnesi, çağrıda sağlanan kısa bir adla ya da uzun bir ad sağlanırsa, konu sıradüzeninde bulunan en yakın kısa ad nesnesiyle bulunur. Bu konu nesnesi üzerinde, abone olma yetkisinin ayarlandığından emin olmak için bir yetki denetimi yapılır ve çıkışa ilişkin yetkinin ayarlandığından emin olmak için hedef kuyrukta bir yetki denetimi yapılır. SDMAN seçeneği kullanılırsa, bu, bu konu nesnesiyle ilişkili yönetilen kuyruk adında bir yetki denetimi yapıldığı ve yönetilmeyen bir kuyruk sağlandığı anlamına gelir; bu, **HOBJ** parametresiyle gösterilen kuyrukta bir yetki denetimi yapıldığı anlamına gelir.
 - SOMAN seçeneği kullanıldığında MQSUB çağrısında döndürülen **HOBJ**, Gerileme eşiği ve Aşırı geriletme yeniden kuyruğa alma adı gibi öznitelikleri bulmak için sorulabilir. Yönetilen kuyruğun adını da sorabilirsiniz, ancak bu kuyruğu doğrudan açmayı denememelisiniz.
 - Abonelikler, birden fazla grubun yayınıla eşleştiği abonelikler grubuna yalnızca tek bir yayının teslim edilmesine izin vererek gruplandırılabilir. Abonelikler SOGRP seçeneği kullanılarak gruplandırılır ve abonelikleri gruplamak için şunları yapmanız gerekir:
 - aynı adlandırılmış kuyruğu (SOMAN seçeneğini kullanmayan) aynı kuyruk yöneticisinde kullanın-MQSUB çağrısında **HOBJ** değiştirilmesiyle gösterilir
 - aynı *SDCID* paylaşımını paylaşma
 - aynı *SDSL*
- Bu öznitelikler, grupta olduğu kabul edilen abonelik kümesini tanımlar ve bir abonelik gruplandığında değiştirilemeyen özniteliklerdir. *SDSL* değiştirmesi RC2512 ile sonuçlanır ve diğerlerinden herhangi birinin değiştirilmesi (abonelik gruplanmamışsa değiştirilebilir) RC2515 ile sonuçlanır.
- MQSD ' deki alanlar, SORES seçeneğini kullanan bir MQSUB çağrısından dönüşte tamamlanır. Döndürülen MQSD, MQSD ' ye uygulanan abonelikte yapmanız gereken değişikliklerle SOALT seçeneğini kullanan bir MQSUB çağrısına doğrudan geçirilebilir. Bazı alanlarda, çizelgede belirtildiği gibi, dikkat edilmesi gereken özel noktalar vardır.

Çizelge 752. MQSUB ' den MQSD çıkışı	
MQSD ' de alan adı	Dikkat edilmesi gereken özel noktalar
Erişim ya da yaratma seçenekleri	MQSUB çağrısından dönüşte bu seçeneklerin hiçbiri ayarlanmadı. MQSD ' yi daha sonra bir MQSUB çağrısında yeniden kullanırsanız, belirtik olarak ayarlanması gerekir.
Dayanıklılık seçenekleri, Hedef seçenekleri, Kayıt Seçenekleri ve Genel Arama Kartı seçenekleri	Bu seçenekler uygun şekilde ayarlanacak
Yayınlama seçenekleri	Bu seçenekler, yalnızca SOCRE için geçerli olan SONEWP dışında uygun şekilde ayarlanacaktır.
Diğer seçenekler	Bu seçenekler, bir MQSUB çağrısından dönüşte değiştirilmez. API çağrılarının nasıl yayınlandığını ve abonelikte birlikte nasıl saklanmadığını denetler. Bunlar, MQSD ' yi yeniden kullanan sonraki MQSUB çağrılarında gerektiği şekilde ayarlanmalıdır.
ObjectName	Bir MQSUB çağrısından dönüşte yalnızca bu giriş alanı değiştirilmez.
ObjectString	Bir MQSUB çağrısından dönüşte yalnızca bu giriş alanı değiştirilmez. Bir arabellek sağlanırsa, <i>SDRO</i> alanında kullanılan Tam konu adı döndürülür.

Çizelge 752. MÖSUB ' den MÖSD çıkışı (devamı var)	
MÖSD ' de alan adı	Dikkat edilmesi gereken özel noktalar
AlternateUserTanıtıcısı ve AlternateSecurityTanıtıcısı	Bir MÖSUB çağrısından dönüşte yalnızca bu giriş alanları değişmez. API çağrılarının nasıl yayınlandığını ve abonelikte birlikte nasıl saklanmadığını denetler. Bunlar, MÖSD ' yi yeniden kullanan sonraki MÖSUB çağrılarında gerektiği gibi ayarlanmalıdır.
SubExpiry	SORES seçeneği kullanılarak bir MÖSUB çağrısından döndüğünüzde bu alan, kalan süre bitimine değil, aboneliğin özgün süre bitimine ayarlanır. SOALT seçeneğini kullanarak MÖSD ' yi bir MÖSUB çağrısında yeniden kullanırsanız, aboneliğin süre bitimini yeniden saymaya başlayacak şekilde sıfırlarsınız.
SubName	Bu alan, MÖSUB çağrısındaki bir giriş alanıdır ve çıkışta değiştirilmez.
SubUserVeri ve SelectionString	Bu değişken uzunluklu alanlar, SORES seçeneği kullanılarak bir MÖSUB çağrısının çıkışında döndürülür (bir arabellek sağlandıysa) ve VCHRPiçinde artı bir arabellek uzunluğu da döndürülür. Herhangi bir arabellek sağlanmazsa, MÖCHARV.If ' un VCHRL alanında sağlanan arabellek, alanı döndürmek için gereken alandan daha küçük bir değer döndürülür; sağlanan arabellekte yalnızca VCHRP byte döndürülür. Daha sonra SOALT seçeneğini kullanarak MÖSD ' yi bir MÖSUB çağrısında yeniden kullanırsanız ve bir arabellek sağlanmazsa, ancak sıfır dışında bir VCHRL sağlanırsa, bu uzunluk alanının var olan uzunluğuyla eşleşirse, alanda herhangi bir değişiklik yapılmaz.
SubCorrelTanıtıcısı ve PubAccountingSimgesi	SOSCID kullanmazsanız, SDCID kuyruk yöneticisi tarafından oluşturulur. SOSETI kullanmazsanız, SDACC kuyruk yöneticisi tarafından oluşturulur. Bu alanlar, SORES seçeneği kullanılarak MÖSUB çağrısından MÖSD ' de döndürülür. Bunlar kuyruk yöneticisi tarafından oluşturulursa, oluşturulan değer SOCRE ya da SOALT seçeneği kullanılarak bir MÖSUB çağrısında döndürülür.
PubPriority, SubLevel & PubApplIdentityData	Bu alanlar MÖSD ' de döndürülür.
ResObjectDizesi	Bir arabellek sağlanırsa, MÖSD ' de yalnızca bu çıkış alanı döndürülür.

Parametreler

MÖSUB çağrısındaki değiştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. HCONN değeri önceki bir MÖCONN ya da MÖCONNX çağrısı tarafından döndürüldü.

SUBDSC (MÖSD)-giriş/çıkış

Bu, uygulama tarafından kaydedilmekte olan, kullanımla nesneyi tanımlayan bir yapıdır. Ek bilgi için bkz. "IBM i üzerinde MÖSD (Abonelik tanımlayıcısı)" sayfa 1193 .

HOBJ (10 basamaklı işaretli tamsayı)-giriş/çıkış

Bu tanıtıcı, bu aboneliğe gönderilen iletileri almak için oluşturulan erişimi temsil eder. Bu iletiler belirli bir kuyrukta saklanabilir ya da kuyruk yöneticisinden belirli bir kuyruğa gerek kalmadan saklama alanını yönetmesi istenebilir.

Nesne tanıtıcısı.

Belirli bir kuyruk kullanılacaksa, yaratma sırasında abonelikte ilişkilendirilmelidir. Bu iki şekilde yapılabilir:

- MÖSUB ' yi SDCRT seçeneğiyle çağırırken bu tanıtıcıyı sağlayarak. Bu tanıtıcı çağrı için bir giriş değiştirgesi olarak sağlandıysa, bu tanıtıcı en az bir OOINP*, OOOOUT (örneğin, bir uzak kuyruk) ya

da OOBROW seçeneđi kullanılarak bir kuyruđun önceki MÖOPEN çağrısından döndürülen geçerli bir nesne tanıtıcısı olmalıdır. Bu durumda arama RC2019 ile başarısız olur. Bir konu nesnesine çözülen diđer ad kuyruđuna ilişkin bir nesne tanıtıcısı olamaz. Bu durumda, arama RC2019 ile başarısız olur.

- DEFINE SUB MÖSC komutunu kullanarak ve o komutu bir kuyruk nesnesi adıyla belirterek.

Kuyruk yöneticisi bu aboneliđe gönderilen iletilerin depolanmasını yönetecekse, SOMAN seçeneđini kullanarak ve parametre deđerini HONONE olarak ayarlayarak abonelik oluşturulduğunda bunu belirtmeniz gerekir. Kuyruk yöneticisi, tanıtıcıyı çağrıda çıkış parametresi olarak döndürür ve döndürülen tanıtıcı yönetilen tanıtıcı olarak bilinir. HONONE belirtilirse ve SOMAN da belirtilmezse, çağrı RC2019 ile başarısız olur.

Kuyruk yöneticisi tarafından döndürülen yönetilen bir tanıtıcı, MÖGET ya da MÖCB çağrısında, göz atma seçenekleriyle ya da göz atma seçenekleri olmadan, MÖINÖ çağrısında ya da MÖCLOSE üzerinde kullanılabilir. MÖPUT, MÖSET ya da sonraki bir MÖSUB üzerinde kullanılamaz; bunu yapma girişimi MÖPUT için RC2039 , MÖSET için RC2040 ya da MÖSUB için RC2038 ile başarısız olur.

Bu aboneliđi sürdürmek için MÖSD yapısındaki OPTS alanındaki SORES seçeneđi kullanılırsa, HONONE belirtildiyse, tanıtıcı bu deđiştirgede uygulamaya döndürülebilir. Abonelik yönetilen bir tanıtıcı kullanıyor ya da kullanmıyor olsa da bunu kullanabilirsiniz. DEFINE SUB komutunda tanımlanan abonelik kuyruđuna ilişkin tanıtıcıyı istiyorsanız, bu işlem DEFINE SUB kullanılarak yaratılan abonelikler için yararlı olabilir. Yönetimsel olarak oluşturulan bir aboneliđin sürdürülmesi durumunda, kuyruk OOINPÖ ve OOBROW ile açılır. Başka seçenekler gerekiyorsa, uygulamanın abonelik kuyruđunu açık olarak açması ve çağrıdaki nesne tanıtıcısını sağlaması gerekir. Kuyruk açılırken bir sorun ortaya çıkarsa, arama RC2522 ile başarısız olur. HÖBJ sağlanırsa, özgün MÖSUB çağrısındaki HÖBJ ile eşdeđer olmalıdır. Bu, bir MÖOPEN çağrısından döndürülen bir nesne tanıtıcısı sağlandığında, tanıtıcı deđeri daha önce kullanılan kuyrukla aynı olmalıdır ya da çağrı RC2019 ile başarısız olur.

Bu abonelik deđiştiriliyorsa, MÖSD yapısındaki OPTS alanındaki SOALT seçeneđi kullanılarak farklı bir HÖBJ sağlanabilir. Bu parametre aracılıđıyla daha önce kuyruđa gönderilen yayınlar bu kuyrukta kalır ve HÖBJ parametresi artık farklı bir kuyruđu gösteriyorsa, bu iletileri almak uygulamanın sorumluluđundadır.

Bu parametrenin çeşitli abonelik seçenekleriyle kullanımı aşıđıdaki tabloda özetlenmiştir:

Seçenekler	Höbj.	Açıklama
SOCRT + SOMAN	Girişte yoksayıldı	Kuyruk yöneticisi tarafından yönetilen ileti depolanmasıyla bir abonelik oluşturur.
SOCRT	Geçerli nesne tanıtıcısı	İletilerin hedefi olarak belirli bir kuyruđu sağlayan bir abonelik oluşturur.
SIRALAR	TONON	Önceden yaratılmış bir aboneliđi sürdürür (yönetilen ya da deđeril) ve kuyruk yöneticisinin nesne tanıtıcısını uygulama tarafından kullanılmak üzere döndürmesini sağlar.
SIRALAR	Geçerli, eşleşen, nesne tanıtıcısı	İletiler için hedef olarak belirli bir kuyruđu kullanan önceden oluşturulmuş bir aboneliđi sürdürür ve belirli açık seçeneklerle bir nesne tanıtıcısı kullanır.
SOALT + SOMAN	TONON	Önceden belirli bir kuyruđu kullanan var olan bir aboneliđi şimdi yönetilecek şekilde deđiştirir.
SOALT	Geçerli nesne tanıtıcısı	Var olan bir aboneliđi, belirli bir kuyruđu (yönetilen ya da farklı bir belirli kuyruktan) kullanacak şekilde deđiştirir.

İster sağlanmış, ister döndürülmüş olsun, yayınları almak için gereksinim duyduğunuz sonraki MÖGET çağrılarında HÖBJ belirtilmelidir.

HOBJ tanıtıcısı, *MQCLOSE* çağrısı yayınlandığında ya da tutamacın kapsamını tanımlayan işlem birimi sona erdiğinde geçerli olmaktan çıkar. Döndürülen nesne tanıtıcısının kapsamı, çağrıda belirlenen bağlantı tanıtıcısıyla aynı. Tanıtıcı kapsamına ilişkin bilgi için bkz. [HCONN](#) . *HOBJ* tanıtıcısının *MQCLOSE* ' nin *HSUB* tanıtıcısı üzerinde bir etkisi yoktur.

HSUB (10 basamaklı işaretli tamsayı)-çıkış

Bu tanıtıcı, yapılan aboneliği gösterir. İki işlem daha için kullanılabilir:

- Abonelik gerçekleştirilirken *SOPUBR* seçeneği kullanıldığında yayınların gönderilmesini istemek için sonraki bir *MQSUBRQ* çağrısında kullanılabilir.
- Yapılan aboneliği kaldırmak için sonraki bir *MQCLOSE* çağrısında kullanılabilir. *HSUB* tanıtıcısı, *MQCLOSE* çağrısı yayınlandığında ya da tanıtıcı kapsamını tanımlayan işlem birimi sona erdiğinde geçerli olmaktan çıkar. Döndürülen nesne tanıtıcısının kapsamı, çağrıda belirlenen bağlantı tanıtıcısıyla aynı. *HSUB* tanıtıcısının *MQCLOSE* ' nin *HOBJ* tanıtıcısı üzerinde bir etkisi yoktur.

Bu tanıtıcı bir *MQGET* ya da *MQCB* çağrısına geçirilemez. **HOBJ** parametresini kullanmalısınız. Bu tanıtıcı, *RC2019* ile sonuçlanan başka bir *IBM MQ* çağrısına geçirilmiştir.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu; aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı

CCWARN

Uyarı (kısmi tamamlama)

CCFAIL

Arama başarısız oldu

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CMPCOD *CCOK* ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD *CCFAIL* ise:

RC2019

(2019 X'07E3') Nesne tanıtıcısı geçerli değil

RC2046

(2046 X'07FE') Seçenekler geçerli değil ya da tutarlı değil

RC2085

(2085 X'0825 ') Tanımlanmış nesne bulunamıyor

RC2161

(2161 X'0871 ') Kuyruk yöneticisi susturuyor

RC2298

(2298 X'08FA') İşlev desteklenmiyor.

RC2424

(2424 X'0978 ') Abonelik tanımlayıcısı (*MQSD*) geçerli değil

RC2425

(2441 X' 979 ') Konu dizgisi geçerli değil

RC2428

(2428 X'097C') Belirtilen abonelik adı var olan aboneliklerle eşleşmiyor

RC2429

(2429 X'097D') Abonelik adı var ve başka bir uygulama tarafından kullanılıyor

RC2431

(2431 X'097F') SubUserVeri alanı geçerli değil

RC2432

(2432 X'0980 ') Abonelik var

RC2434

(2434 X'0982 ') Abonelik adı var olan abonelikte eşleşiyor

RC2440

(2440 X'0988 ') SubName alanı geçerli değil

RC2441

(2441 X'0989 ') Nesne dizgisi alanı geçersiz

RC2435

(2435 X'0983 ') Öznitelik SDALT kullanılarak değiştirilemez ya da abonelik SDIMM ile oluşturulmuştur.

RC2436

(2436 X'0984 ') SODUR seçeneği geçerli değil

RC2459

(2459, X'99B') Seçim dizgisi sözdizimi hatası.

RC2503

(2503 X'09C7') MQSUB çağrıları şu anda abone olunan konular için engelleniyor.

RC2519

(2519, X'9D7') Seçim dizgisi, MQCHARV yapısının nasıl kullanılacağına ilişkin tanımda belirtildiği gibi değil.

RC2551

(2551, X'9F7') Belirtilen seçim dizgisi kullanılmıyor.

RPG Bildirimi

```
C*..1.....2.....3.....4.....5.....6.....7..
C                               CALLP      MQSUB(HCONN : SUBDSC : HOBJ :
C                               HSUB : CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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 üzerinde MQSUBRQ (Abonelik İsteği)

MQSUBRQ çağrısı, abonelikte ilgili bir istekte bulunmasını sağlar.

- [“Sözdizimi” sayfa 1343](#)
- [“Kullanım notları” sayfa 1343](#)
- [“Parametreler” sayfa 1343](#)
- [“RPG Bildirimi” sayfa 1344](#)

Sözdizimi

MQSUBRQ (*HCONN*, *HSUB*, *ACTION*, *SUBROPT*, *CMPCOD*, *REASON*)

Kullanım notları

Aşağıdaki kullanım notları SRAPUB kullanımını için geçerlidir:

1. Bu komut başarıyla tamamlanırsa, belirtilen abonelikte eşleşen alıkonan yayınlar aboneliğe gönderilir ve aboneliği yaratan özgün MQSUB yüklemi üzerinde döndürülen HOBJ kullanılarak MQGET ya da MQCB kullanılarak alınabilir.
2. Aboneliği yaratan özgün MQSUB komutu tarafından abone olunan konu bir genel arama karakteri içeriyorsa, birden çok alıkonan yayın gönderilebilir. Bu çağrı sonucunda gönderilen yayınların sayısı, SBROPT yapısındaki *SRNMP* alanında kaydedilir.
3. Bu komut RC2437 neden koduyla tamamlanırsa, belirtilen konu için şu anda alıkonan yayın yoktur.
4. Bu komut RC2525 ya da RC2526 neden koduyla tamamlanırsa, belirtilen konu için alıkonan yayınlar vardır, ancak teslim edilemedikleri anlamına gelen bir hata ortaya çıkmıştır.
5. Bu çağrıyı yapabilmesi için uygulamanın konuya ilişkin geçerli bir aboneliği olmalıdır. Abonelik, uygulamanın önceki bir eşgörünümünde yapıldıysa ve abonelik için geçerli bir tanıtıcı yoksa, uygulamanın bu çağrıda kullanılmak üzere bir tanıtıcı elde etmesi için önce SORES seçeneğiyle MQSUB 'u çağırması gerekir.
6. Yayınlar, bu uygulamanın geçerli aboneliğiyle kullanılmak üzere kayıtlı hedefe gönderilir. Yayınların başka bir yere gönderilmesi gerekiyorsa, abonelik önce SOALT seçeneğiyle MQSUB çağrısı kullanılarak değiştirilmelidir.

Parametreler

MQSUBRQ çağrısındaki değiştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *HCONN* değeri önceki bir MQCONN ya da MQCONNX çağrısı tarafından döndürüldü.

z/OS for CICS uygulamalarında MQCONN çağrısı atlanabilir ve *HCONN* için belirtilen şu değer:

HCDEFH

Varsayılan bağlantı tanıtıcısı.

HSUB (10 basamaklı işaretli tamsayı)-giriş

Bu tanıtıcı, güncelleme isteğinde bulunulacak aboneliği gösterir. Önceki bir MQSUB çağrısından *HSUB* değeri döndürüldü.

İŞLEM (10 basamaklı işaretli tamsayı)-giriş

Bu parametre, abonelikte istenen işlemi denetler. Aşağıdakilerden biri (ve yalnızca biri) belirtilmelidir:

SRAPUB

Bu işlem, belirtilen konu için bir güncelleme yayınının gönderilmesini ister. Olağan durumda, abone aboneliği yaptığı MQSUB çağrısında SOPUBR seçeneğini belirttiğinde bu kullanılır. Kuyruk yöneticisinin konu için alıkonan bir yayını varsa, bu aboneye gönderilir. Yoksa, arama başarısız olur. Bir uygulamaya alıkonan bir yayını gönderilirse, bu, o yayının MQIsRetained ileti özelliğiyle gösterilir.

HSUB değiştirgesiyle gösterilen var olan abonelikteki konu genel arama karakterleri içerebileceğinden, abone birden çok alıkonan yayını alabilir.

SBROPT (MQSRO)-giriş/çıkış

Bu seçenekler MQSUBRQ işlemi denetler, ayrıntılar için bkz. "[MQSRO-Abonelik isteği seçenekleri](#)" sayfa 583 .

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu; aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı

CCWARN

Uyarı (kısmi tamamlama)

CCFAIL

Arama başarısız oldu

Neden (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici *CMPCOD*.

CPMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CPMPCOD CCFAIL ise:

RC2298

2298 (X'08FA') İstenen işlev yürürlükteki ortamda kullanılamıyor.

RC2437

2437 (X'0985 ') Bu konu için saklanan hiçbir yayın yok.

RC2046

2046 (X'07FE') Options parametresi ya da alanı geçerli olmayan seçenekler ya da geçerli olmayan seçenekler birleşimi içeriyor.

RC2161

2161 (X'0871 ') Kuyruk yöneticisi susturuyor

RC2438

2438 (X'0986 ') MQSUBRQ çağrısında, Abonelik İsteği Seçenekleri MQSRO geçerli değil.

RPG Bildirimi

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQSUBRQ(HCONN : HSUB : ACTION :
C          SBROPT : CMPCOD : REASON)
```

Çağrıya ilişkin prototip tanımlaması:

```
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
```

IBM i

IBM i üzerindeki nesnelerin öznitelikleri

Bu konu derlemi, yalnızca MQINQ işlev çağrısının konusu olabilen IBM MQ nesnelerini listeler ve sorulabilecek özniteliklerin ve kullanılacak seçicilerin ayrıntılarını verir.

Kuyruklara ilişkin öznitelikler

Farklı kuyruk tanımlaması tipleri ve her biri tarafından desteklenen öznitelikler hakkında bilgi edinmek için bu bilgileri kullanın.

Kuyruk tipleri: Kuyruk yöneticisi aşağıdaki kuyruk tanımlaması tiplerini destekler:

Yerel kuyruk

Bu, iletileri saklayan fiziksel bir kuyruktur. Kuyruk yerel kuyruk yöneticisinde var.

Yerel kuyruk yöneticisine bağlı uygulamalar, bu tipteki kuyruklara ileti yerleştirebilir ve kuyruklardan ileti kaldırabilir. **QType** kuyruk özniteliğinin değeri QTLOC.

Paylaşılan kuyruk

Bu, iletileri saklayan fiziksel bir kuyruktur. Kuyruk, paylaşılan havuza sahip olan kuyruk paylaşım grubuna ait olan tüm kuyruk yöneticilerinin erişebileceği paylaşılan bir havuzda var.

Kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisine bağlı uygulamalar, bu tipteki kuyruklara ileti yerleştirebilir ve kuyruklardan ileti kaldırabilir. Bu tür kuyruklar, yerel kuyruklarla etkili bir şekilde aynıdır. **QType** kuyruk özniteliğinin değeri QTLOC.

- Paylaşılan kuyruklar yalnızca z/OS üzerinde desteklenir.

Küme kuyruğu

Bu, iletileri saklayan fiziksel bir kuyruktur. Kuyruk, yerel kuyruk yöneticisinde ya da yerel kuyruk yöneticisiyle aynı kümeye ait olan bir ya da daha çok kuyruk yöneticisinde var.

Yerel kuyruk yöneticisine bağlı uygulamalar, kuyruğun konumundan bağımsız olarak, bu tipteki kuyruklara ileti yerleştirebilirler. Kuyruğun bir eşgörünümü yerel kuyruk yöneticisinde varsa, kuyruk yerel bir kuyruk gibi davranır ve yerel kuyruk yöneticisine bağlı uygulamalar kuyruktan iletileri kaldırabilir. **QType** kuyruk özniteliğinin değeri QTCLUS.

Diğer ad kuyruğu

Bu bir fiziksel kuyruk değil-yerel bir kuyruk için alternatif bir addır. Diğer adın çözüldüğü yerel kuyruğun adı, diğer ad kuyruğunun tanımının bir parçasıdır.

Yerel kuyruk yöneticisine bağlı uygulamalar, diğer ad kuyruklarına ileti yerleştirip diğer ad kuyruklarından ileti kaldırabilir; iletiler, diğer adın çözüldüğü yerel kuyruğa yerleştirilir ve kuyruktan kaldırılır. **QType** kuyruk özniteliğinin değeri QTALS.

Uzak kuyruk

Bu bir fiziksel kuyruk değildir; uzak kuyruk yöneticisinde varolan bir kuyruğun yerel tanımıdır. Uzak kuyruğun yerel tanımlaması, yerel kuyruk yöneticisine iletilerin uzak kuyruk yöneticisine nasıl yönlendirileceğini bildiren bilgiler içerir.

Yerel kuyruk yöneticisine bağlı uygulamalar uzak kuyruklara ileti yerleştirebilirler; iletiler, iletileri uzak kuyruk yöneticisine yöneltmek için kullanılan yerel iletim kuyruğuna yerleştirilir. Uygulamalar uzak kuyruklardan ileti kaldıramaz. **QType** kuyruk özniteliğinin değeri QTREM 'dir.

Uzak kuyruk tanımlaması aşağıdakiler için de kullanılabilir:

- Yanıt kuyruğu diğer ad alma

Bu durumda, tanımlamanın adı bir yanıt kuyruğunun adıdır. Daha fazla bilgi için bkz. [Yanıt kuyruğu diğer adı tanımlamaları](#).

- Kuyruk yöneticisi diğer adını alma

Bu durumda, tanımlamanın adı bir kuyruk yöneticisinin adı değil, diğer adıdır. Daha fazla bilgi için bkz. [Kuyruk yöneticisi diğer adı tanımlamaları](#).

Model kuyruğu

Bu bir fiziksel kuyruk değildir; yerel bir kuyruğun yaratılabileceği bir kuyruk öznitelikleri kümeidir.

İletiler bu tipteki kuyruklarda saklanamaz.

Bazı kuyruk öznitelikleri tüm kuyruk tipleri için geçerlidir; diğer kuyruk öznitelikleri yalnızca belirli kuyruk tipleri için geçerlidir. Bir özneliğin uygulandığı kuyruk tipleri, [Çizelge 754 sayfa 1346](#) ve sonraki tablolarda "X" ile gösterilir.

[Çizelge 754 sayfa 1346](#) , kuyruklara özgü öznitelikleri özetler. Öznitelikler alfabetik sırayla açıklanır.

Çizelgede gösterilen özniteliklerin adları, MQINQ ve MQSET çağrılarıyla kullanılan adlardır. Öznitelikleri tanımlamak, değiştirmek ya da görüntülemek için MQSC komutları kullanıldığında, diğer kısa adlar kullanılır; ayrıntılar için [MQSC komutları](#) konusuna bakın.

Aşağıdaki tabloda sütunlar aşağıdaki gibi geçerlidir:

- Yerel kuyruklara ilişkin kolon, paylaşılan kuyruklara da uygulanır.
- Model kuyruklarına ilişkin kolon, model kuyruğundan yaratılan yerel kuyruk tarafından devralınan öznitelikleri gösterir.
- Küme kuyruklarına ilişkin kolon, küme kuyruğu tek başına sorgu için ya da sorgu ve çıkış için açıldığında sorulabilecek öznitelikleri gösterir. Küme kuyruğu sorgunun yanı sıra bir ya da daha fazla giriş, göz atma ya da küme için açılırsa, bunun yerine yerel kuyruklara ilişkin sütun geçerli olur.

<i>Çizelge 754. Kuyruklara ilişkin öznitelikler</i>						
Öznitelik	Açıklama	Yerel	Modeli	Diğer Ad	Uzak	Küme
AlterationDate	Tanımının en son değiştirildiği tarih	X		X	X	
AlterationTime	Tanımlamanın en son değiştirildiği zaman	X		X	X	
BackoutRequeueQName	Aşırı geriletme yeniden kuyruğa alma kuyruk adı	X	X			
BackoutThreshold	Geriletme eşiği	X	X			
BaseQName	Diğer adın çözümleneceği kuyruk adı			X		
ClusterChannelAdı	Küme-gönderen kanal adı	✓	✓			
ClusterName	Kuyruğun ait olduğu kümenin adı	X		X	X	
ClusterNameList	Kuyruğun ait olduğu kümelerin adlarını içeren ad listesi nesnesinin adı	X		X	X	
CreationDate	Kuyruğun yaratıldığı tarih	X				
CreationTime	Kuyruğun yaratıldığı saat	X				
CurrentQDepth	Yürürlükteki kuyruk derinliği	X				
DefBind	Varsayılan bağ tanımı	X		X	X	X
DefinitionType	Kuyruk tanımlaması tipi	X	X			

Çizelge 754. Kuyruklara ilişkin öznitelikler (devamı var)

Öznitelik	Açıklama	Yerel	Modeli	Diğer Ad	Uzak	Küme
DefInputOpenOption	Varsayılan giriş açma seçeneği	X	X			
DefPersistence	Varsayılan ileti kalıcılığı	X	X	X	X	X
DefPriority	Varsayılan ileti önceliği	X	X	X	X	X
DistLists	Dağıtım listesi desteği	X	X			
HardenGetGeri Çıkış	Doğru bir geriletme sayısının korunup korunmayacağını belirler	X	X			
InhibitGet	Kuyruk için alma işlemlerine izin verilip verilmediğini denetler	X	X	X		
InhibitPut	Kuyruğa ilişkin koyma işlemlerine izin verilip verilmediğini denetler	X	X	X	X	X
InitiationQName	Başlatma kuyruğunun adı	X	X			
MaxMsgUzunluğu	İleti uzunluğu üst sınırı (byte)	X	X			
MaxQDepth	Kuyruk derinliği üst sınırı	X	X			
MediaLog	En eski günlük kapsamının tanıtıcısı (ya da IBM i üzerindeki en eski günlük nesnesinin tanıtıcısı) belirtilen bir kuyruk ortamı kurtarma işlemi için gerekli	✓	✓			
MsgDeliverySırası	İleti teslim sırası	X	X			
OpenInputSayısı	Giriş için açma sayısı	X				
OpenOutputSayısı	Çıkış için açma sayısı	X				
ProcessName	İşlem adı	X	X			
QDepthHighOlayı	Kuyruk Derinliği Yüksek olaylarının oluşturulup oluşturulmayacağını denetler	X	X			
QDepthHighSınırı	Kuyruk derinliği için üst sınır	X	X			

Çizelge 754. Kuyruklara ilişkin öznitelikler (devamı var)

Öznitelik	Açıklama	Yerel	Modeli	Diğer Ad	Uzak	Küme
QDepthLowOlayı	Kuyruk Derinliği Düşük olaylarının oluşturulup oluşturulmayacağını denetler	X	X			
QDepthLowSınırı	Kuyruk derinliği için alt sınır	X	X			
QDepthMaxOlayı	Tam Kuyruk olaylarının oluşturulup oluşturulmayacağını denetler	X	X			
QDesc	Kuyruk tanımlaması	X	X	X	X	X
QName	Kuyruk adı	X		X	X	X
QServiceInterval	Kuyruk hizmeti aralığı için hedef	X	X			
QServiceIntervalOlayı	Hizmet Aralığı Yüksek ya da Hizmet Aralığı Tamam olaylarının oluşturulup oluşturulmayacağını denetler	X	X			
QType	Kuyruk tipi	X		X	X	X
RemoteQMgrAdı	Uzak kuyruk yöneticisinin adı				X	
RemoteQName	Uzak kuyruğun adı				X	
RetentionInterval	Alıkoyma aralığı	X	X			
Kapsam	Kuyruğa ilişkin bir girişin bir hücre dizininde de bulunup bulunmadığını denetler	X		X	X	
Paylaşılabilir	Kuyruk paylaşılabilir	X	X			
TriggerControl	Tetikleme denetimi	X	X			
TriggerData	Tetikleyici verileri	X	X			
TriggerDepth	Tetikleme derinliği	X	X			
TriggerMsgÖncelik	Tetikleyiciler için eşik iletisi önceliği	X	X			
TriggerType	Tetikleme Tipi	X	X			
Kullanım	Kuyruk kullanımı	X	X			
XmitQName	İletim kuyruğu adı				X	

IBM i**IBM i üzerinde AlterationDate (12 baytlık karakter dizgisi)**

Tanımının en son değiştirildiği tarih.

Çizelge 755. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Bu, tanımın en son değiştirildiği tarihtir. Tarihin biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için iki sondaki boşlukla doldurulur (örneğin, 1992-09-23--), burada -- iki boş karakteri temsil eder).

Kuyruk yöneticisi çalışırken, belirli özneliklerin değerleri (örneğin, *CurrentQDepth*) değişir. Bu özneliklerde yapılan değişiklikler *AlterationDate* özneliğini etkilemez.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla CAALTD seçicisini kullanın. Bu özneliğin uzunluğu LNDATE tarafından verilir.

IBM i**AlterationTime (8 baytlık karakter dizgisi)- IBM i**

Tanımlamanın en son değiştirildiği zaman.

Çizelge 756. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Bu, tanımın en son değiştirildiği zamandır. Saatin biçimi HH.MM.SS 24 saatlik zaman biçimini kullanır; saat 10 'dan küçükse (örneğin, 09.10.20), başında sıfır vardır. Saat yerel saattir.

Kuyruk yöneticisi çalışırken, belirli özneliklerin değerleri (örneğin, *CurrentQDepth*) değişir. Bu özneliklerde yapılan değişiklikler *AlterationTime* özneliğini etkilemez.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla CAALTT seçicisini kullanın. Bu özneliğin uzunluğu LNTIME tarafından verilir.

IBM i**BackoutRequeueQName (48 baytlık karakter dizgisi) IBM i**

Çok fazla geriletme yeniden kuyruğa alma kuyruk adı.

Çizelge 757. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

WebSphere Application Server içinde çalışan uygulamalar ve IBM MQ Application Server Facilities kullanan uygulamalar, geriletilmiş iletilerin nereye gitmesi gerektiğini belirlemek için bu özneliği kullanır. Diğer tüm uygulamalar için, kuyruk yöneticisi, değerinin sorgulanmasına izin vermenin yanı sıra, özneliğin değerine dayalı olarak herhangi bir işlem gerçekleştirmez.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla CABRQN seçicisini kullanın. Bu özneliğin uzunluğu LNQN tarafından verilir.

IBM i**IBM i üzerinde BackoutThreshold (10 basamaklı işaretli tamsayı)**

Geriletme eşiği.

Çizelge 758. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

WebSphere Application Server içinde çalışan uygulamalar ve IBM MQ Application Server Facilities kullanan uygulamalar, bir iletinin geriletilmesi gerekip gerekmediğini belirlemek için bu özniteliği kullanır. Diğer tüm uygulamalar için, kuyruk yöneticisi, değerinin sorgulanmasına izin vermenin yanı sıra, özniteliğin değerine dayalı olarak herhangi bir işlem gerçekleştirmez.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IABTHR seçicisini kullanın.

IBM i **IBM i üzerinde BaseQName (48 baytlık karakter dizgisi)**

Diğer adın çözümleneceği kuyruk adı.

Çizelge 759. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
		X		

Bu, yerel kuyruk yöneticisine tanımlanan bir kuyruğun adıdır. (Kuyruk adlarıyla ilgili ek bilgi için MQOD içindeki *ODON* alanının tanımına bakın. Kuyruk aşağıdaki tiplerden biridir:

QTLOC

Yerel kuyruk.

QTREM

Uzak kuyruğun yerel tanımlaması.

QTCLUS

Küme kuyruğu.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CABASQ seçicisini kullanın. Bu özniteliğin uzunluğu LNQN tarafından verilir.

IBM i **IBM i üzerinde BaseType (tamsayı parametre yapısı)**

Diğer adın çözümleneceği nesnenin tipi.

Çizelge 760. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
		X		

Bu öznitelik aşağıdaki değerlerden birine sahip olabilir:

OTQ

Temel nesne tipi bir kuyruk

OTTOP

Temel nesne tipi bir konudur

IBM i **CFStrucName (12 baytlık karakter dizilimi)- IBM i**

Bağlaşım olanağı yapısı adı.

Çizelge 761. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, kuyruktaki iletilerin saklandığı bağlaşım olanağı yapısının adıdır. Adın ilk karakteri A-Z aralığındadır ve diğer karakterler A-Z, 0-9 ya da boşluk aralığındadır.

Bağlaşım olanağındaki yapının tam adı, **CFStrucName** kuyruk özniteliğinin değeriyle **QSGName** kuyruk yöneticisi özniteliğinin değerinin soneklenmesiyle elde edilir.

Bu öznitelik yalnızca paylaşılan kuyruklar için geçerlidir; *QSGDisp*, *QSGDSH* değerine sahip değilse yoksayılr.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CACFSN seçicisini kullanın. Bu özniteliğin uzunluğu LNCFSN tarafından verilir.

z/OS Bu öznitelik yalnızca z/OS üzerinde desteklenir.

ClusterChannelAdı (20 baytlık karakter dizgisi)

ClusterChannelName, bu kuyruğu bir iletim kuyruğu olarak kullanan küme gönderen kanallarının sosyal adıdır. Öznitelik, hangi küme-gönderen kanallarının bu küme iletim kuyruğundan bir küme-alıcı kanalına ileti gönderdiğini belirtir.

Çizelge 762. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Varsayılan kuyruk yöneticisi yapılandırması, tek bir iletim kuyruğundan (SYSTEM.CLUSTER.TRANSMIT.QUEUE) ileti göndermek için tüm küme gönderen kanalları içindir. Varsayılan yapılanış, **DefClusterXmitQueueType** kuyruk yöneticisi özniteliği değiştirilerek değiştirilebilir. Özniteliğin varsayılan değeri SCTQ' dur. Değeri CHANNEL olarak değiştirebilirsiniz.

DefClusterXmitQueueType özniteliğini CHANNEL olarak ayarlarsanız, her küme gönderen kanalı varsayılan olarak belirli bir küme iletim kuyruğunu (SYSTEM.CLUSTER.TRANSMIT.ChannelName) kullanır.

İletim kuyruğu özniteliğini **ClusterChannelName** özniteliğini kümeyi gönderen bir kanala el ile ayarlayabilirsiniz. Kümeyi gönderen kanal tarafından bağlanan kuyruk yöneticisine yönlendirilen iletiler, küme gönderen kanalı tanımlayan iletim kuyruğunda saklanır. Bunlar varsayılan küme iletim kuyruğunda saklanmaz. **ClusterChannelName** özniteliğini boş olarak ayarlarsanız, kanal yeniden başlatıldığında kanal varsayılan küme iletim kuyruğuna geçer. Varsayılan kuyruk, kuyruk yöneticisinin değerine **DefClusterXmitQueueType** (Tip) özniteliğine bağlı olarak SYSTEM.CLUSTER.TRANSMIT.ChannelName ya da SYSTEM.CLUSTER.TRANSMIT.QUEUE şeklinde olur.

ClusterChannelName içinde yıldız işaretlerini ("*") belirterek, bir iletim kuyruğunu bir küme gönderen kanalları kümesiyle ilişkilendirebilirsiniz. Yıldız işaretleri, kanal adı dizgisinin başında, sonunda ya da ortasında herhangi bir sayıda yer olabilir. **ClusterChannelName** 20 karakterlik bir uzunlukla sınırlıdır: MQ_CHANNEL_NAME_LENGTH.

IBM i üzerinde ClusterName (48 baytlık karakter dizgisi)

Kuyruğun ait olduğu kümenin adı.

Çizelge 763. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Bu, kuyruğun ait olduğu kümenin adıdır. Kuyruk birden çok kümeye aitse *ClusterNameList*, kümeleri tanımlayan bir ad listesi nesnesinin adını belirtir ve *ClusterName* boş olur. *ClusterName* ve *ClusterNameList* ürünlerinden en az biri boş olmalıdır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CACLN seçicisini kullanın. Bu özniteliğin uzunluğu LNCLUN tarafından verilir.

IBM i ClusterNameList (48 baytlık karakter dizgisi)- IBM i

Kuyruğun ait olduğu kümelerin adlarını içeren ad listesi nesnesinin adı.

Çizelge 764. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Bu, bu kuyruğun ait olduğu kümelerin adlarını içeren bir ad listesi nesnesinin adıdır. Kuyruk tek bir kümeye aitse, ad listesi nesnesi tek bir ad içerir. Diğer bir seçenek olarak, *ClusterName* kümenin adını belirtmek için kullanılabilir; bu durumda *ClusterNameList* boş olur. *ClusterName* ve *ClusterNameList* ürünlerinden en az biri boş olmalıdır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CACLNL seçicisini kullanın. Bu özniteliğin uzunluğu LNNLN tarafından verilir.

IBM i IBM i üzerinde CreationDate (12 baytlık karakter dizgisi)

Kuyruğun yaratıldığı tarih.

Çizelge 765. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Bu, kuyruğun yaratıldığı tarihtir. Tarihin biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için iki sondaki boşlukla doldurulur (örneğin, 1992-09-23-- , burada -- iki boş karakteri temsil eder).

- IBM sistemlerinde, bir kuyruğun yaratılma tarihi, kuyruğu temsil eden temel işletim sistemi varlığından (dosya ya da kullanıcı alanı) farklı olabilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CACRTD seçicisini kullanın. Bu özniteliğin uzunluğu LNCRTD tarafından verilir.

IBM i IBM i üzerinde CreationTime (8 baytlık karakter dizgisi)

Kuyruğun yaratıldığı saat.

Çizelge 766. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Bu, kuyruğun yaratıldığı zamandır. Saatin biçimi HH.MM.SS 24 saatlik zaman biçimini kullanır; saat 10 'dan küçükse (örneğin, 09.10.20), başında sıfır vardır. Saat yerel saattir.

- IBM sistemlerinde, bir kuyruğun yaratılma zamanı, kuyruğu temsil eden temel işletim sistemi varlığından (dosya ya da kullanıcı alanı) farklı olabilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CACRTT seçicisini kullanın. Bu özniteliğin uzunluğu LNCRTT tarafından verilir.

IBM i **CurrentQDepth (On IBM i üzerinde 10 basamaklı işaretli tamsayı)**
Yürürlükteki kuyruk derinliği.

Çizelge 767. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Bu, kuyrukta bulunan iletilerin sayısıdır. MQPUT çağrısı sırasında ve MQGET çağrısının geriletme sırasında artırılır. Göz atmayan bir MQGET çağrısı sırasında ve bir MQPUT çağrısının geriletme sırasında azaltılır. Bunun sonucu, sayım, MQGET çağrısı tarafından alınmaya uygun olmasa da, henüz kesinleştirilmemiş olan, ancak bir iş birimi içinde kuyruğa konan iletileri içerir. Benzer şekilde, MQGET çağrısı kullanılarak bir iş birimi içinde alınan, ancak henüz kesinleştirilmemiş iletileri dışlar.

Sayım ayrıca, süresi sona ermiş ancak henüz atılmamış olan iletileri de içerir, ancak bu iletiler alınmaya uygun değildir. “IBM i üzerinde MQMD (İleti tanımlayıcı)” sayfa 1093’ünde açıklanan MDEXP alanına bakın.

İş birimi işleme ve iletilerin bölümlenmesi *CurrentQDepth* ' un *MaxQDepth* değerini aşmasına neden olabilir. Ancak bu, iletilerin alınabilirliğini etkilemez-kuyruktaki tüm iletiler, MQGET çağrısı kullanılarak olağan şekilde alınabilir.

Kuyruk yöneticisi çalışırken bu özniteliğin değeri dalgalandı.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IACDEP seçicisini kullanın.

IBM i **DefBind (IBM i üzerinde 10 basamaklı işaretli tamsayı)**
Varsayılan bağ tanımı.

Çizelge 768. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Bu öznitelik, MQOPEN çağrısında OOBNDQ belirtildiğinde ve kuyruk bir küme kuyruğu olduğunda kullanılan varsayılan bağ tanımdır. DefBind aşağıdaki değerlerden birine sahip olabilir:

BNDOPN

Bağ tanımı MQOPEN çağrısı tarafından düzeltildi.

BNDNOT

Bağlama düzeltilmedi.

BNDGRP

Bağ tanımı MQOPEN çağrısı tarafından düzeltilmedi, ancak bir mantıksal gruptaki tüm iletiler için MQPUT üzerinde düzeltildi.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IADBND seçicisini kullanın.

IBM i **DefinitionType (10 basamaklı işaretli tamsayı)- IBM i**
Kuyruk tanımlaması tipi.

Çizelge 769. Bu özniteliğin geçerli olduğu kuyruk tipleri

Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, kuyruğun nasıl tanımlandığını gösterir. Değer aşağıdakilerden biridir:

QDPRE

Önceden tanımlanmış kalıcı kuyruk.

Kuyruk, sistem denetimcisi tarafından yaratılan kalıcı bir kuyruktur; kuyruğu yalnızca sistem denetimcisi silebilir.

Önceden tanımlanmış kuyruklar DEFINE MQSC komutu kullanılarak yaratılır ve yalnızca DELETE MQSC komutu kullanılarak silinebilir. Model kuyruklarından önceden tanımlanmış kuyruklar yaratılamaz.

Komutlar bir işletmen tarafından ya da yetkili bir kullanıcı tarafından komut giriş kuyruğuna komut iletili göndererek verilebilir (bkz. [“IBM i üzerindeki kuyruk yöneticisine ilişkin öznitelikler” sayfa 1376](#) içinde açıklanan **CommandInputQName** özniteliğine bakın).

QDPERM

Dinamik olarak tanımlanmış kalıcı kuyruk.

Kuyruk, nesne tanımlayıcı MQOD ' de belirtilen bir model kuyruğu adıyla MQOPEN çağrısı yayınlayan bir uygulama tarafından yaratılan kalıcı bir kuyruktur. Model kuyruğu tanımı, **DefinitionType** özniteliği için QDPERM değerine sahiptir.

Bu kuyruk tipi, MQCLOSE çağrısı kullanılarak silinebilir. Daha fazla ayrıntı için bkz. [“IBM i üzerinde MQCLOSE \(Nesneyi kapat\)” sayfa 1247](#) .

Kalıcı dinamik kuyruk için **QSGDisp** özniteliğinin değeri QSGDQM 'dir.

QDTEMP

Dinamik olarak tanımlanmış geçici kuyruk.

Kuyruk, MQOD nesne tanımlayıcısında belirtilen bir model kuyruğu adıyla MQOPEN çağrısı yayınlayan bir uygulama tarafından yaratılan geçici bir kuyruktur. Model kuyruğu tanımlaması, **DefinitionType** özniteliği için QDTEMP değerine sahiptir.

Bu kuyruk tipi, yaratan uygulama tarafından kapatıldığında MQCLOSE çağrısı tarafından otomatik olarak silinir.

Geçici bir dinamik kuyruk için **QSGDisp** özniteliğinin değeri QSGDQM 'dir.

QDSHAR

Dinamik olarak tanımlanan paylaşılan kuyruk.

Kuyruk, nesne tanımlayıcı MQOD ' de belirtilen bir model kuyruğu adıyla MQOPEN çağrısı yayınlayan bir uygulama tarafından yaratılan paylaşılan bir kalıcı kuyruktur. Model kuyruğu tanımlaması, **DefinitionType** özniteliği için QDSHAR değerine sahiptir.

Bu kuyruk tipi, MQCLOSE çağrısı kullanılarak silinebilir. Daha fazla ayrıntı için bkz. [“IBM i üzerinde MQCLOSE \(Nesneyi kapat\)” sayfa 1247](#) .

Paylaşılan bir dinamik kuyruk için **QSGDisp** özniteliğinin değeri QSGDSH 'dir.

Model kuyruğu tanımlamasındaki bu öznitelik, model kuyruklarının her zaman önceden tanımlanmış olması nedeniyle, model kuyruğunun nasıl tanımlandığını göstermez. Bunun yerine, model kuyruğundaki bu özniteliğin değeri, MQOPEN çağrısı kullanılarak model kuyruğu tanımlamasından yaratılan her dinamik kuyruğun *DefinitionType* değerini saptamak için kullanılır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IADEFTE seçicisini kullanın.

IBM i

DefInputOpenOption (10 basamaklı işaretli tamsayı)- IBM i

Varsayılan giriş açma seçeneği.

Çizelge 770. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, kuyruğun giriş için açılması gereken varsayılan yoldur. Kuyruk açıldığında MQOPEN çağrısında OOINPQ seçeneği belirtildiyse geçerlidir. Bu, aşağıdaki değerlerden birine sahip olabilir:

OOINPX

Dışlayıcı erişimi olan iletileri almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Kuyruk şu anda bu ya da başka bir uygulama tarafından herhangi bir tip giriş (OOINPS ya da OOINPX) için açıksa, çağrı RC2042 neden koduyla başarısız olur.

OOINPS

Paylaşılan erişimi olan iletileri almak için kuyruğu açın.

Kuyruk, sonraki MQGET çağrılarıyla kullanılmak üzere açılır. Kuyruk bu ya da OOINPS içeren başka bir uygulama tarafından açıksa, ancak kuyruk şu anda OOINPX ile açıksa RC2042 neden koduyla başarısız olursa çağrı başarılı olabilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IADINP seçicisini kullanın.

IBM i üzerinde DefPersistence (10 basamaklı işaretli tamsayı)

Varsayılan ileti kalıcılığı.

Çizelge 771. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Bu, kuyruktaki iletilerin varsayılan kalıcılığıdır. İleti konduğunda ileti tanımlayıcısında PEQDEF belirtilirse geçerlidir.

Kuyruk adı çözüm yolunda birden çok tanımlama varsa, varsayılan kalıcılık, MQPUT ya da MQPUT1 çağrısı sırasında yoldaki ilk tanımlamasında bu özniteliğin değerinden alınır. Bu şöyle olabilir:

- Diğer ad kuyruğu
- Yerel bir kuyruk
- Uzak kuyruğun yerel tanımlaması
- Kuyruk yöneticisi diğer adı
- Bir iletim kuyruğu (örneğin, *DefXmitQName* kuyruğu)

Bu, aşağıdaki değerlerden birine sahip olabilir:

PEPER

İleti kalıcı.

Bu, iletinin sistem hatalarından kurtulup kuyruk yöneticisini yeniden başlatması anlamına gelir. Kalıcı iletilerin yerleştirilemediği yer:

- Geçici dinamik kuyruklar
- Paylaşılan kuyruklar

Kalıcı iletiler kalıcı dinamik kuyruklara ve önceden tanımlanmış kuyruklara yerleştirilebilir.

PENPER

İleti kalıcı değil.

Bu, iletinin olağan durumda sistem hatalarından ya da kuyruk yöneticisini yeniden başlatmadan kurtulmadığı anlamına gelir. Bu, kuyruk yöneticisi yeniden başlatılırken yardımcı bellekte iletinin bozulmamış bir kopyası bulursa da geçerlidir.

Paylaşılan kuyrukların özel durumunda, kalıcı olmayan iletiler *yapma* kuyruk paylaşım grubundaki kuyruk yöneticilerinin yeniden başlatmalarına devam ediyor, ancak paylaşılan kuyruklardaki iletileri saklamak için kullanılan bağlaşım olanağının başarısızlıklarına dayanmıyor.

Kalıcı ve kalıcı olmayan iletiler aynı kuyruksa bulunabilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IADPER seçicisini kullanın.

IBM i DefPriority (IBM i üzerinde 10 basamaklı işaretli tamsayı)

Varsayılan ileti önceliği.

Çizelge 772. Bu özniteliğin geçerli olduğu kuyruk tipleri

Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Bu, kuyruktaki iletiler için varsayılan önceliklidir. İleti kuyruğa konduğunda ileti tanımlayıcısında PRQDEF belirtilirse bu geçerlidir.

Kuyruk adı çözme yolunda birden çok tanımlama varsa, iletinin varsayılan önceliği, koyma işlemi sırasında yoldaki *ilk* tanımlamasında bu özniteliğin değerinden alınır. Bu şöyle olabilir:

- Diğer ad kuyruğu
- Yerel bir kuyruk
- Uzak kuyruğun yerel tanımlaması
- Kuyruk yöneticisi diğer adı
- Bir iletim kuyruğu (örneğin, *DefXmitQName* kuyruğu)

Bir iletinin kuyruğa yerleştirilme şekli, kuyruğun **MsgDeliverySequence** özniteliğinin değerine bağlıdır:

- **MsgDeliverySequence** özniteliği MSPRIO ise, kuyruğa bir iletinin yerleştirildiği mantıksal konum, ileti tanımlayıcıdaki *MDPRI* alanının değerine bağlıdır.
- **MsgDeliverySequence** özniteliği MSFIFO ise, iletiler, ileti tanımlayıcısındaki *MDPRI* alanının değerine bakılmaksızın, çözülmüş kuyruğun *DefPriority* değerine eşit bir öncelikleri varmış gibi kuyruğa yerleştirilir. Ancak, *MDPRI* alanı iletiyi koyan uygulama tarafından belirtilen değeri korur. Daha fazla bilgi için "[Kuyruklara ilişkin öznitelikler](#)" sayfa 1345 içinde açıklanan **MsgDeliverySequence** özniteliğine bakın.

Öncelikler sıfır (en düşük)- *MaxPriority* (en yüksek) aralığındadır; "IBM i üzerindeki kuyruk yöneticisine ilişkin öznitelikler" sayfa 1376 içinde açıklanan **MaxPriority** özniteliğine bakın.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IADPRI seçicisini kullanın.

IBM i DefReadÖnde (10 basamaklı işaretli tamsayı) IBM i

İstemciye teslim edilen kalıcı olmayan iletiler için varsayılan önden okuma davranışını belirtir.

Çizelge 773. Bu özniteliğin geçerli olduğu kuyruk tipleri

Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X		

DefReadÖnden aşağıdaki değerlerden birine ayarlanabilir:

RAHNO.

Kalıcı olmayan iletiler, bir uygulama istemeden önce istemciye gönderilmez. İstemci olağandışı bir şekilde sona ererse, en çok bir kalıcı olmayan ileti kaybolabilir.

RAHYES.

Kalıcı olmayan iletiler, bir uygulama istemeden önce istemciye gönderilir. İstemci olağandışı bir şekilde sona ererse ya da istemci gönderildiği tüm iletileri tüketmezse, kalıcı olmayan iletiler kaybolabilir.

RAHDIS

Bu kuyruk için etkinleştirilmemiş, kalıcı olmayan iletilerin önden okunması. İletiler, istemci uygulaması tarafından önden okuma istenip istenmemesinden bağımsız olarak istemciye gönderilmez.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IADRAH seçicisini kullanın.

IBM i DefPResp (10 basamaklı işaretli tamsayı)- IBM i

Varsayılan koyma yanıtı tipi (DEFPRESP) özniteliği, MQPMO içindeki PutResponsetipi PMRASQ olarak ayarlandığında uygulamalar tarafından kullanılan değeri tanımlar. Bu öznitelik tüm kuyruk tipleri için geçerlidir.

Çizelge 774. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Bu, aşağıdaki değerlerden birine sahip olabilir:

Sync

Koyma işlemi zamanuyumlu olarak bir yanıt döndürerek yayınlanır.

Async

Koyma işlemi zamanuyumsuz olarak yayınlanır ve MQMD alanlarının bir alt kümesi döndürülüyor.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IADPRT seçicisini kullanın.

IBM i DistLists (10 basamaklı işaretli tamsayı)- IBM i

Dağıtım listesi desteği.

Çizelge 775. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, dağıtım listesi iletilerinin kuyruğa yerleştirilip yerleştirilemeyeceğini gösterir. Öznitelik, yerel kuyruk yöneticisine, kanalın diğer ucundaki kuyruk yöneticisinin dağıtım listelerini destekleyip desteklemediğini bildirmek için bir ileti kanalı aracı (MCA) tarafından ayarlanır. Bu ikinci kuyruk yöneticisi ("ortak kuyruk yöneticisi" olarak adlandırılır), iletiyi gönderen bir MCA tarafından yerel iletim kuyruğundan kaldırıldıktan sonra bir sonraki ileti alır.

Öznitelik, ortak kuyruk yöneticisinde alan MCA ile bağlantı kurduğunda gönderen MCA tarafından ayarlanır. Bu şekilde, gönderen MCA, yerel kuyruk yöneticisinin yalnızca ortak kuyruk yöneticisinin doğru olarak işleyebileceği iletileri iletim kuyruğuna yerleştirmesine neden olabilir.

Bu öznitelik öncelikle iletim kuyruklarıyla kullanılmak içindir, ancak tanımlanan işlem, kuyruk için tanımlanan kullanımdan bağımsız olarak gerçekleştirilir (**Usage** öznitelğine bakın).

Bu, aşağıdaki değerlerden birine sahip olabilir:

DLSUPP

Dağıtım listeleri desteklenir.

Bu, dağıtım listesi iletilerinin kuyrukta saklanabileceğini ve bu formda ortak kuyruk yöneticisine iletebileceğini gösterir. Bu, iletiyi birden çok hedefe göndermek için gereken işleme miktarını azaltır.

DLNSUP

Dağıtım listeleri desteklenmiyor.

Bu, dağıtım listesi iletilerinin, ortak kuyruk yöneticisi dağıtım listelerini desteklemediği için kuyrukta saklanmadığını gösterir. Bir uygulama bir dağıtım listesi iletileri koyarsa ve bu ileti bu kuyruğa yerleştirilirse, kuyruk yöneticisi dağıtım listesi iletilerini böler ve tek tek iletileri kuyruğa yerleştirir. Bu, iletiyi birden çok hedefe göndermek için gereken işleme miktarını artırır, ancak iletilerin ortak kuyruk yöneticisi tarafından doğru şekilde işlenmesini sağlar.

Bu özneteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IADIST seçicisini kullanın. Bu özneteliğin değerini değiştirmek için MQSET çağrısına bakın.

IBM i

HardenGetArka plan (10 basamaklı işaretli tamsayı) IBM i

Doğru bir geriletme sayısının korunup korunmayacağını belirler.

Çizelge 776. Bu özneteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Her ileti için, bir iş birimi içindeki bir MQGET çağrısı tarafından iletilerin alınma sayısı kadar bir sayı tutulur ve bu iş birimi daha sonra geriletildi. Bu sayı, MQGET çağrısı tamamlandıktan sonra ileti tanımlayıcısındaki MDBOC alanında kullanılabilir.

Kuyruk yöneticisi yeniden başlatıldığında ileti geriletme sayısı devam eder. Ancak, saymanın doğru olduğundan emin olmak için, bu kuyruğa ilişkin bir iş birimi içindeki bir MQGET çağrısıyla bir ileti her alındığında bilgilerin "sertleştirilmiş" olması (diske ya da diğer bir kalıcı depolama aygıtına kaydedilmesi) gerekir. Bu yapılmazsa ve kuyruk yöneticisinde MQGET çağrısının gerilemesi ile birlikte bir hata oluşursa, sayı artırılmayabilir.

Ancak, bir iş birimindeki her MQGET çağrısına ilişkin bilgilerin güçlendirilmesinde bir performans maliyeti ortaya çıkar ve **HardenGetBackout** özneteliği yalnızca saymanın doğru olması gerekirse QABH olarak ayarlanmalıdır.

- IBM üzerinde, bu özneteliğin ayarından bağımsız olarak, ileti geriletme sayısı her zaman sertleşmiştir.

Aşağıdaki değerler kullanılabilir:

QABH

Geri sayım hatırlanıyor.

Bu kuyruktaki iletilere ilişkin geriletme sayısının doğru olduğundan emin olmak için sertleme kullanılır.

QABNH

Geriletme sayısı anımsanmayabilir.

Bu kuyruktaki iletilere ilişkin geriletme sayısının doğru olduğundan emin olmak için sertleme kullanılmaz. Bu nedenle sayım olması gerekenden daha düşük olabilir.

Bu özneteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IAHGB seçicisini kullanın.

IBM i

InhibitGet (10 basamaklı işaretli tamsayı) IBM i

Bu kuyruk için alma işlemlerine izin verilip verilmeyeceğini denetler.

Çizelge 777. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X		

Kuyruk bir diğer ad kuyruğuysa, MQGET çağrısının başarılı olması için, alma işlemi sırasında hem diğer ad, hem de temel kuyruk için alma işlemlerine izin verilmelidir. Değer aşağıdakilerden biridir:

QAGETI

Alma işlemleri engellenmiş.

MQGET çağrıları RC2016neden koduyla başarısız oluyor. Bu, GMBRWF ya da GMBRWN belirten MQGET çağrılarını içerir.

Not: Bir iş birimi içinde çalışan bir MQGET çağrısı başarıyla tamamlanırsa, **InhibitGet** özniteliğinin değerinin QAGETI olarak değiştirilmesi, iş biriminin kesinleştirilmesini engellemez.

QAGETA

Alma işlemlerine izin verilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IAIGET seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

IBM i **InhibitPut (10 basamaklı işaretli tamsayı) IBM i**

Bu kuyruk için koyma işlemlerine izin verilip verilmediğini denetler.

Çizelge 778. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Kuyruk-adı çözüm yolunda birden çok tanımlama varsa, MQPUT ya da MQPUT1 çağrısının başarılı olması için, koyma işlemi sırasında yoldaki *her* tanımlaması (kuyruk yöneticisi diğer adı tanımlamaları da içinde olmak üzere) için koyma işlemlerine izin verilmelidir. Bu, aşağıdaki değerlerden birine sahip olabilir:

QAPUTI

Koyma işlemleri engellenmiş.

MQPUT ve MQPUT1 çağrıları RC2051neden koduyla başarısız oluyor.

Not: Bir iş birimi içinde çalışan bir MQPUT çağrısı başarıyla tamamlanırsa, **InhibitPut** özniteliğinin değerinin daha sonra QAPUTI olarak değiştirilmesi, iş biriminin kesinleştirilmesini engellemez.

QAPUTA

Koyma işlemlerine izin verilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IAIPUT seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

IBM i **InitiationQName (48 baytlık karakter dizgisi)- IBM i**

Başlatma kuyruğunun adı.

Çizelge 779. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, yerel kuyruk yöneticisinde tanımlanan bir kuyruğun adıdır; kuyruk QTLOC tipinde olmalıdır. Kuyruk yöneticisi, bu özniteliğin ait olduğu kuyruğa gelen bir iletinin sonucu olarak uygulama başlatma gerektiğinde başlatma kuyruğuna bir tetikleyici iletisi gönderir. Başlatma kuyruğunun, tetikleme iletisi alındıktan sonra uygun uygulamayı başlatacak bir tetikleyici izleme uygulaması tarafından izlenmesi gerekir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAINIQ seçicisini kullanın. Bu özniteliğin uzunluğu LNQN tarafından verilir.

IBM i MaxMsgUzunluğu (10 basamaklı işaretli tamsayı) IBM i
İleti uzunluğu üst sınırı (byte).

Çizelge 780. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, kuyruğa yerleştirilebilecek en uzun *fiziksel* iletinin uzunluğu için bir üst sınırdır. Ancak, **MaxMsgLength** kuyruk özniteliği **MaxMsgLength** kuyruk yöneticisi özniteliğinden bağımsız olarak ayarlanabildiğinden, kuyruğa yerleştirilebilecek en uzun fiziksel iletinin uzunluğu için gerçek üst sınır, bu iki değerden daha azdır.

Kuyruk yöneticisi bölümlenmeyi destekliyorsa, bir uygulamanın iki **MaxMsgLength** özniteliğinin daha küçük olanından daha uzun bir *mantıksal* ileti koyması mümkündür, ancak yalnızca uygulama MQMD 'de MFSEGA işaretini belirtiyorsa. Bu işaret belirtilirse, mantıksal ileti uzunluğu üst sınırı 999 999 bayttır, ancak genellikle işletim sistemi ya da uygulamanın çalıştığı ortam tarafından uygulanan kaynak kısıtlamaları alt sınırla sonuçlanır.

Çok uzun bir iletiyi kuyruğa yerleştirme girişimi başarısız oldu. Neden kodu:

- İleti kuyruk için çok büyükse RC2030
- RC2031 İleti kuyruk yöneticisi için çok büyükse, ancak kuyruk için çok büyük değilse

MaxMsgLength özniteliği için alt sınır sıfırdır. Üst sınır ortam tarafından belirlenir:

- IBM i' de ileti uzunluğu üst sınırı 100 MB (104 857 600 bayt).

Daha fazla bilgi için, “IBM i üzerinde MQPUT (Put message)” sayfa 1310’ünde açıklanan **BUFLN** parametresine bakın.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IAMLEN seçicisini kullanın.

IBM i MaxQDepth (10 basamaklı işaretli tamsayı)- IBM i
Kuyruk derinliği üst sınırı.

Çizelge 781. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, herhangi bir zamanda kuyrukta bulunabilecek fiziksel ileti sayısı için tanımlanan üst sınırdır. **MaxQDepth** ileti içeren bir kuyruğa ileti koyma girişimi, RC2053neden koduyla başarısız oldu.

İş birimi işleme ve iletilerin bölümlenmesi, kuyruktaki gerçek fiziksel ileti sayısının **MaxQDepth** değerini aşmasına neden olabilir. Ancak bu, iletilerin alınabilirliğini etkilemez-kuyruktaki *tüm* iletiler, MQGET çağrısı kullanılarak olağan şekilde alınabilir.

Bu özniteliğin değeri sıfır ya da daha büyük. Üst sınır ortam tarafından belirlenir.

Not: Kuyrukta *MaxQDepth* iletiden daha az ileti olsa da, kuyruğun kullanabileceği saklama alanı tükenebilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IAMDEP seçicisini kullanın.

IBM i

MediaLog (IBM i üzerinde 10 basamaklı işaretli tamsayı)

Günlük kapsamının (ya da IBM i üzerindeki günlük nesnesinin) tanıtıcısı belirli bir kuyruğun ortam kurtarma işlemi için gereklidir.

Çizelge 782. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Çevrimsel günlük kaydının kullanıldığı kuyruk yöneticilerindeki değer boş bir dizgi olarak döndürülür.

IBM i

MsgDeliverySırası (10 basamaklı işaretli tamsayı) IBM i

İleti teslim sırası.

Çizelge 783. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, iletilerin MQGET çağrısı tarafından uygulamaya döndürüleceği sırayı belirler:

MSFIFO

İletiler FIFO sırasıyla döndürülür (ilk giren ilk çıkar).

Bu, bir MQGET çağrısının, iletinin önceliğine bakılmaksızın, çağrıda belirtilen seçim ölçütlerini karşılayan *ilk* iletiyi döndüreceği anlamına gelir.

MSPRIO

İletiler öncelik sırasına göre döndürülür.

Bu, bir MQGET çağrısıyla, çağrıda belirtilen seçim ölçütlerini karşılayan *en yüksek öncelik* iletisinin döndürüleceği anlamına gelir. Her öncelik düzeyinde, iletiler FIFO sırasıyla döndürülür (ilk giren ilk çıkar).

Kuyrukta iletiler varken ilgili öznitelikler değiştirilirse, teslim sırası aşağıdaki gibidir:

- İletilerin MQGET çağrısı tarafından döndürülme sırası, ileti kuyruğa geldiğinde kuyruk için yürürlükte olan **MsgDeliverySequence** ve **DefPriority** özniteliklerinin değerlerine göre belirlenir:
 - İleti geldiğinde *MsgDeliverySequence* MSFIFO ise, ileti, önceliği *DefPriority* gibi kuyruğa yerleştirilir. Bu, iletinin ileti tanımlayıcısındaki *MDPRI* alanının değerini etkilemez; bu alan, ileti ilk bulunduğu anda sahip olduğu değeri korur.
 - İleti geldiğinde *MsgDeliverySequence* MSPRIO ise, ileti, ileti tanımlayıcıda *MDPRI* alanı tarafından belirtilen önceliğe uygun olan yerde kuyruğa yerleştirilir.

Kuyruktaki iletiler varken **MsgDeliverySequence** özniteliğinin değeri değiştirilirse, kuyruktaki iletilerin sırası değişmez.

Kuyrukta iletiler varken **DefPriority** özniteliğinin değeri değiştirilirse, **MsgDeliverySequence** özniteliği MSFIFO olarak ayarlanmış olsa da, iletilerin FIFO sırasıyla teslim edilmesi gerekmez; kuyruğa daha yüksek öncelikte konanlar önce teslim edilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IAMDS seçicisini kullanın.

IBM i**OpenInputSayı (10 basamaklı işaretli tamsayı) IBM i**

Giriş için açma sayısı.

Çizelge 784. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Bu, MQGET çağrısıyla kuyruktan ileti kaldırmak için geçerli olan tanıtıcı sayısıdır. Bu, *yerel* kuyruk yöneticisi tarafından bilinen bu tür tanıtıcıların toplam sayısıdır. Kuyruk paylaşılan bir kuyruksa, bu sayı, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubundaki diğer kuyruk yöneticilerindeki kuyruk için gerçekleştirilen giriş açılarını içermez.

Sayı, bu kuyruğa çözülen diğer ad kuyruğunun giriş için açıldığı tanıtıcıları içerir. Sayı, giriş içermeyen işlemler için kuyruğun açıldığı noktaları içermez (örneğin, yalnızca göz atmak için açılan bir kuyruk).

Kuyruk yöneticisi çalışırken bu özneliğin değeri dalgalandı.

Bu özneliğin değerini belirlemek için MQINQ çağrısıyla birlikte IAİKÖ seçicisini kullanın.

IBM i**OpenOutputSayı (10 basamaklı işaretli tamsayı) IBM i**

Çıkış için açma sayısı.

Çizelge 785. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X				

Bu, MQPUT çağrısıyla kuyruğa ileti eklemek için geçerli olan tanıtıcı sayısıdır. Bu, *yerel* kuyruk yöneticisi tarafından bilinen bu tür tanıtıcıların toplam sayısıdır; uzak kuyruk yöneticisinde bu kuyruk için gerçekleştirilen çıkışlara ilişkin açma işlemlerini içermez. Kuyruk paylaşılan bir kuyruksa, bu sayı, yerel kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubundaki diğer kuyruk yöneticilerindeki kuyruk için gerçekleştirilen çıkışa ilişkin açılarını içermez.

Sayı, bu kuyruğa çözülen bir diğer ad kuyruğunun çıkış için açıldığı tanıtıcıları içerir. Bu sayı, çıkış içermeyen işlemler (örneğin, yalnızca sorgu için açılan bir kuyruk) için kuyruğun açıldığı tanıtıcıları içermez.

Kuyruk yöneticisi çalışırken bu özneliğin değeri dalgalandı.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla IAOC seçicisini kullanın.

IBM i**IBM i üzerinde ProcessName (48 baytlık karakter dizgisi)**

Süreç adı.

Çizelge 786. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, yerel kuyruk yöneticisinde tanımlanan bir süreç nesnesinin adıdır. Süreç nesnesi, kuyruğa hizmet verebilen bir programı tanıtır.

Bu özneliğin değerini belirlemek için MQINQ çağrısıyla CAPRON seçicisini kullanın. Bu özneliğin uzunluğu LNPRON tarafından verilir.

QDepthHighOlay (10 basamaklı işaretli tamsayı) IBM i

Kuyruk Derinliği Yüksek olaylarının oluşturulup oluşturulmayacağını denetler.

Çizelge 787. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Kuyruk Derinliği Yüksek olayı, bir uygulamanın kuyruğa bir ileti koyduğunu ve kuyruktaki ileti sayısının kuyruk derinliği yüksek eşliğinden büyük ya da ona eşit olmasına neden olduğunu gösterir (bkz. **QDepthHighLimit** özneliği).

Not: Bu özneliğin değeri dinamik olarak değişebilir.

QDepthHighOlayının iki değeri olabilir:

EVRDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin değerini belirlemek için MQINQ çağrısıyla birlikte IAQDHE seçicisini kullanın.

QDepthHighÜst sınır (10 basamaklı işaretli tamsayı) IBM i

Kuyruk derinliği için üst sınır.

Çizelge 788. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, bir Kuyruk Derinliği Yüksek olayı oluşturmak için kuyruk derinliğinin karşılaştırıldığı eşiktir. Bu olay, bir uygulamanın kuyruğa bir ileti koyduğunu ve bu, kuyruktaki ileti sayısının kuyruk derinliği yüksek eşliğinden büyük ya da ona eşit olmasına neden olduğunu gösterir. **QDepthHighEvent** özneliğine bakın.

Değer, kuyruk derinliği üst sınırı (**MaxQDepth** özneliği) yüzdesi olarak ifade edilir ve sıfır ile 100 aralığındadır. Varsayılan değer 80 'dir.

Bu özneliğin değerini belirlemek için MQINQ aramasıyla birlikte IAQDHL seçicisini kullanın.

QDepthLowOlay (10 basamaklı imzalı tamsayı) IBM i

Kuyruk Derinliği Düşük olaylarının oluşturulup oluşturulmayacağını denetler.

Çizelge 789. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Kuyruk Derinliği Düşük olayı, bir uygulamanın kuyruktan bir ileti aldığını ve kuyruktaki ileti sayısının kuyruk derinliği alt eşliğinden az ya da ona eşit olmasına neden olduğunu gösterir (bkz. **QDepthLowLimit** özneliği).

Not: Bu özneliğin değeri dinamik olarak değişebilir.

QDepthLowOlayı aşağıdaki değerlerden birine sahip olabilir:

EVARDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin değerini saptamak için, MQINQ aramasıyla IAQDLE seçicisini kullanın.

IBM i QDepthLowÜst sınır (10 basamaklı imzalı tamsayı) IBM i
Kuyruk derinliği için alt sınır.

Çizelge 790. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, bir Kuyruk Derinliği Düşük olayı oluşturmak için kuyruk derinliğinin karşılaştırıldığı eşiktir. Bu olay, bir uygulamanın kuyruktan bir ileti aldığı ve bu, kuyruktaki ileti sayısının kuyruk derinliği alt eşliğinden daha az ya da ona eşit olmasına neden olduğunu gösterir. **QDepthLowEvent** özneliğine bakın.

Değer, kuyruk derinliği üst sınırı (**MaxQDepth** özneliği) yüzdesi olarak ifade edilir ve sıfır ile 100 aralığındadır. Varsayılan değer 20'dir.

Bu özneliğin değerini saptamak için, MQINQ aramasıyla IAQDLL seçicisini kullanın.

IBM i QDepthMaxOlay (10 basamaklı işaretli tamsayı)- IBM i
Tam Kuyruk olaylarının oluşturulup oluşturulmayacağını denetler.

Çizelge 791. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Tam Kuyruk olayı, kuyruk dolu olduğu için bir kuyruğa koyma işleminin reddedildiğini, yani kuyruk derinliğinin zaten üst sınır değerine ulaştığını gösterir.

Not: Bu özneliğin değeri dinamik olarak değişebilir.

Bu, aşağıdaki değerlerden birine sahip olabilir:

EVARDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özneliğin değerini saptamak için, MQINQ aramasıyla IAQDME seçicisini kullanın.

IBM i IBM i üzerinde QDesc (64 baytlık karakter dizgisi)
Kuyruk tanımlaması.

Çizelge 792. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X	X	X	X

Bu, açıklayıcı açıklama için kullanılabilir bir alandır. Alanın içeriği kuyruk yöneticisi için önemli değildir, ancak kuyruk yöneticisi alanın yalnızca görüntülenebilecek karakterleri içermesini gerektirebilir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 bayttır).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özniteliğinde tanımlandığı gibi) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAQD seçicisini kullanın. Bu özniteliğin uzunluğu LNQD tarafından verilir.

IBM i **IBM i üzerinde QName (48 baytlık karakter dizisi)**

Kuyruk adı.

Çizelge 793. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Bu, yerel kuyruk yöneticisinde tanımlı bir kuyruğun adıdır. Kuyruk adlarıyla ilgili ek bilgi için **IBM MQ nesnelerini adlandırma kuralları** başlıklı konuya bakın. Bir kuyruk yöneticisinde tanımlanan tüm kuyruklar aynı kuyruk ad alanını paylaşır. Bu nedenle, QTLOC kuyruğu ve QTALS kuyruğu aynı ada sahip olamaz.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAQN seçicisini kullanın. Bu özniteliğin uzunluğu LNQN tarafından verilir.

IBM i **QServiceInterval (10 basamaklı işaretli tamsayı)- IBM i**

Kuyruk hizmeti aralığı için hedef.

Çizelge 794. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, Hizmet Aralığı Yüksek ve Hizmet Aralığı Tamam olaylarını oluşturmak için kullanılan hizmet aralığıdır. **QServiceIntervalEvent** özniteliğine bakın.

Değer milisaniye cinsinden ve sıfır ile 999 999 999 aralığındadır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IAQSI seçicisini kullanın.

IBM i **QServiceIntervalOlay (10 basamaklı işaretli tamsayı)- IBM i**

Hizmet Aralığı Yüksek ya da Hizmet Aralığı Tamam olaylarının oluşturulup oluşturulmayacağını denetler.

Çizelge 795. Bu özniteliğin geçerli olduğu kuyruk tipleri

Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

- Bir denetim, kuyruktan en az **QServiceInterval** özniteliğiyle belirtilen süre boyunca herhangi bir ileti alınmadığını belirttiğinde bir Hizmet Aralığı Yüksek Olayı oluşturulur.
- Bir denetim, iletilerin **QServiceInterval** özniteliğiyle belirtilen süre içinde kuyruktan alındığını gösterdiğinde bir Hizmet Aralığı Tamam olayı oluşturulur.

Not: Bu özniteliğin değeri dinamik olarak değişebilir.

Bu öznitelik aşağıdaki değerlerden birine sahip olabilir:

QSIEHI

Kuyruk Hizmeti Aralığı Yüksek olayları etkinleştirildi.

- Kuyruk Hizmeti Aralığı Yüksek olayları **etkin** ve
- Kuyruk Hizmeti Aralığı Tamam olayları **devre dışı**.

QSIEOK

Kuyruk Hizmeti Aralığı Tamam olayları etkinleştirildi.

- Kuyruk Hizmeti Aralığı Yüksek olayları **devre dışı** ve
- Kuyruk Hizmeti Aralığı Tamam olayları **etkin**.

QSIENO

Herhangi bir kuyruk hizmeti aralığı olayı etkinleştirilmedi.

- Kuyruk Hizmeti Aralığı Yüksek olayları **devre dışı** ve
- Kuyruk Hizmeti Aralığı Tamam olayları da **devre dışı** bırakılır.

Paylaşılan kuyruklar için, bu özniteliğin değeri yoksayılır; QSIENO değeri varsayılır.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özniteliğin değerini belirlemek için MQINQ aramasıyla IAQSIE seçicisini kullanın.

IBM i üzerinde QSGDisp (10 basamaklı işaretli tamsayı)

Kuyruk paylaşım grubu düzenleme.

Çizelge 796. Bu özniteliğin geçerli olduğu kuyruk tipleri

Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Bu, kuyruğun atılacağını belirtir. Değer aşağıdakilerden biridir:

QSGDQM

Kuyruk yöneticisi düzenleme.

Nesnenin kuyruk yöneticisi yok etme işlemi var. Bu, nesne tanımlamasının yalnızca yerel kuyruk yöneticisi tarafından tanındığı; tanımın kuyruk paylaşım grubundaki diğer kuyruk yöneticileri tarafından bilinmediği anlamına gelir.

Kuyruk paylaşım grubundaki her kuyruk yöneticisinin yürürlükteki nesneyle aynı ada ve tipe sahip bir nesnesi olması olasıdır, ancak bunlar ayrı nesnelere ve bunlar arasında ilinti yoktur. Öznitelikleri birbirleriyle aynı olacak şekilde kısıtlanmaz.

QSGDCP

Kopyalanan nesne yok etme.

Nesne, paylaşılan havuzda var olan bir ana nesne tanımlamasının yerel bir kopyasıdır. Kuyruk paylaşım grubundaki her kuyruk yöneticisinin nesnenin kendi kopyası olabilir. Başlangıçta tüm kopyalar aynı özniteliklere sahiptir, ancak MQSC komutları kullanılarak her kopya, öznitelikleri diğer kopyalardan farklı olacak şekilde değiştirilebilir. Paylaşılan havuzdaki ana tanımlama değiştirildiğinde, kopyaların öznitelikleri yeniden eşitlenir.

QSGDSH

Paylaşılan yok etme.

Nesne paylaşılan yok etme içeriyor. Bu, paylaşılan havuzda, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından bilinen nesnenin tek bir eşgörünümünün bulunduğu anlamına gelir. Gruptaki bir kuyruk yöneticisi nesneye eriştiğinde, nesnenin paylaşılan tek eşgörünümüne erişir.

Bu özniteliğin değerini saptamak için, MQINQ aramasıyla IAQSGD seçicisini kullanın.

z/OS Bu öznitelik yalnızca z/OS üzerinde desteklenir.

IBM i **IBM i üzerinde QType (10 basamaklı işaretli tamsayı)**

Kuyruk tipi.

Çizelge 797. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	X

Bu öznitelik aşağıdaki değerlerden birini içeriyor:

QTALS

Diğer ad kuyruğu tanımlaması.

QTCLUS

Küme kuyruğu.

QTLOC

Yerel kuyruk.

QTREM

Uzak kuyruğun yerel tanımlaması.

Bu özniteliğin değerini saptamak için, MQINQ aramasıyla IAQTYP seçicisini kullanın.

IBM i **RemoteQMGrAd (48 baytlık karakter dizgisi) IBM i**

Uzak kuyruk yöneticisinin adı.

Çizelge 798. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
			X	

Bu, *RemoteQName* kuyruğunun tanımlandığı uzak kuyruk yöneticisinin adıdır. *RemoteQName* kuyruğunun *QSGDisp* değeri *QSGDCP* ya da *QSGDSH* ise, *RemoteQMGrName*, *RemoteQName*' un sahibi olan kuyruk paylaşım grubunun adı olabilir.

Bir uygulama uzak kuyruğun yerel tanımlamasını açarsa, *RemoteQMGrName* boş olmamalı ve yerel kuyruk yöneticisinin adı olmamalıdır. *XmitQName* boşsa, iletim kuyruğu olarak *RemoteQMGrName* ile aynı adı taşıyan yerel kuyruk kullanılır. *RemoteQMGrName* adlı bir kuyruk yoksa, **DefXmitQName** kuyruk yöneticisi özniteliğiyle tanımlanan kuyruk kullanılır.

Bu tanımlama bir kuyruk yöneticisi diğer adı için kullanılıyorsa, *RemoteQMgrName* , diğer adı kullanılmakta olan kuyruk yöneticisinin adıdır. Yerel kuyruk yöneticisinin adı olabilir. Ters durumda, açma işlemi gerçekleştiğinde *XmitQName* boşsa, *RemoteQMgrName* ile aynı ada sahip bir yerel kuyruk olmalıdır; bu kuyruk, iletim kuyruğu olarak kullanılır.

Bu tanımlama bir yanıt diğer adı için kullanılıyorsa, bu ad, *MDRM* olacak kuyruk yöneticisinin adıdır.

Not: Kuyruk tanımlaması yaratıldığında ya da değiştirildiğinde, bu öznitelik için belirtilen değer üzerinde geçerlilik denetimi gerçekleştirilmez.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla CARQMN seçicisini kullanın. Bu özneliğin uzunluğu LNQMN tarafından verilir.

IBM i **RemoteQName (48 baytlık karakter dizgisi)- IBM i**

Uzak kuyruğun adı.

Çizelge 799. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
			X	

Bu, *RemoteQMgrName* uzak kuyruk yöneticisinde bilinen kuyruğun adıdır.

Bir uygulama uzak kuyruğun yerel tanımını açarsa, *RemoteQName* açıldığında boş olmamalıdır.

Bu tanımlama bir kuyruk yöneticisi diğer adı tanımlaması için kullanılıyorsa, açma işlemi gerçekleştiğinde *RemoteQName* boş olmalıdır.

Tanım bir yanıt diğer adı için kullanılıyorsa, bu ad, *MDRQ* olacak kuyruğun adıdır.

Not: Kuyruk tanımlaması yaratıldığında ya da değiştirildiğinde, bu öznitelik için belirtilen değer üzerinde geçerlilik denetimi gerçekleştirilmez.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla birlikte CARQN seçicisini kullanın. Bu özneliğin uzunluğu LNQN tarafından verilir.

IBM i **RetentionInterval (10 basamaklı işaretli tamsayı)- IBM i**

Alıkoyma aralığı.

Çizelge 800. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, kuyruğun alıkonması gereken zamandır. Bu süre geçtikten sonra, kuyruk silinmeye uygun olur.

Saat, kuyruğun yaratıldığı tarih ve saatten itibaren saat cinsinden ölçülür. Kuyruğun yaratılma tarihi *CreationDate* içine kaydedilir ve kuyruğun yaratma saati **CreationTime** özneliğine kaydedilir.

Bu bilgiler, bir bakım uygulamasının ya da işletmenin artık gerekli olmayan kuyrukları tanımlamasını ve silmesini sağlamak için sağlanır.

Not: Kuyruk yöneticisi, bu özneliğe dayalı kuyrukları silmeyi ya da süresi dolmamış bir alıkoyma aralığına sahip kuyrukların silinmesini önlemeyi hiçbir zaman denemez; gerekli herhangi bir işlemin yapılmasına neden olmak kullanıcının sorumluluğundadır.

Kalıcı dinamik kuyrukların birikmesini önlemek için gerçekçi bir alıkoyma aralığı kullanılmalıdır (bkz. *DefinitionType*). Ancak, bu öznitelik önceden tanımlanmış kuyruklarla da kullanılabilir.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla IARINT seçicisini kullanın.

Bu kuyruğa ilişkin bir girişin bir hücre dizininde de bulunup bulunmadığını denetler.

Çizelge 801. Bu özneteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X		X	X	

Kurulabilir bir Ad hizmeti tarafından bir hücre dizini sağlanır. Bu, aşağıdaki değerlerden birine sahip olabilir:

SCOQM (Kalite Yönetimi)

Kuyruk yöneticisi kapsamı.

Kuyruk tanımlamasının kuyruk yöneticisi kapsamı var. Bu, kuyruğun tanımının, kuyruğun iyisi olan kuyruk yöneticisini aşmadığı anlamına gelir. Kuyruğun başka bir kuyruk yöneticisinden çıkışı için açılabilmesi için, sahip olan kuyruk yöneticisinin adı belirtilmelidir ya da diğer kuyruk yöneticisinin yerel bir kuyruk tanımlaması olmalıdır.

SCOCEL.

Hücre kapsamı.

Kuyruk tanımlamasının hücre kapsamı var. Bu, kuyruk tanımlamasının, hücredeki tüm kuyruk yöneticilerinin kullanabileceği bir hücre dizinine yerleştirileceği anlamına gelir. Kuyruk, yalnızca kuyruğun adı belirtilerek hücredeki kuyruk yöneticilerinden herhangi birinin çıkışı için açılabilir; kuyruğun sahibi olan kuyruk yöneticisinin adı belirtilmesi gerekmez. Ancak, kuyruk tanımlaması, yerel tanımlama öncelikli olduğu için, o adı taşıyan bir kuyruğun yerel tanımına da sahip olan hücredeki hiçbir kuyruk yöneticisi tarafından kullanılamaz.

Hücre dizini, LDAP (Lightweight Directory Access Protocol; Temel Dizin Erişimi Protokolü) gibi kurulabilir bir ad hizmeti tarafından sağlanır. IBM MQ ' in daha önce bir DCE dizinine kuyruk tanımlamaları eklemek için kullanılan DCE (Distributed Computing Environment) ad hizmetini artık desteklemediğini (artık desteklenmediğini) unutmayın.

Model ve dinamik kuyrukların hücre kapsamı olamaz.

Bu değer, yalnızca bir hücre dizinini destekleyen bir ad hizmetinin konfigürasyonu tanımlandıysa geçerlidir.

Bu özneteliğin değerini saptamak için, MQINQ çağrısıyla IASCOP seçicisini kullanın.

Bu öznetelik için destek aşağıdaki kısıtlamalara tabidir:

- IBM üzerinde öznetelik desteklenir, ancak yalnızca SCOQM geçerlidir.

Kuyruğun giriş için paylaşılıp paylaşılamayacağını belirler.

Çizelge 802. Bu özneteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Kuyruğun aynı anda birden çok kez giriş için açılıp açılmayacağını gösterir. Bu, aşağıdaki değerlerden birine sahip olabilir:

QASHR

Kuyruk paylaşılabilir.

OOINPS seçeneğiyle birden çok açılmasına izin verilir.

QANSHR

Kuyruk paylaşılabilir değil.

OOINPS seçeneğiyle bir MQOPEN çağrısı OOINPX olarak işlenir.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla IASHAR seçicisini kullanın.

IBM i **IBM i üzerinde TriggerControl (10 basamaklı işaretli tamsayı)**

Tetikleme kontrolü.

Çizelge 803. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, bir uygulamanın kuyruğa hizmet vermek üzere başlatılmasına neden olmak için tetikleyici iletilerin bir başlatma kuyruğuna yazılıp yazılmayacağını denetler. Bu, aşağıdakilerden biridir:

TCOFF

Tetikleyici iletileri gerekli değil.

Bu kuyruk için yazılacak tetikleyici ileti yok. *TriggerType* değeri bu durumda alakasız.

TCON.

Tetikleyici iletileri gerekiyor.

Tetikleyici iletiler, uygun tetikleyici olayları oluştuğunda bu kuyruk için yazılır.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla birlikte IATRGC seçicisini kullanın. Bu özneliğin değerini değiştirmek için MQSET çağrısına bakın.

IBM i **IBM i üzerinde TriggerData (64 baytlık karakter dizgisi)**

Tetikleyici veri.

Çizelge 804. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Kuyruk yöneticisinin, bu kuyruğa gelen bir ileti, başlatma kuyruğuna bir tetikleyici iletilerinin yazılmasına neden olduğunda tetikleyici iletiye eklediği serbest biçimli verilerdir.

Bu verilerin içeriği kuyruk yöneticisi için önemli değildir. Bu, başlatma kuyruğunu işleyen tetikleyici uygulaması ya da tetikleyici izleme programı tarafından başlatılan uygulama için anlamlıdır.

Karakter dizgisi boş değer içeremez. Gerekirse sağa doğru boşluklarla doldurulur.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla birlikte CATRGD seçicisini kullanın. Bu özneliğin değerini değiştirmek için MQSET çağrısına bakın. Bu özneliğin uzunluğu LNTRGD tarafından verilir.

IBM i **IBM i üzerinde TriggerDepth (10 basamaklı işaretli tamsayı)**

Tetikleme derinliği.

Çizelge 805. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, bir tetikleyici iletisi yazılmadan önce kuyrukta olması gereken *TriggerMsgPriority* ya da daha yüksek öncelikli iletilerin sayısıdır. Bu, *TriggerType* TTDPTH olarak ayarlandığında geçerlidir. *TriggerDepth* değeri bir ya da daha büyük. Aksi halde bu öznitelik kullanılmaz.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla birlikte IATRGRD seçicisini kullanın. Bu özneliğin değerini değiştirmek için MQSET çağrısına bakın.

IBM i TriggerMsgÖncelik (10 basamaklı imzalı tamsayı) IBM i

IBM MQ for IBM üzerindeki tetikleyiciler için eşik ileti önceliği.

Çizelge 806. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, iletilerin tetikleyici iletilerin oluşturulmasına katkıda bulunmadığı ileti önceliğidir (yani, kuyruk yöneticisi, bir tetikleyici iletisinin oluşturulup oluşturulmayacağını saptanırken bu iletileri yoksayar). *TriggerMsgPriority*, sıfır (en düşük)- *MaxPriority* aralığında olabilir (yüksek; bkz. "IBM i üzerindeki kuyruk yöneticisine ilişkin öznitelikler" sayfa 1376); sıfır değeri, tüm iletilerin tetikleyici iletilerin oluşturulmasına katkıda bulunmasına neden olur.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla IATRGRD seçicisini kullanın. Bu özneliğin değerini değiştirmek için MQSET çağrısına bakın.

IBM i IBM i üzerinde TriggerType (10 basamaklı işaretli tamsayı)

Tetikleyici tipi.

Çizelge 807. Bu özneliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, bu kuyruğa gelen iletilerin sonucu olarak tetikleyici iletilerin yazılacağı koşulları denetler. Değer aşağıdakilerden biridir:

TTNONE (TNONE)

Tetikleyici ileti yok.

Bu kuyruktaki iletilerin sonucu olarak tetikleyici ileti yazılmaz. Bu, *TriggerControl* ayarının TCOFF olarak ayarlanmasıyla aynı etkiye sahiptir.

TFRST

Kuyruk derinliği 0 'dan 1 'e geçtiğinde tetikleyici ileti.

Kuyruktaki *TriggerMsgPriority* ya da daha yüksek öncelikli iletilerin sayısı 0 'dan 1 'e değiştiğinde bir tetikleyici iletisi yazılır.

TTEVRY

Her ileti için tetikleyici ileti.

Kuyruğa *TriggerMsgPriority* ya da daha yüksek öncelikli bir ileti geldiğinde tetikleyici ileti yazılır.

TTDPTH

Derinlik eşiği aşıldığında tetikleyici ileti.

Kuyruktaki *TriggerMsgPriority* ya da daha yüksek öncelikli iletilerin sayısı *TriggerDepth* değerine eşit ya da değerini aşarsa bir tetikleyici iletisi yazılır. Tetikleyici iletisi yazıldıktan sonra, belirtik olarak yeniden açılıncaya kadar tetikleme için *TriggerControl* TCOFF olarak ayarlanır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IATRGT seçicisini kullanın. Bu özniteliğin değerini değiştirmek için MQSET çağrısına bakın.

IBM i **IBM i üzerinde kullanım (10 basamaklı işaretli tamsayı)**

Kuyruk kullanımı.

Çizelge 808. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
X	X			

Bu, kuyruğun ne için kullanıldığını gösterir. Değer aşağıdakilerden biridir:

USNORM

Normal kullanım.

Bu, normal uygulamaların ileti yerleştirirken ve alınırken kullandığı bir kuyruktur; kuyruk bir iletim kuyruğu değildir.

USTRAN.

İletim kuyruğu.

Bu, uzak kuyruk yöneticilerine gönderilen iletileri tutmak için kullanılan bir kuyruktur. Olağan bir uygulama uzak bir kuyruğa ileti gönderdiğinde, yerel kuyruk yöneticisi iletiyi geçici olarak uygun iletim kuyruğuna özel bir biçimde saklar. Bir ileti kanalı aracısı, iletiyi iletim kuyruğundan okur ve iletiyi uzak kuyruk yöneticisine iletir. İletim kuyruklarıyla ilgili ek bilgi için bkz. [İletim kuyrukları](#).

Yalnızca ayrıcalıklı uygulamalar, doğrudan ileti koymak için OOOOUT için bir iletim kuyruğunu açabilir. Normalde bunu yalnızca yardımcı program uygulamalarının yapması beklenir. İleti veri biçiminin doğru olmasına dikkat edilmelidir (bkz. [“IBM i üzerinde MQXQH \(İletim kuyruğu üstbilgisi\)” sayfa 1225](#)); tersi durumda, iletim işlemi sırasında hatalar oluşabilir. PM* bağlam seçeneklerinden biri belirtilmedikçe bağlam iletilmez ya da ayarlanmaz.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IAUSAG seçicisini kullanın.

IBM i **XmitQName (48 baytlık karakter dizgisi) IBM i**

İletim kuyruğu adı.

Çizelge 809. Bu özniteliğin geçerli olduğu kuyruk tipleri				
Yerel	Modeli	Diğer Ad	Uzak	Küme
			X	

Bir uzak kuyruk ya da kuyruk yöneticisi diğer adı tanımlaması açıldığında bu öznitelik boş değilse, iletiyi iletmek için kullanılacak yerel iletim kuyruğunun adını belirler.

XmitQName boşsa, iletim kuyruğu olarak *RemoteQMGrName* ile aynı ada sahip yerel kuyruk kullanılır. *RemoteQMGrName* adlı bir kuyruk yoksa, **DefXmitQName** kuyruk yöneticisi özniteliğiyle tanıtılan kuyruk kullanılır.

Tanımlama bir kuyruk yöneticisi diğer adı olarak kullanılıyorsa ve *RemoteQMGrName* yerel kuyruk yöneticisinin adıyla, bu öznitelik yoksayılr. Tanım, yanıtlanacak kuyruk diğer adı tanımlaması olarak kullanılıyorsa da yoksayılr.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAXQN seçicisini kullanın. Bu özniteliğin uzunluğu LNQN tarafından verilir.

Ad listesi öznitelikleri

Bu konu, ad melistlerine özgü öznitelikleri özetler. Öznitelikler alfabetik sırayla açıklanır.

Not: Gösterilen özniteliklerin adları, MQINQ ve MQSET çağrılılarıyla kullanılan adlardır.

Öznitelik açıklamaları

Ad listesi nesnesi aşağıdaki öznitelikleri içerir:

AlterationDate (12 baytlık karakter dizgisi)

Tanımının en son değiştirildiği tarih.

Bu, tanımın en son değiştirildiği tarihtir. Tarih biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için sondaki iki boşlukla doldurulur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAALTD seçicisini kullanın. Bu özniteliğin uzunluğu LNDATE tarafından verilir.

AlterationTime (8 baytlık karakter dizgisi)

Tanımlamanın en son değiştirildiği zaman.

Bu, tanımın en son değiştirildiği zamandır. Saatin biçimi şöyledir: HH.MM.SS.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAALTT seçicisini kullanın. Bu özniteliğin uzunluğu LNTIME tarafından verilir.

NameCount (10 basamaklı işaretli tamsayı)

Ad listesindeki ad sayısı.

Bu, sıfırdan büyük ya da sıfıra eşittir. Aşağıdaki değer tanımlandı:

NCMXNL

Ad listelerinde ad sayısı üst sınırı.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IANAMC seçicisini kullanın.

NamelistDesc (64 baytlık karakter dizgisi)

Namelist tanımı.

Bu, açıklayıcı açıklama için kullanılabilecek bir alandır; değeri, tanımlama süreci tarafından oluşturulur. Alanın içeriği kuyruk yöneticisi için önemli değildir, ancak kuyruk yöneticisi alanın yalnızca görüntülenebilecek karakterleri içermesini gerektirebilir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, bu alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 byte olabilir).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özniteliğinde tanımlandığı gibi) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CALSTD seçicisini kullanın.

Bu özniteliğin uzunluğu LNNLD tarafından verilir.

NamelistName (48 baytlık karakter dizgisi)

Namelist ismi.

Bu, yerel kuyruk yöneticisinde tanımlı bir ad listesi adıdır.

Her ad listesi, kuyruk yöneticisine ait diğer ad listelerinin adlarından farklı bir ada sahiptir, ancak farklı tiplerdeki (örneğin, kuyruklar) diğer kuyruk yöneticisi nesnelerinin adlarını yineleyebilir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CALSTN seçicisini kullanın.

Bu özniteliğin uzunluğu LNNLN tarafından verilir.

Adlar (48 baytlık karakter dizgisi x NameCount)

NameCount adlarının bir listesi.

Her ad, yerel kuyruk yöneticisine tanımlanan bir nesnenin adıdır. Nesne adlarıyla ilgili ek bilgi için [IBM MQ nesnelere adlandırma](#) başlıklı konuya bakın.

Bu özneteliğin değerini saptamak için, MQINQ çağrısıyla CANAMS seçicisini kullanın.

Listedeki her adın uzunluğu LNOBJN tarafından verilir.

IBM i IBM i üzerinde süreç tanımlamalarına ilişkin öznetelikler

Bu konu, süreç tanımlamalarına özgü öznetelikleri özetler. Öznetelikler alfabetik sırayla açıklanır.

Not: Gösterilen özneteliklerin adları, MQINQ ve MQSET çağrılarıyla kullanılan adlardır. Öznetelikleri tanımlamak, değiştirmek ya da görüntülemek için MQSC komutları kullanıldığında, diğer kısa adlar kullanılır; ayrıntılar için [MQSC komutları](#) konusuna bakın.

Öznetelik açıklamaları

Bir süreç tanımlaması nesnesi aşağıdaki öznetelikleri içerir:

AlterationDate (12 baytlık karakter dizgisi)

Tanımının en son değiştirildiği tarih.

Bu, tanımın en son değiştirildiği tarihtir. Tarih biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için sondaki iki boşlukla doldurulur.

Bu özneteliğin değerini saptamak için, MQINQ çağrısıyla CAALTD seçicisini kullanın. Bu özneteliğin uzunluğu LNDATE tarafından verilir.

AlterationTime (8 baytlık karakter dizgisi)

Tanımlamanın en son değiştirildiği zaman.

Bu, tanımın en son değiştirildiği zamandır. Saatin biçimi şöyledir: HH.MM.SS.

Bu özneteliğin değerini saptamak için, MQINQ çağrısıyla CAALTT seçicisini kullanın. Bu özneteliğin uzunluğu LNTIME tarafından verilir.

ApplId (256 baytlık karakter dizgisi)

Uygulama tanıtıcısı.

Bu, başlatılacak uygulamayı tanıtan bir karakter dizilimidir. Bu bilgiler, başlatma kuyruğundaki iletileri işleyen bir tetikleyici uygulaması tarafından kullanılır; bilgiler, tetikleme iletilerinin bir parçası olarak başlatma kuyruğuna gönderilir.

ApplId 'in anlamı, tetikleyici-izleyici uygulaması tarafından belirlenir. IBM MQ tarafından sağlanan tetikleyici, *ApplId* 'in yürütülebilir bir programın adı olmasını gerektirir.

Karakter dizgisi boş değer içeremez. Gerekirse sağa doğru boşluklarla doldurulur.

Bu özneteliğin değerini saptamak için, MQINQ çağrısıyla CAAPPI seçicisini kullanın. Bu özneteliğin uzunluğu LNPROA tarafından verilir.

ApplType (10 basamaklı işaretli tamsayı)

Uygulama tipi.

Bu, bir tetikleyici iletilerinin alınmasına yanıt olarak başlatılacak programın niteliğini tanımlar. Bu bilgiler, başlatma kuyruğundaki iletileri işleyen bir tetikleyici uygulaması tarafından kullanılır; bilgiler, tetikleme iletilerinin bir parçası olarak başlatma kuyruğuna gönderilir.

ApplType herhangi bir değere sahip olabilir. Standart tipler için aşağıdaki değerleri kullanabilirsiniz; kullanıcı tanımlı uygulama tipleri ATUFST-ATULST aralığındaki değerlerle sınırlıdır:

SAATCICS

CICS hareketi.

AT400

IBM i uygulaması.

ATUFST

Kullanıcı tanımlı uygulama tipi için en düşük değer.

ATULST

Kullanıcı tanımlı uygulama tipi için en yüksek değer.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IAAPPT seçicisini kullanın.

EnvData (128 baytlık karakter dizgisi)

Ortam verileri.

Bu, başlatılacak uygulamayla ilgili ortamla ilgili bilgileri içeren bir karakter dizilimidir. Bu bilgiler, başlatma kuyruğundaki iletileri işleyen bir tetikleyici uygulaması tarafından kullanılır; bilgiler, tetikleme iletilerinin bir parçası olarak başlatma kuyruğuna gönderilir.

EnvData ' in anlamı, tetikleyici-izleyici uygulaması tarafından belirlenir. IBM MQ tarafından sağlanan tetikleyici izleme programı, başlatılan uygulamaya geçirilen parametre listesinin sonuna *EnvData* eklenir. Değiştirge listesi, MQTMC2 yapısından oluşur. Ardından bir boşluk ve ardından sondaki boşluklar kaldırılmış *EnvData* gelir.

Karakter dizgisi boş değer içeremez. Gerekirse sağa doğru boşluklarla doldurulur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAENVD seçicisini kullanın. Bu özniteliğin uzunluğu LNPROE tarafından verilir.

ProcessDesc (64 baytlık karakter dizgisi)

Süreç açıklaması.

Bu, açıklayıcı açıklama için kullanılacak bir alandır. Alanın içeriği kuyruk yöneticisi için önemli değil, ancak kuyruk yöneticisi alanın yalnızca görüntülenebilecek karakterleri içermesini gerektirebilir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 bayttır).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özniteliğinde tanımlandığı gibi) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

Bu özniteliğin değerini belirlemek için MQINQ çağrısıyla CAPROD seçicisini kullanın.

Bu özniteliğin uzunluğu LNPROD tarafından verilir.

ProcessName (48 baytlık karakter dizgisi)

Süreç adı.

Bu, yerel kuyruk yöneticisinde tanımlanan bir süreç tanımlamasının adıdır.

Her süreç tanımlamasının adı, kuyruk yöneticisine ilişkin diğer süreç tanımlamalarının adlarından farklıdır. Ancak, süreç tanımlamasının adı, farklı tiplerdeki (örneğin, kuyruklar) diğer kuyruk yöneticisi nesnelere aynı olabilir.

Bu özniteliğin değerini belirlemek için MQINQ çağrısıyla CAPRON seçicisini kullanın.

Bu özniteliğin uzunluğu LNPRON tarafından verilir.

UserData (128 baytlık karakter dizgisi)

Kullanıcı verileri.

Bu, başlatılacak uygulamaya ilişkin kullanıcı bilgilerini içeren bir karakter dizilimidir. Bu bilgiler, başlatma kuyruğundaki iletileri ya da tetikleyici izleme programı tarafından başlatılan uygulamayı işleyen bir tetikleyici uygulaması tarafından kullanılır. Bilgiler, tetikleme iletilerinin bir parçası olarak başlatma kuyruğuna gönderilir.

UserData ' in anlamı, tetikleyici-izleyici uygulaması tarafından belirlenir. IBM MQ tarafından sağlanan tetikleyici izleme programı, parametre listesinin bir parçası olarak başlatılan uygulamaya *UserData* iletir. Değiştirge listesi, MQTMC2 yapısından (*UserData* içeren) oluşur; bunu izleyen bir boşluk ve ardından boşluk kaldırılmış *EnvData* izler.

Karakter dizgisi boş değer içeremez. Gerekirse sağa doğru boşluklarla doldurulur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAUSRD seçicisini kullanın. Bu özniteliğin uzunluğu LNPROU tarafından verilir.

IBM i IBM i üzerindeki kuyruk yöneticisine ilişkin öznitelikler

Kuyruk yöneticisi özniteliklerinin özeti.

Bazı kuyruk yöneticisi öznitelikleri belirli somutlamalar için düzeltilirken, diğerleri ALTER QMGRMQSC komutu kullanılarak değiştirilebilir. Öznitelikler, DISPLAY QMGRkomutu kullanılarak da görüntülenebilir. Çoğu kuyruk yöneticisi özniteliği, özel bir OTQM nesnesi açılarak ve döndürülen tanıtıcı ile MQINQ çağrısı kullanılarak sorgulanır.

Aşağıdaki çizelge, kuyruk yöneticisine özgü öznitelikleri özetler. Öznitelikler alfabetik sırayla açıklanır.

Not: Bu kısımda gösterilen özniteliklerin adları, MQINQ ve MQSET çağrılarıyla kullanılan adlardır. Öznitelikleri tanımlamak, değiştirmek ya da görüntülemek için MQSC komutları kullanıldığında, diğer kısa adlar kullanılır; ek bilgi için [MQSC komutları](#) konusuna bakın.

Öznitelik	Açıklama
AlterationDate	Tanımının en son değiştirildiği tarih
AlterationTime	Tanımlamanın en son değiştirildiği zaman
AuthorityEvent	Yetkilendirme (Yetkili Değil) olaylarının oluşturulup oluşturulmayacağını denetler
BridgeEvent	IMS köprü olaylarının oluşturulup oluşturulmadığını denetler
ChannelAutoDef	Otomatik kanal tanımına izin verilip verilmediğini denetler
ChannelAutoDefEvent	Kanal otomatik tanımlama olaylarının oluşturulup oluşturulmayacağını denetler
ChannelAutoDefExit	Otomatik kanal tanımı için kullanıcı çıkışının adı
ChannelEvent	Kanal olaylarının oluşturulup oluşturulmayacağını denetler
ClusterCacheTip	Küme önbelleğinin boyut olarak sabitlenip düzeltilmeyeceğini ya da dinamik olarak boyutlandırılıp boyutlandırılmayacağını denetler
ClusterWorkloadVerileri	Küme iş yükü çıkışına ilişkin kullanıcı verileri
ClusterWorkloadÇık	Küme iş yükü yönetimi için kullanıcı çıkışının adı
ClusterWorkloadUzunluğu	Küme iş yükü çıkışına geçirilen ileti verileri uzunluğu üst sınırı
CodedCharSetId	Kodlanmış karakter takımı tanıtıcısı
CommandEvent	Komut olayı iletilerinin kuyruğa alınıp alınmayacağını denetler
CommandInputQName	Komut giriş kuyruğu adı
CommandLevel	Komut düzeyi
ConfigurationEvent	Yapılandırma olayı
DeadLetterQName	Gönderilmeyen iletiler kuyruğunun adı

<i>Çizelge 810. Kuyruk yöneticisine ilişkin öznitelikler (devamı var)</i>	
Öznitelik	Açıklama
<u>DefClusterXmitQueueTip</u>	Varsayılan küme iletim kuyruğu tipi
<u>DefXmitQName</u>	Varsayılan iletim kuyruğu adı
<u>DistLists</u>	Dağıtım listesi desteği
<u>InhibitEvent</u>	Engelleme (Girmeyi Engelleme ve Engelleme) olaylarının oluşturulup oluşturulmayacağını denetler
<u>LocalEvent</u>	Yerel hata olaylarının oluşturulup oluşturulmayacağını denetler
<u>LoggerEvent</u>	Kurtarma günlüğü olaylarının oluşturulup oluşturulmayacağını denetler
<u>MaxHandles</u>	Tanıtıcı sayısı üst sınırı
<u>MaxMsgUzunluğu</u>	İleti uzunluğu üst sınırı (byte)
<u>MaxPriority</u>	Öncelik üst sınırı
<u>MaxUncommittedİletiler</u>	Bir iş birimindeki kesinleştirilmemiş ileti sayısı üst sınırı
<u>PerformanceEvent</u>	Performansla ilgili olayların oluşturulup oluşturulmayacağını denetler
<u>Hizmet olarak sunulan</u>	Kuyruk yöneticisinin çalıştığı platform
<u>PubSubKipi</u>	Yayınlama/abone olma motorunun ve kuyruğa alınan yayınlama/abone olma arabiriminin çalışıp çalışmadığını belirler
<u>QMgrDesc</u>	Kuyruk yöneticisi açıklaması
<u>QMgrIdentifier</u>	Kuyruk yöneticisinin dahili olarak oluşturulan benzersiz tanıtıcısı
<u>QMgrName</u>	Kuyruk yöneticisi adı
<u>RemoteEvent</u>	Uzak hata olaylarının oluşturulup oluşturulmayacağını denetler
<u>RepositoryName</u>	Bu kuyruk yöneticisinin havuz hizmetleri sağladığı kümenin adı
<u>RepositoryNamelist</u>	Bu kuyruk yöneticisinin havuz hizmetleri sağladığı kümelerin adlarını içeren ad listesi nesnesinin adı
<u>SSLCRLNamelist</u>	Kimlik doğrulama bilgisi nesnelere adlarını içeren ad listesi nesnesinin adı (Bkz. Not 1)
<u>SSLEvent</u>	TLS olaylarının oluşturulup oluşturulmadığını denetler
<u>SSLKeyRepository</u>	TLS anahtar havuzunun konumu (Bkz. Not 1)
<u>SSLKeyResetSayı</u>	Şifreleme anahtarı yeniden anlaşılmadan önce TLS etkileşimi içinde gönderilen ve alınan şifrelenmemiş bayt sayısını belirler
<u>StartStopOlayı</u>	Başlatma ve durdurma olaylarının oluşturulup oluşturulmayacağını denetler
<u>SyncPoint</u>	Eşitleme noktası kullanılabilirliği
<u>TraceRouteKaydı</u>	İletilere ilişkin izleme rotası bilgilerinin kaydedilmesini denetler
<u>TreeLifeSüresi</u>	Yönetimle ilgili olmayan konuların geçerlik süresi (saniye olarak)
<u>TriggerInterval</u>	Tetikleyici-ileti aralığı

Çizelge 810. Kuyruk yöneticisine ilişkin öznitelikler (devamı var)

Öznitelik	Açıklama
Notlar:	
1. Bu öznitelik MQINQ çağrısı kullanılarak sorulamaz ve bu kısımda tanımlanmaz. Bu öznitelikle ilgili daha fazla bilgi için bkz. Kuyruk Yöneticisini Değiştir .	

IBM i IBM i üzerinde AlterationDate (12 baytlık karakter dizgisi)

Tanımının en son değiştirildiği tarih.

Bu, tanımın en son değiştirildiği tarihtir. Tarih biçimi şöyledir: YYYY-MM-DD, 12 bayt uzunluğunu yapmak için sondaki iki boşlukla doldurulur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAALTD seçicisini kullanın. Bu özniteliğin uzunluğu LNDATE tarafından verilir.

IBM i AlterationTime (8 baytlık karakter dizgisi)- IBM i

Tanımlamanın en son değiştirildiği zaman.

Bu, tanımın en son değiştirildiği zamandır. Saatin biçimi şöyledir: HH.MM.SS.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAALTT seçicisini kullanın. Bu özniteliğin uzunluğu LNTIME tarafından verilir.

IBM i AuthorityEvent (IBM i üzerinde 10 basamaklı işaretli tamsayı)

Yetkilendirme (Yetkili Değil) olaylarının oluşturulup oluşturulmayacağını denetler.

AuthorityEvent özniteliği aşağıdaki değerlerden birine ayarlanmalıdır:

EVRDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IAAUTE seçicisini kullanın.

IBM i IBM i üzerinde BridgeEvent (karakter dizgisi)

Bu öznitelik, IMS köprüsü olay iletilerinin SYSTEM.ADMIN.CHANNEL.EVENT kuyruğu. Yalnızca z/OS üzerinde desteklenir.

IBM i ChannelAutoOnda (10 basamaklı işaretli tamsayı) IBM i

Otomatik kanal tanımına izin verip verilmediğini denetler.

Bu öznitelik, CTCRVR ve CTSVCN tipi kanalların otomatik tanımlanmasını denetler. CTCLSD kanallarının otomatik tanımının her zaman etkin olduğunu unutmayın. Bu, aşağıdaki değerlerden birine sahip olabilir:

CHADDI.

Kanal otomatik tanımlaması geçersiz kılındı.

CHADEN.

Kanal otomatik tanımlaması etkinleştirildi.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IACAD seçicisini kullanın.

IBM i ChannelAutoDefEvent (IBM i üzerinde 10 basamaklı işaretli tamsayı)

Kanal otomatik tanımlama olaylarının oluşturulup oluşturulmayacağını denetler.

Bu, CTCVR, CTSVCN ve CTCLSD tipi kanallar için geçerlidir. Bu, aşağıdaki değerlerden birine sahip olabilir:

EVRDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [İzleme ve performans](#).

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IACADE seçicisini kullanın.

IBM i ChannelAutoDefExit (20 baytlık karakter dizgisi) IBM i

Otomatik kanal tanımlaması için kullanıcı çıkışının adı.

Bu ad boş değilse ve *ChannelAutoDef* değeri CHADEN ise, kuyruk yöneticisi bir kanal tanımlaması oluşturmak üzere olduğunda çıkış çağrılır. Bu, CTCVR, CTSVCN ve CTCLSD tipi kanallar için geçerlidir. Çıkış aşağıdakilerden birini yapabilir:

- Kanal tanımının oluşturulmasına değişiklik yapılmadan devam edilmesine izin verin.
- Oluşturulan kanal tanımının özniteliklerini değiştirin.
- Kanalın oluşturulmasını tamamen engelle.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CACADX seçicisini kullanın. Bu özniteliğin uzunluğu LNEXT tarafından verilir.

IBM i ChannelEvent (karakter dizgisi)- IBM i

Kanal olayı iletilerinin oluşturulup oluşturulmayacağıni belirler.

Bu öznitelik, kanal olayı iletilerinin SYSTEM.ADMIN.CHANNEL.EVENT kuyruğu ve varsa, kuyruğa alınan ileti tipi (örneğin, 'kanal başlatıldı', 'kanal durduruldu', 'kanal etkinleştirilmedi'). Bu özniteliğin uygulanmasından önce, kanal olayı iletilerinin kuyruğa alınmasını önlemenin tek yolu hedef kuyruğu silmektir.

Bu öznitelik, yalnızca IMS köprüsü olaylarını toplamanıza da olanak sağlar (artık kanal olaylarını kapatabilirsiniz, bunlar aynı kuyruğa konmaz). Aynı durum, kanal olaylarını toplamak zorunda kalmadan da toplanabilen TLS olayları için de geçerlidir.

Bu öznitelik, yalnızca önemli olayları toplamanızı da sağlar (örneğin, kanallar olağan şekilde başlatılıp durdurulduğunda değil, hata içerdiğinde).

ChannelEvent özniteliğinin değeri aşağıdakilerden biri olabilir:

- EVREXP (yalnızca şu kanal olayları oluşturulur: RC2279, RC2283, RC2284, RC2295, RC2296).
- EVRENA (tüm kanal olayları oluşturulur; EVREXP tarafından oluşturulan olaylara ek olarak RC2282ve RC2283 olayları da oluşturulur).
- EVRDIS (kanal olayı oluşturulmaz; bu, kuyruk yöneticisinin ilk varsayılan değeridir).

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IACHNE seçicisini kullanın.

IBM i ClusterCacheTip (32 baytlık karakter dizgisi)- IBM i

Küme önbelleğinin sabit boyutta olup olmadığını ya da dinamik olarak boyutlandırılıp boyutlandırılmadığını denetler.

Bu, çağrıldığında küme iş yükü çıkışına geçirilen kullanıcı tanımlı 32 baytlık bir karakter dizgisidir. Çıkışa geçirilemeyecek veri yoksa, dizgi boş olur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CACLWD seçicisini kullanın.

IBM i ClusterWorkloadData (32 baytlık karakter dizgisi) on IBM i

Küme iş yükü çıkışına ilişkin kullanıcı verileri.

Bu, çağrıldığında küme iş yükü çıkışına geçirilen kullanıcı tanımlı 32 baytlık bir karakter dizisidir. Çıkışa geçirilemeyecek veri yoksa, dizgi boş olur.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla CACLWD seçicisini kullanın.

IBM i ClusterWorkloadExit (20 baytlık karakter dizisi) on IBM i

Küme iş yükü yönetimi için kullanıcı çıkışının adı.

Bu ad boş değilse, bir iletinin bir küme kuyruğuna konması ya da bir küme gönderen kuyruğundan diğerine taşınması durumunda çıkış çağrılır. Çıkış, kuyruk yöneticisi tarafından seçilen kuyruk eşgörünümünü iletinin hedefi olarak kabul edebilir ya da başka bir kuyruk eşgörünümü seçebilir.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla CACLWX seçicisini kullanın. Bu özneliğin uzunluğu LNEXTN tarafından verilir.

IBM i ClusterWorkloadLength (10 basamaklı işaretli tamsayı) on IBM i

Küme iş yükü çıkışına geçirilen ileti verileri uzunluğu üst sınırı.

Küme iş yükü çıkışına geçirilen ileti verileri uzunluğu üst sınırıdır. Çıkışa geçirilen verilerin gerçek uzunluğu, aşağıdakilerin en alt sınırıdır:

- İletinin uzunluğu.
- Kuyruk yöneticisinin **MaxMsgLength** özneliği.
- **ClusterWorkloadLength** özneliği.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla IACLWL seçicisini kullanın.

IBM i CodedCharSetId (10 basamaklı imzalı tamsayı)- IBM i

Kodlanmış karakter takımı tanıtıcısı.

Bu, kuyruk yöneticisi tarafından MQI 'da tanımlanan tüm karakter dizisi alanları için kullanılan nesne adları ve kuyruk yaratma tarihi ve saati gibi karakter kümesini tanımlar. Karakter takımı, nesne adlarında geçerli olan karakterler için tek baytlık karakterler içeren bir karakter olmalıdır. İletide taşınan uygulama verileri için geçerli değildir. Değer ortama bağlıdır:

- IBM 'i' da değer, kuyruk yöneticisi ilk yaratıldığında ortamda ayarlanan değerdir.

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla IACCSI seçicisini kullanın.

IBM i IBM i üzerinde CommandEvent (tamsayı)

Komutlar verildiğinde iletilerin yerel kuyruğa konup konmayacağını denetler.

Bu, iletilerin yeni bir olay kuyruğuna (SYSTEM.ADMIN.COMMAND.EVENT. Bu özellik, komut izleme bildirim ve sorun tanımlama için yararlıdır. CommandEvent kuyruk yöneticisi özneliği hakkında bilgi almak için, aşağıdaki değerlerden biriyle yeni öznelik seçici iacev 'i kullanın:

- EVRENA komutu olay iletileri oluşturulur ve tüm başarılı komutlar için kuyruğa yerleştirilir.
- EVND-komut olayı iletileri oluşturulur ve DISPLAY (MQSC) komutu ve Inquire (PCF) komutu dışındaki tüm başarılı komutlar için kuyruğa konur.
- EVRDIS-komut olayı iletileri oluşturulmaz ya da kuyruğa konmaz (bu, kuyruk yöneticisinin ilk varsayılan değeridir).

Bu özneliğin değerini saptamak için, MQINQ çağrısıyla CMDEV seçicisini kullanın.

IBM i CommandInputQName (48 baytlık karakter dizilimi)- IBM i

Komut giriş kuyruğu adı.

CommandInputQName, yerel kuyruk yöneticisinde tanımlanan komut giriş kuyruğunun adıdır. Yetki verildiyse, kullanıcıların komut gönderebileceği bir kuyruktur. Kuyruğun adı ortama bağlıdır:

- IBM işletim sistemi üzerinde, kuyruğun adı SYSTEM.ADMIN.COMMAND.QUEUE ve yalnızca PCF komutları gönderilebilir. Ancak, MQSC komutu CMESC tipinde bir PCF komutuna dahil edilirse, bu kuyruğa bir MQSC komutu gönderilebilir. Escape komutuna ilişkin ek bilgi için bkz. [Escape](#).

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CACMDQ seçicisini kullanın. Bu özniteliğin uzunluğu LNQN tarafından verilir.

IBM i

CommandLevel (On IBM i üzerinde 10 basamaklı işaretli tamsayı)

Komuta Seviyesi. Bu, kuyruk yöneticisi tarafından desteklenen sistem denetimi komutlarının düzeyini gösterir.

Düzye aşağıdaki değerlerden biridir:

CML800

Sistem denetim komutlarının 800. düzeyi.

Bu değer aşağıdaki uygulamalar tarafından döndürülür:

- IBM MQ for IBM i
 - Sürüm 8.0

CML900

Sistem kontrol komutlarının 900. düzeyi.

Bu değer aşağıdaki uygulamalar tarafından döndürülür:

- IBM MQ for IBM i
 - Sürüm 9.0

CML910

Sistem denetim komutlarının düzeyi 910.

Bu değer aşağıdaki uygulamalar tarafından döndürülür:

- IBM MQ for IBM i
 - Sürüm 9.1

CML920

Sistem denetim komutlarının düzeyi 920.

Bu değer aşağıdaki uygulamalar tarafından döndürülür:

- IBM MQ for IBM i
 - Sürüm 9.2

CML930

Sistem denetim komutlarının düzeyi 930.

Bu değer aşağıdaki uygulamalar tarafından döndürülür:

- IBM MQ for IBM i
 - Sürüm 9.3

CommandLevel özniteliğinin belirli bir değerine karşılık gelen sistem denetim komutları kümesi, **Platform** özniteliğinin değerine göre değişir; her ikisi de hangi sistem denetim komutlarının desteklendiğine karar vermek için kullanılmalıdır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IACMDL seçicisini kullanın.

IBM i

IBM i üzerinde ConfigurationEvent

Yapılandırma olaylarının oluşturulup oluşturulmayacağını ve SYSTEM.ADMIN.CONFIG.EVENT kuyruğu varsayılan nesnesi.

ConfigurationEvent özniteliği aşağıdaki değerlerden biri olabilir:

- EVRENA
- EVRDIS

ConfigurationEvent özniteliği EVRENA olarak ayarlanırsa ve bazı komutlar runmqsc ya da PCF tarafından başarıyla verilirse, yapılandırma olayları oluşturulur ve SYSTEM.ADMIN.CONFIG.EVENT kuyruğu. Bir değiştirme komutu ilgili nesneyi değiştirmese de, aşağıdaki komutlara ilişkin olaylar yayınlanır. Yapılandırma olaylarının oluşturulduğu ve gönderildiği komutlar şunlardır:

- DEFINE/ALTER AUTHINFO
- KANALI TANIM/DEĞİŞTİR
- DEFINE/ALTER NAMELIST
- SÜRECI TANIM/XX_ENCODE_CASE_CAPS_LOCK_OFF DEĞİŞTİR
- DEFINE/ALTER QLOCAL (geçici bir dinamik kuyruk değilse)
- DEFINE/ALTER QMODEL/QALIAS/QREMOTE
- AUTHINFO ÖĞESİNİ SIL
- BAĞLANTI SIL
- ADI SIL
- Süreci Sil
- DELETE QLOCAL (geçici bir dinamik kuyruk değilse)
- QMODEL/QALIAS/QREMOTE DEĞERİNİ SIL
- ALTER QMGR (CONFIGEV özniteliği geçersiz kılınmadıysa ve etkin olarak değiştirilmediyse)
- QMGR ' YI YENILE
- Geçici dinamik kuyruk dışında bir MQSET çağırısı.

Aşağıdaki durumlarda olaylar oluşturulmaz (etkinleştirildiyse):

- Komut ya da MQSET çağırısı başarısız oldu.
- Kuyruk yöneticisi olay iletisini olay kuyruğuna koyamıyor. Komut başarıyla tamamlanmalıdır.
- Geçici dinamik kuyruklar.
- İç öznitelik değişiklikleri doğrudan ya da örtük olarak (MQSET ya da komut tarafından değil) yapılır; bu, TRIGGER, CURDEPTH, IPPROCS, OPPROCS, QDPHIEV, QDPLOEV, QDPMAXEV, QSVCIEV ' i etkiler.
- Yapılandırma olay kuyruğu değiştirildiğinde, yenileme istendiğinde bu değişiklik için bir olay iletisi oluşturulacaktır.
- REFRESH/RESET CLUSTER ve RESUME/SUSPEND QMGR komutlarıyla değişiklikleri kümeleme.
- Kuyruk yöneticisi yaratılıyor ya da siliniyor.

IBM i DeadLetter IBM i üzerinde QName (48 baytlık karakter dizgisi)

Teslim edilmeyen (undelivered-message) kuyruğunun adı.

Bu, yerel kuyruk yöneticisinde tanımlı bir kuyruğun adıdır. İletiler doğru hedeflerine yönlendirilemezse, bu kuyruğa gönderilir.

Örneğin, aşağıdaki durumda iletiler bu kuyruğa yerleştirilir:

- Kuyruk yöneticisine, o kuyruk yöneticisinde henüz tanımlanmamış bir kuyruğa yönlendirilmiş bir ileti gelir.
- Bir kuyruk yöneticisine bir ileti gelir, ancak gönderildiği kuyruk iletiyi alamaz; bunun nedeni aşağıdakilerden biri olabilir:
 - Kuyruk dolu
 - Koyma istekleri engellenmiş
 - Gönderen düğümün kuyruğa ileti koyma yetkisi yok

Uygulamalar, iletileri gönderilemeyen iletiler kuyruğuna da yerleştirebilirler.

Rapor iletileri sıradan iletilerle aynı şekilde işlenir; rapor iletisi hedef kuyruğuna teslim edilemezse (genellikle özgün iletinin ileti tanımlayıcısındaki *MDRQ* alanı tarafından belirlenen kuyruk), rapor iletisi teslim edilmeyen iletiler (teslim edilmeyen iletiler) kuyruğuna yerleştirilir.

Not: Messages that have passed their expiry time (see the *MDEXP* field described in “IBM i üzerinde MQMD (İleti tanımlayıcı)” sayfa 1093) are **değil** transferred to this queue when they are discarded. Ancak, gönderen uygulama tarafından istenirse, bir son kullanma raporu iletisi (ROEXP) oluşturulmaya devam eder ve *MDRQ* kuyruğuna gönderilir.

Koyma isteğini yayınlayan uygulamaya, MQPUT ya da MQPUT1 çağrısının döndürdüğü neden koduyla (örneğin, koyma isteklerinin engellendiği yerel bir kuyruğa konan bir ileti) ilgili sorun zamanuyumlu olarak bildirildiğinde, iletiler teslim edilmeyen ileti kuyruğuna konmaz.

Teslim edilmeyen iletiler (teslim edilmeyen iletiler) kuyruğundaki iletilerin uygulama iletisi verilerinin başında bazen bir MQDLH yapısı vardır. Bu yapı, iletinin neden teslim edilmeyen ileti kuyruğuna yerleştirildiğini gösteren ek bilgiler içerir. Bu yapının daha fazla ayrıntısı için bkz. “MQDLH (Ölü harf üstbilgisi)- IBM i” sayfa 1049 .

Bu kuyruk, USNORM **Usage** özniteliğine sahip yerel bir kuyruk olmalıdır.

Bir teslim edilmeyen (undelivered-message) kuyruk bir kuyruk yöneticisi tarafından desteklenmiyorsa ya da kuyruk yöneticisi tanımlanmamışsa, ad tamamen boşluktur. Tüm IBM MQ kuyruk yöneticileri, teslim edilmeyen (teslim edilmeyen) bir kuyruğu destekler, ancak varsayılan olarak bu kuyruk tanımlanmaz.

Teslim edilmeyen (teslim edilmeyen) kuyruk tanımlı değilse ya da dolu ya da başka bir nedenle kullanılamaz durumdaysa, ileti kanalı aracısı tarafından iletilebilecek bir ileti iletim kuyruğunda alıkonur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CADLQ seçicisini kullanın. Bu özniteliğin uzunluğu LNQN tarafından verilir.

DefClusterXmitQueueType (10 basamaklı işaretli tamsayı)

DefClusterXmitQueueType özniteliği, iletileri küme alıcı kanallarına göndermek için küme gönderen kanalları tarafından varsayılan olarak hangi iletim kuyruğunun seçileceğini denetler.

DefClusterXmitQueueType değerleri MQCLXQ_SCTQ ya da MQCLXQ_CHANNEL' dir.

MQCLXQ_SCTQ

Tüm küme gönderen kanallar SYSTEM . CLUSTER . TRANSMIT . QUEUE' den ileti gönderir. İletim kuyruğuna yerleştirilen iletilerin *correlID* , iletinin gönderileceği küme gönderen kanalı belirler.

SCTQ , bir kuyruk yöneticisi tanımlandığında ayarlanır.

MQCLXQ_CHANNEL

Her küme gönderen kanal farklı bir iletim kuyruğundan ileti gönderir. Her iletim kuyruğu, SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUEmodel kuyruğundan kalıcı dinamik bir kuyruk olarak yaratılır.

Kuyruk yöneticisi özniteliği DefClusterXmitQueueType, CHANNEL, varsayılan yapılandırma, tek tek küme iletim kuyruklarıyla ilişkilendirilmiş küme gönderen kanalları olarak değiştirilir. İletim kuyrukları, SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUEmodel kuyruğundan yaratılan kalıcı dinamik kuyruklardır. Her iletim kuyruğu bir küme-gönderen kanalla ilişkilendirilir. Bir küme gönderen kanal bir küme iletim kuyruğuna hizmet sunarken, iletim kuyruğu tek bir kümede yalnızca bir kuyruk yöneticisine ilişkin iletiler içerir. Kümeleri, bir kümedeki her kuyruk yöneticisinin tek bir küme kuyruğu içereceği şekilde yapılandırabilirsiniz. Bu durumda, bir kuyruk yöneticisinden her küme kuyruğuna ileti trafiği iletilerden diğer kuyruklara ayrı olarak aktarılır. olarak ayarlanırsa

Değeri sorgulamak için MQINQ' i çağırın ya da bir Sorgu Kuyruğu Yöneticisi (MQCMD_INQUIRE_Q_MGR) PCF komutu gönderin, MQIA_DEF_CLUSTER_XMIT_Q_TYPE seçicisini ayarlayın. Değeri değiştirmek için, MQIA_DEF_CLUSTER_XMIT_Q_TYPE seçicisini ayarlayarak bir Change Queue Manager (MQCMD_CHANGE_Q_MGR) PCF komutu gönderin.

İlgili başvurular

[Kuyruk Yöneticisini Değiştir](#)

[Sorma Kuyruğu Yöneticisi](#)

“IBM i üzerinde MQINQ (Nesne öznitelikleri hakkında sorgu)” sayfa 1283
MQINQ çağırısı, bir nesnenin özniteliklerini içeren bir dizi tamsayı ve karakter dizgisi döndürür.

IBM i DefXmitQName (48 baytlık karakter dizgisi)- IBM i

Varsayılan iletim kuyruğu adı.

Bu, hangi iletim kuyruğunun kullanılacağına ilişkin başka bir belirti yoksa, uzak kuyruk yöneticilerine iletimde kullanılan iletim kuyruğunun adıdır.

Varsayılan iletim kuyruğu yoksa, ad tamamen boştur. Bu özniteliğin ilk değeri boş.

Bu özniteliğin değerini saptamak için, MQINQ çağırısıyla CADXQN seçicisini kullanın. Bu özniteliğin uzunluğu LNQN tarafından verilir.

IBM i DistLists (10 basamaklı işaretli tamsayı)- IBM i

Dağıtım listesi desteği.

Bu, yerel kuyruk yöneticisinin MQPUT ve MQPUT1 çağrılarında dağıtım listelerini destekleyip desteklemediğini gösterir. Bu, aşağıdaki değerlerden birine sahip olabilir:

DLSUPP

Dağıtım listeleri desteklenir.

DLNSUP

Dağıtım listeleri desteklenmiyor.

Bu özniteliğin değerini saptamak için, MQINQ çağırısıyla birlikte IADIST seçicisini kullanın.

IBM i InhibitEvent (IBM i üzerinde 10 basamaklı işaretli tamsayı)

Engelleme (Koymayı Engelle ve Engelle) olaylarının oluşturulup oluşturulmayacağını denetler.

Bu, aşağıdaki değerlerden birine sahip olabilir:

EVRDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [İzleme ve performans](#).

Bu özniteliğin değerini saptamak için, MQINQ çağırısıyla IAINHE seçicisini kullanın.

IBM i LocalEvent (IBM i üzerinde 10 basamaklı işaretli tamsayı)

Yerel hata olaylarının oluşturulup oluşturulmayacağını denetler.

Değer aşağıdakilerden biridir:

EVRDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#)

Bu özniteliğin değerini saptamak için, MQINQ çağırısıyla IALCLE seçicisini kullanın.

IBM i LoggerEvent (IBM i üzerinde 10 basamaklı işaretli tamsayı)

Kurtarma kaydedici olaylarının oluşturulup oluşturulmayacağını denetler.

Bu, aşağıdaki değerlerden birine sahip olabilir:

Etkinleştirildi

Günlüğe kaydedici olayları oluşturulur.

DEVRE DIŐI

Günlüğe kaydedici olayları oluşturulmaz. Bu, kuyruk yöneticilerinin ilk varsayılan değeridir.

Olaylarla ilgili daha fazla bilgi için bkz. [İzleme ve performans](#).

IBM i **MaxHandles (IBM i üzerinde 10 basamaklı işaretli tamsayı)**

Tanıtıcı sayısı üst sınırı.

Bu, herhangi bir görevin eşzamanlı olarak kullanabileceği açık tanıtıcı sayısı üst sınırıdır. Tek bir kuyruk için (ya da kuyruk olmayan bir nesne için) her başarılı MQOPEN çağrısı tek bir tanıtıcı kullanır. Bu tanıtıcı, nesne kapatıldığında yeniden kullanılabilir duruma gelir. Ancak, bir dağıtım listesi açıldığında, dağıtım listesindeki her kuyruğa ayrı bir tanıtıcı ayrılır; böylece, MQOPEN çağrısı dağıtım listesindeki kuyruklar kadar tanıtıcı kullanır. Bu, *MaxHandles* için uygun bir değere karar verilirken dikkate alınmalıdır.

MQPUT1 çağrısı, işleminin bir parçası olarak bir MQOPEN çağrısı gerçekleştirir; sonuç olarak, MQPUT1 , MQOPEN ' in kullanacağı sayıda tanıtıcı kullanır, ancak bu tanıtıcıları yalnızca MQPUT1 çağrısının kendisini çağırıldığı süre boyunca kullanır.

Değer, 1-999 999 999 999 aralığındadır. IBM i' da varsayılan değer 256 'dır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IAMHND seçicisini kullanın.

IBM i **MaxMsgUzunluğu (10 basamaklı işaretli tamsayı) IBM i**

İleti uzunluğu üst sınırı (byte).

Bu, kuyruk yöneticisi tarafından işlenebilecek en uzun *fiziksel* iletinin uzunluğudur. Ancak, **MaxMsgLength** kuyruk yöneticisi özniteliği **MaxMsgLength** kuyruk özniteliğinden bağımsız olarak ayarlanabildiğinden, bir kuyruğa yerleştirilebilecek en uzun fiziksel ileti bu iki değerden daha azdır.

Kuyruk yöneticisi bölümlenmeyi destekliyorsa, bir uygulamanın iki **MaxMsgLength** özniteliğinin daha küçük olanından daha uzun bir *mantıksal* ileti koyması mümkündür, ancak yalnızca uygulama MQMD ' de MFSEGA işaretini belirtiyorsa. Bu işaret belirtilirse, mantıksal ileti uzunluğu üst sınırı 999 999 999 bayttır, ancak genellikle işletim sistemi ya da uygulamanın çalıştığı ortam tarafından uygulanan kaynak kısıtlamaları alt sınırla sonuçlanır.

MaxMsgLength özniteliği için alt sınır 32 KB 'dir (32 768 bayt). IBM i' de ileti uzunluğu üst sınırı 100 MB (104 857 600 bayt).

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IAMLEN seçicisini kullanın.

IBM i **MaxPriority (IBM i üzerinde 10 basamaklı işaretli tamsayı)**

Maksimum öncelik.

Bu, kuyruk yöneticisi tarafından desteklenen ileti önceliği üst sınırıdır. Öncelikler sıfır (en düşük) ile *MaxPriority* (en yüksek) arasında değişir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IAMPRI seçicisini kullanın.

IBM i **MaxUncommittedMsgs (10 basamaklı imzalı tamsayı) on IBM i**

Bir iş birimindeki kesinleştirilmemiş ileti sayısı üst sınırı.

Bu, bir iş biriminde var olabilecek kesinleştirilmemiş ileti sayısı üst sınırıdır. Kesinleştirilmemiş iletilerin sayısı, yürürlükteki iş biriminin başlangıcından bu yana aşağıdakilerin toplamıdır:

- Uygulama tarafından PMSYP seçeneğiyle konan iletiler
- Uygulama tarafından GMSYP seçeneğiyle alınan iletiler
- PMSYP seçeneğiyle konan iletiler için kuyruk yöneticisi tarafından oluşturulan tetikleyici iletiler ve COA rapor iletileri
- GMSYP seçeneğiyle alınan iletiler için kuyruk yöneticisi tarafından oluşturulan COD rapor iletileri

Aşağıdaki iletiler kesinleştirilmemiş olarak sayılmaz:

- Uygulama tarafından bir iş biriminin dışına konan ya da alınan iletiler

- Bir iş biriminin dışına konan ya da alınan iletilerin sonucu olarak kuyruk yöneticisi tarafından oluşturulan tetikleyici iletiler ya da COA/COD rapor iletileri
- Kuyruk yöneticisi tarafından oluşturulan süre bitim raporu iletileri (süre bitim raporu iletilerine neden olan çağrı GMSYP ' yi belirtse bile)
- Kuyruk yöneticisi tarafından oluşturulan olay iletileri (olay iletilerine neden olan çağrı PMSYP ya da GMSYP belirtse bile)

Not:

1. Kural dışı durum raporu iletileri, Message Channel Agent (MCA) ya da uygulama tarafından oluşturulur ve bu nedenle, uygulama tarafından konan ya da alınan olağan iletilerle aynı şekilde işlenir.
2. Bir ileti ya da kesim PMSYP seçeneğiyle birlikte konduğunda, kesinleştirilmemiş iletilerin sayısı, koymadan elde edilen fiziksel iletilerin sayısından bağımsız olarak bir artırılır. (Kuyruk yöneticisinin iletiyi ya da kesimi alt bölümlere ayırması gerekiyorsa birden çok fiziksel ileti oluşabilir.)
3. PMSYP seçeneğiyle bir dağıtım listesi konduğunda, kesinleştirilmemiş iletilerin sayısı, oluşturulan her fiziksel ileti için bir *artırılır*. Bu, bir kişi kadar küçük ya da dağıtım listesindeki hedef sayısı kadar büyük olabilir.

Bu özniteliğin alt sınırı 1; üst sınır 999 999 999 'dur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IAMUNC seçicisini kullanın.

IBM i PerformanceEvent (10 basamaklı işaretli tamsayı) IBM i

Performansla ilgili olayların oluşturulup oluşturulmayacağını denetler.

PerformanceEvent aşağıdaki değerlerden birini içerebilir:

EVRDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IAPFME seçicisini kullanın.

IBM i IBM i üzerinde platform (10 basamaklı işaretli tamsayı)

Kuyruk yöneticisinin çalıştığı platform.

Bu, kuyruk yöneticisinin çalıştığı işletim sistemini gösterir. Değer:

PL400

IBM i.

IBM i PubSubMode (IBM i üzerinde 10 basamaklı işaretli tamsayı)

Yayınlama/abone olma motorunun ve kuyruğa alınan yayınlama/abone olma arabiriminin çalışıp çalışmadığını, bu nedenle uygulamaların uygulama programlama arabirimini ve kuyruğa alınan yayınlama/abone olma arabirimi tarafından izlenmekte olan kuyrukları kullanarak yayınlama/abone olma işlemlerini gerçekleştirmesine izin verir.

Bu, aşağıdaki değerlerden birine sahip olabilir:

PSMCP

Yayınlama/abone olma altyapısı çalışıyor. Bu nedenle, uygulama programlama arabirimini kullanarak yayınlama/abone olma mümkündür. Kuyruğa alınan yayınlama/abone olma arabirimi çalışmıyor; bu nedenle, kuyruğa alınan yayınlama/abone olma arabirimi tarafından izlenen kuyruklara konan hiçbir ileti üzerinde işlem yürütülmez. Bu ayar, kuyruğa alınan yayınlama/abone olma arabiriminin olağan olarak okuduğu kuyrukları okuması gerektiğinden, bu kuyruk yöneticisini kullanan WebSphere Message Broker V6 ya da önceki sürümlerle uyumluluk için kullanılır.

PSMDS

Yayınlama/abone olma altyapısı ve kuyruğa alınan yayınlama/abone olma arabirimi çalışmıyor. Bu nedenle, uygulama programlama arabirimini kullanarak yayınlama/abone olma yapılamaz. Kuyruğa alınan yayınlama/abone olma arabirimi tarafından izlenen kuyruklara konan yayınlama/abone olma iletileri üzerinde işlem uygulanmaz.

PSMEN

Yayınlama/abone olma altyapısı ve kuyruğa alınan yayınlama/abone olma arabirimi çalışıyor. Bu nedenle, kuyruğa alınan yayınlama/abone olma arabirimi tarafından izlenmekte olan kuyruklar ve uygulama programlama arabirimi kullanılarak yayınlama/abone olma mümkündür. Bu, kuyruk yöneticisinin ilk varsayılan değeridir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla PSMODE seçicisini kullanın.

IBM i **QMGrDesc (64 baytlık karakter dizgisi) IBM i**

Kuyruk yöneticisi tanımlaması.

Bu, açıklayıcı açıklama için kullanılabilir bir alandır. Alanın içeriği kuyruk yöneticisi için önemli değil, ancak kuyruk yöneticisi alanın yalnızca görüntülenebilecek karakterleri içermesini gerektirebilir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, bu alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 byte olabilir).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özniteliğinde tanımlandığı gibi) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

IBM i' da varsayılan değer boşluktur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAQMD seçicisini kullanın. Bu özniteliğin uzunluğu LNQMD tarafından verilir.

IBM i **QMGrIdentifler (48 baytlık karakter dizilimi)- IBM i**

Kuyruk yöneticisinin dahili olarak oluşturulan benzersiz tanıtıcısı.

Bu, kuyruk yöneticisi için dahili olarak oluşturulan benzersiz bir addir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAQMID seçicisini kullanın. Bu özniteliğin uzunluğu LNQMID tarafından verilir.

IBM i **QMGrName (48 baytlık karakter dizgisi)- IBM i**

Kuyruk yöneticisi adı.

Bu, uygulamanın bağlı olduğu kuyruk yöneticisinin adı olan yerel kuyruk yöneticisinin adıdır.

Adın ilk 12 karakteri, benzersiz bir ileti tanıtıcısı oluşturmak için kullanılır (bkz. “IBM i üzerinde MQMD (İleti tanımlayıcı)” sayfa 1093 içinde açıklanan *MDMID* alanına bakın). Bu nedenle, ileti tanıtıcılarının kuyruk yöneticisi ağında benzersiz olması için, iletişim kurabilen kuyruk yöneticilerinin ilk 12 karakterden farklı adları olmalıdır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CAQMN seçicisini kullanın. Bu özniteliğin uzunluğu LNQMN tarafından verilir.

IBM i **RemoteEvent (10 basamaklı işaretli tamsayı) IBM i**

Uzak hata olaylarının oluşturulup oluşturulmayacağını denetler.

Değer aşağıdakilerden biridir:

EVRDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IARMTE seçicisini kullanın.

IBM i RepositoryName (48 baytlık karakter dizgisi) on IBM i

Bu kuyruk yöneticisinin havuz hizmetleri sağladığı kümenin adı.

Bu, bu kuyruk yöneticisinin havuz yöneticisi hizmeti sağladığı bir kümenin adıdır. Kuyruk yöneticisi bu hizmeti birden çok küme için sağlıyorsa, *RepositoryNameList* kümeleri tanıtan bir ad listesi nesnesinin adını belirtir ve *RepositoryName* boş olur. *RepositoryName* ve *RepositoryNameList* ürünlerinden en az biri boş olmalıdır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CARPN seçicisini kullanın. Bu özniteliğin uzunluğu LNQMN tarafından verilir.

IBM i RepositoryNameList (48 baytlık karakter dizgisi) on IBM i

Bu kuyruk yöneticisinin havuz hizmetleri sağladığı kümelerin adlarını içeren ad listesi nesnesinin adı.

Bu, bu kuyruk yöneticisinin havuz yöneticisi hizmeti sağladığı kümelerin adlarını içeren bir ad listesi nesnesinin adıdır. Kuyruk yöneticisi bu hizmeti tek bir küme için sağlarsa, ad listesi nesnesi tek bir ad içerir. Diğer bir seçenek olarak, *RepositoryName* kümenin adını belirtmek için kullanılabilir; bu durumda *RepositoryNameList* boş olur. *RepositoryName* ve *RepositoryNameList* ürünlerinden en az biri boş olmalıdır.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla CARPNL seçicisini kullanın. Bu özniteliğin uzunluğu LNNLN tarafından verilir.

IBM i IBM i üzerinde SSLEvent (karakter dizgisi)

TLS olaylarının oluşturulup oluşturulmadığını belirler.

Değer aşağıdakilerden biridir:

- EVRENA (MQINQ/PCF/config olayı) ENABLED (MQSC): TLS olayları oluşturulur (yani, RC2371 olayı oluşturulur).
- EVRDIS (MQINQ/PCF/config olayı) DISABLED (MQSC): TLS olayları oluşturulmaz. Bu, kuyruk yöneticisinin ilk varsayılan değeridir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla IASSLE seçicisini kullanın.

IBM i SSLKeyResetSayı (tamsayı) IBM i

Gizli anahtar yeniden anlaşılmadan önce bir TLS etkileşimi içinde gönderilen ve alınan şifrenmemiş baytların toplam sayısını belirler. Bayt sayısı, ileti kanalı aracısı (MCA) tarafından gönderilen denetim bilgilerini içerir.

Bu değer yalnızca, bu kuyruk yöneticisinden iletişimi başlatan TLS kanal MCA ' ları (gönderen ve alıcı kanal eşleşmesinde gönderen kanal MCA ' sı) tarafından kullanılır.

Bu özniteliğin değeri 0 ' dan büyükse ve bir kanal için kanal sağlıklı işletim bildirimleri etkinleştirildiyse, veriler gönderilmeden ya da bir kanal sağlıklı işletim bildirimleri sonrasında alınmadan önce gizli anahtar da yeniden anlaşılır. Her başarılı yeniden anlaşma gerçekleştiği zaman bir sonraki gizli anahtar yeniden anlaşma sıfırlanmaya kadar bayt sayısı.

Değer, 0-999 999 999 aralığında olabilir. Bu öznitelik için 0 değeri, gizli anahtarın hiçbir zaman yeniden anlaşılmadığını gösterir. 1-32 KB aralığında bir TLS gizli anahtar sıfırlama sayısı belirtirseniz, TLS kanalları 32 KB gizli anahtar sıfırlama sayısını kullanır. Bu, küçük TLS gizli anahtar sıfırlama değerleri için ortaya çıkabilecek aşırı anahtar sıfırlamalarını işleme maliyetini önlemek içindir.

SSL sunucusu bir IBM MQ kuyruk yöneticisi olduğunda ve hem gizli anahtar sıfırlama hem de kanal sağlıklı işletim bildirimleri etkinleştirildiğinde, her bir kanal sağlıklı işletim sinyalinden hemen sonra yeniden anlaşma gerçekleşir.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IASSRC seçicisini kullanın.

IBM i

StartStopOlay (10 basamaklı imzalı tamsayı)- IBM i

Başlatma ve durdurma olaylarının oluşturulup oluşturulmayacağını denetler.

Bu öznitelik aşağıdaki değerlerden birine sahip olabilir:

EVRDIS

Olay raporlama devre dışı bırakıldı.

EVRENA

Olay raporlama etkinleştirildi.

Olaylarla ilgili daha fazla bilgi için bkz. [Olay izleme](#).

Bu öz niteliğin değerini saptamak için, MQINQ çağrısıyla IASSE seçicisini kullanın.

IBM i

IBM i üzerinde SyncPoint (10 basamaklı işaretli tamsayı)

Eşitleme noktası kullanılabilirliği.

Bu, yerel kuyruk yöneticisinin MQGET, MQPUT ve MQPUT1 çağrılarını iş birimlerini destekleyip desteklemediğini gösterir.

SPAVL

İş birimleri ve eşitleme kullanılabilir.

SPNAVL

İş birimleri ve eşitleme kullanılmıyor.

Bu öz niteliğin değerini saptamak için, MQINQ çağrısıyla IASync seçicisini kullanın.

IBM i

TraceRouteKayıt (10 basamaklı imzalı tamsayı) IBM i

Bu, iletilere ilişkin bilgilerin bir kuyruk yöneticisinden akarken kaydedilip kaydedilmeyeceğini denetler.

Değer aşağıdakilerden biridir:

- RECD: İzleme rotası iletilerinin sonuna eklenmesine izin verilmez
- RECDQ: İletiler sabit bir adlandırılmış kuyruğa yerleştirilir
- RECDM: iletiyi kullanarak sapta (ilk varsayılan ayar budur)

İzleme rotası iletilerinin sistemde kalmasını önlemek için, sistemde sıfırdan büyük bir süre sonu değeri belirleyin ve RODISC rapor seçeneğini belirleyin. Sistemde kalan rapor ya da yanıt iletilerini önlemek için rapor seçeneğini ROPDAE olarak ayarlayın. Daha fazla bilgi için "[IBM i üzerinde rapor seçenekleri ve ileti işaretleri](#)" sayfa 1409 başlıklı konuya bakın.

Bu öz niteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IATRGI seçicisini kullanın.

IBM i

TreeLife IBM i üzerinde saat (10 basamaklı işaretli tamsayı)

Yönetimle ilgili olmayan konuların geçerlik süresi (saniye olarak).

Yönetimle ilgili olmayan konular, bir uygulama denetim düğümü olarak varolmayan bir konu dizgisine yayınlandığında ya da bu dizgiye abone olduğunda yaratılan konulardır. Denetim düğümü olmayan bu düğümün artık etkin aboneliği yoksa, bu değiştirge kuyruk yöneticisinin o düğümü kaldırmadan önce ne kadar bekleyeceğini belirler. Kuyruk yöneticisi yeniden dönüştürüldükten sonra yalnızca sürekli abonelik tarafından kullanılan yönetimle ilgili olmayan konular kalır.

0-604 000 aralığında bir değer belirtin. 0 değeri, yönetici olmayan konuların kuyruk yöneticisi tarafından kaldırılmadığı anlamına gelir. Kuyruk yöneticisinin ilk varsayılan değeri 1800 'dür.

Bu öz niteliğin değerini saptamak için, MQINQ çağrısıyla IATRLFT seçicisini kullanın.

IBM i

IBM i üzerinde TriggerInterval (10 basamaklı işaretli tamsayı)

Tetikleyici ileti aralığı.

Bu, tetikleyici iletilerin sayısını sınırlamak için kullanılan bir zaman aralığıdır (milisaniye cinsinden). Bu yalnızca *TriggerType* TFRST olduğunda geçerlidir. Bu durumda, tetikleyici iletiler olağan durumda yalnızca kuyruğa uygun bir ileti geldiğinde ve kuyruk daha önce boş olduğunda üretilir. Bununla birlikte, belirli koşullar altında, kuyruk boş olmasa bile TFRST tetiklemeyle birlikte ek bir tetikleyici ileti oluşturulabilir. Bu ek tetikleyici iletileri her *TriggerInterval* milisaniyeden daha sık oluşturulmaz.

Tetikleme hakkında daha fazla bilgi için bkz. [Kanalları tetikleme](#).

Değer, sıfırdan 999 999 999 999 aralığındadır. Varsayılan değer 999 999 999 'dur.

Bu özniteliğin değerini saptamak için, MQINQ çağrısıyla birlikte IATRGI seçicisini kullanın.

Uygulamalar

Bu bilgiler, RPG için IBM MQ for IBM i ile birlikte gönderilen örnek programları açıklar. Ayrıca, yazdığınız programlardan yürütülebilir uygulamaların nasıl oluşturulacağını öğrenin.

Uygulamanızı oluşturma

IBM i yayınları, yazdığınız programlardan yürütülebilir uygulamaların nasıl oluşturulacağını açıklar. Bu konuda, ek görevler ve standart görevlerde yapılan değişiklikler açıklanmaktadır. IBM i altında çalışacak IBM MQ for IBM i uygulamaları oluştururken gerçekleştirmeniz gerekir.

Kaynak kodunuzda MQI çağrılarını kodlamaya ek olarak, RPG diline ilişkin IBM MQ for IBM i kopya dosyalarını eklemek için uygun dil deyimlerini de eklemeniz gerekir. Bu dosyaların içeriğini tanımanız gerekir; bu dosyaların adları ve içeriklerinin kısa bir açıklaması aşağıdaki metinde verilmiştir.

IBM i IBM MQ dosyaları kopyalama IBM i

IBM MQ for IBM i , RPG programlama dilinde uygulamalarınızı yazmanıza yardımcı olacak kopyalama dosyaları sağlar. Bunlar, WebSphere Development Toolset (5722 WDS) ILE RPG 4 Compiler ile kullanıma uygundur.

IBM MQ for IBM i ' in sağladığı kopya dosyaları, kanal çıkışlarının yazılmasına yardımcı olmak için [İleti alışverişi kanalları için kanal çıkış programları](#) başlıklı konuda açıklanmaktadır.

RPG için IBM MQ for IBM i kopyalama dosyalarının adları CMQ öneğine sahip olur. "G" ya da "H" ekine sahipler. Adlandırılmış değişmezleri içeren ayrı kopya dosyaları ve yapıların her biri için bir dosya vardır. Kopyalama dosyaları "[Dille ilgili dikkat edilmesi gereken noktalar](#)" sayfa 994'inde listelenir.

Not: ILE RPG/400 için, dosya üyeleri olarak sağlanır. QMQM kitaplığındaki QRPGLSERC.

Yapı bildirimleri DS deyimleri içermiyor. Bu, uygulamanın DS deyimini kodlayarak ve bildirim geri kalan kısmında kopyalamak için /COPY deyimini kullanarak bir veri yapısını (ya da çoklu geçiş veri yapısını) bildirmesini sağlar:

ILE RPG/400 için deyim şöyledir:

```
D*..1.....2.....3.....4.....5.....6.....7
D* Declare an MQMD data structure
D MQMD          DS
D/COPY CMQMDG
```

Programlarınızın çalıştırılmak üzere hazırlanması

Yürütülebilir bir IBM MQ for IBM i uygulaması oluşturmak için yazdığınız kaynak kodu derlemeniz gerekir.

Bunu ILE RPG/400 için yapmak üzere tipik IBM i komutlarını, CRTRPGMOD ve CRTPGM komutlarını kullanabilirsiniz.

*MODULE modülünüzü yarattıktan sonra, CRTPGM komutunda BNDSRVPGM (QMQM/LIBMQM) değerini belirlemeniz gerekir. Bu, programınızdaki çeşitli IBM MQ yordamlarını içerir.

Derleme işlemini gerçekleştirirken, kopyalama dosyalarını (QMOM) içeren kitaplığın kitaplık listesinde bulunduğundan emin olun.

İstemci kipleri de içinde olmak üzere, programlama konularına ilişkin daha fazla bilgi için bkz. [“Dille ilgili dikkat edilmesi gereken noktalar” sayfa 994.](#)

IBM i dış eşitleme noktası yöneticisine ilişkin arabirimler

IBM MQ for IBM i , dış uyumlulaştırma noktası eşgüdümçüsü olarak yerel IBM i kesinleştirme denetimini kullanır.

IBM i' in kesinleştirme denetimi yeteneklerine ilişkin ek bilgi için *IBM i Programming: Backup and Recovery Guide* belgesine bakın.

IBM i kesinleştirme denetimi olanaklarını başlatmak için STRCMTCTL sistem komutunu kullanın. Kesinleştirme denetimini sona erdirmek için ENDCMTCTL sistem komutunu kullanın.

Not: *Kesinleştirme tanımı kapsamı* varsayılan değeri *ACTGRP ' dir. Bu, IBM için IBM MQ için *JOB olarak tanımlanmalıdır. Örneğin:

```
STRCMTCTL LCKLVL(*ALL) CMTSCOPE(*JOB)
```

Kesinleştirme denetimi başlatıldıktan sonra PMSYP ya da GMSYP belirterek MQPUT, MQPUT1ya da MQGET çağrılırsa, IBM MQ for IBM i kendisini kesinleştirme tanımına API kesinleştirme kaynağı olarak ekler. Bu genellikle bir işte bu tür ilk çağrıdır. Belirli bir kesinleştirme tanımı altında kayıtlı API kesinleştirme kaynakları olsa da, bu tanım için kesinleştirme denetimini sona erdiremezsiniz.

IBM MQ for IBM i , yürürlükteki iş biriminde beklemedeki MQI işlemleri yoksa, kuyruk yöneticisiyle bağlantınızı kestiğinizde API kesinleştirme kaynağı olarak kaydını kaldırır.

Yürürlükteki iş biriminde bekleyen MQPUT, MQPUT1ya da MQGET işlemleri varken kuyruk yöneticisiyle bağlantınızı keserseniz, IBM MQ for IBM i sonraki kesinleştirme ya da geriye işleme işleminin bildirilmesi için bir API kesinleştirme kaynağı olarak kayıtlı kalır. Sonraki eşitleme noktasına ulaşıldığında, IBM MQ değişiklikleri gerektiği gibi kesinleştirir ya da geri alır. Bir uygulamanın etkin bir iş birimi sırasında bir kuyruk yöneticisiyle bağlantısı kesilmesi ve yeniden bağlanması ve aynı iş birimi içinde başka MQGET ve MQPUT işlemleri gerçekleştirilmesi mümkündür (bu, bekleyen bir bağlantıdır).

Bu kesinleştirme tanımı için bir ENDCMTCTL sistem komutu verme girişiminde bulunursanız, bekleyen değişikliklerin etkin olduğunu belirten CPF8355 iletisi görüntülenir. Bu ileti, iş sona erdiğinde iş günlüğünde de görüntülenir. Bunu önlemek için, bekleyen tüm IBM MQ işlemlerini kesinleştirdiğinizden ya da geri aldığınızdan ve kuyruk yöneticisiyle bağlantınızı kestiğinizden emin olun. Bu nedenle, ENDCMTCTL ' den önce COMMIT ya da ROLLBACK komutlarının kullanılması, kesinleştirme sonu denetiminin başarıyla tamamlanmasını sağlamalıdır.

IBM i kesinleştirme denetimi dış uyumlulaştırma noktası eşgüdümçüsü olarak kullanıldığında, MQCMIT, MQBACK ve MQBEGIN çağrıları yayınlanmayabilir. Bu işlemlere yapılan çağrılar RC2012neden koduyla başarısız olur.

İş biriminizi kesinleştirmek ya da geri yüklemek (yani, geri almak) için, kesinleştirme denetimini destekleyen programlama dillerinden birini kullanın. Örneğin:

- CL komutları: COMMIT ve ROLLBACK
- ILE C Programlama İşlevleri: _Rcommit ve _Rollback
- RPG/400: COMMIT ve ROLBK
- COBOL/400: COMMIT ve ROLLBACK

IBM i uygulamaları için CICS içindeki eşitleme noktaları

IBM MQ for IBM i , CICS ile birlikte çalışma birimlerine katılır. İletileri yürürlükteki iş biriminin içine koymak ve almak için bir CICS uygulaması içindeki MQI ' ı kullanabilirsiniz.

IBM MQ for IBM i işlemlerini içeren bir eşitleme noktası oluşturmak için EXEC CICS SYNCPOINT komutunu kullanabilirsiniz. Önceki eşitleme noktasına kadar olan tüm değişiklikleri geri yüklemek için EXEC CICS SYNCPOINT ROLLBACK komutunu kullanabilirsiniz.

Bir CICS uygulamasında ayarlı MQPUT, MQPUT1 ya da PMSYP ya da GMSYP ile MQGET seçeneğini kullanırsanız, IBM MQ for IBM i kaydını API kesinleştirme kaynağı olarak kaldırıncaya kadar CICS oturumunuzu kapatamazsınız. Bu nedenle, kuyruk yöneticisiyle bağlantınızı kesmeden önce, bekleyen koyma ya da alma işlemlerini kesinleştirmeniz ya da geri almanız gerekir. Bu, CICS oturumunu kapatmanızı sağlar.

IBM i üzerinde örnek programlar

Bu konuda, RPG için IBM MQ for IBM i ile birlikte gönderilen örnek programlar açıklanmaktadır. Örnekler, İleti Kuyruğu Arabirimi 'nin (MQI) tipik kullanımlarını gösterir.

Bu örneklerin genel programlama tekniklerini göstermesi amaçlanmamıştır, bu nedenle bir üretim programına dahil etmek isteyebileceğiniz bazı hata kontrolleri atlanmıştır. Ancak bu örnekler, kendi ileti kuyruklama programlarınız için temel olarak kullanılmaya uygundur.

Tüm örneklerin kaynak kodu ürünle birlikte sağlanır; bu kaynak, programlarda gösterilen ileti kuyruklama tekniklerini açıklayan açıklamalar içerir.

Bir dizi ILE örnek programı vardır:

1. MQI 'a prototip çağrılarını kullanan programlar (statik bağlı çağrılar)

Kaynak QMQMSAMP/QRPGLESRC içinde var. Üyeler AMQ3xxx4 olarak adlandırılır; burada xxx örnek işlevi gösterir. Kopya üyeleri QMQM/QRPGLESRC içinde var. Her üye adının bir son eki G ya da Holur.

Çizelge 811 sayfa 1392 içinde, IBM MQ for IBM i ile birlikte gönderilen örnek programların tam bir listesi verilir ve desteklenen programlama dillerinin her birinde programların adları gösterilir. Adların hepsinin AMQ önekiyle başladığını, addaki dördüncü karakterin programlama dilini gösterdiğini fark edin.

Çizelge 811. Örnek programların adları	
	RPG (ILE)
Koyma örnekleri	AMQ3PUT4
Örneklere göz at	AMQ3GBR4
Örneklere al	AMQ3GET4
İstek örnekleri	AMQ3REQ4
Echo örnekleri	AMQ3ECH4
Sorma örnekleri	AMQ3INQ4
Örneklere ayarla	AMQ3SET4
Tetikleyici İzleyicisi örneği	AMQ3TRG4
Trigger Server Sample	AMQ3SRV4

Bunlara ek olarak, IBM MQ for IBM i örnek seçeneği, yönetim görevlerini gösteren belirli örnek programlara ve örnek CL programlarına giriş olarak kullanılabilen AMQSDATA adlı bir örnek veri dosyası içerir. CL örnekleri, [Yönetme IBM i](#) başlıklı konuda açıklanmıştır. Bu konuda açıklanan örnek programlarla kullanılacak kuyruklar yaratmak için örnek CL programını kullanabilirsiniz.

Örnek programların nasıl çalıştırılacağı hakkında bilgi için bkz. ["IBM i üzerinde örnek programların hazırlanması ve çalıştırılması"](#) sayfa 1393.

IBM i üzerindeki örnek programlarda gösterilen özellikler

IBM MQ for IBM i örnek programları tarafından gösterilen teknikleri gösteren bir çizelge.

Bazı teknikler birden çok örnek programda ortaya çıkar, ancak çizelgede yalnızca bir program listelenir. MQOPEN ve MQCLOSE çağrılarını kullanan tüm örnekler açık ve kapalı kuyruklar, bu nedenle bu teknikler çizelgede ayrı olarak listelenmez.

<i>Çizelge 812. MQI kullanımını gösteren örnek programlar</i>	
Teknik	RPG (ILE)
MQCONN ve MQDISC çağrılarının kullanılması	AMQ3ECH4 ya da AMQ3INQ4
Örtük olarak bağlanıyor ve bağlantı kesiliyor	AMQ3PUT4
İletilerin MQPUT çağrısı kullanılarak konmasını sağlar	AMQ3PUT4
MQPUT1 çağrısı kullanılarak tek bir ileti konması	AMQ3ECH4 ya da AMQ3INQ4
İstek iletinin yanıtlanması	AMQ3INQ4
İletiler alınıyor (bekleme yok)	AMQ3GBR4
İleti alma (süre sınırı ile bekleme)	AMQ3GET4
İleti alma (veri dönüştürme ile)	AMQ3ECH4
Kuyruğa göz atma	AMQ3GBR4
Paylaşılan giriş kuyruğunun kullanılması	AMQ3INQ4
Dışlayıcı giriş kuyruğunun kullanılması	AMQ3REQ4
MQINQ çağrısı kullanma	AMQ3INQ4
MQSET çağrısı kullanma	AMQ3SET4
Yanıt Kuyruğunun Kullanılması	AMQ3REQ4
Kural dışı durum iletileri isteniyor	AMQ3REQ4
Kesilmiş bir iletiyi kabul etme	AMQ3GBR4
Çözülmüş kuyruk adının kullanılması	AMQ3GBR4
Tetikleyici işleme	AMQ3SRV4 ya da AMQ3TRG4

Not: Tüm örnek programlar, işlemin sonuçlarını içeren bir kuyruğa yollanmış dosya üretir.

IBM i üzerinde örnek programların hazırlanması ve çalıştırılması

IBM MQ for IBM i örnek programlarını çalıştırmadan önce, bunları diğer IBM MQ for IBM i uygulamaları gibi derlemeniz gerekir. Bunu yapmak için, CRTRPGMOD ve CRTPGM IBM i komutlarını kullanabilirsiniz.

AMQ3xxx4 programlarını oluşturduğunuzda, CRTPGM komutunda BNDSRVPGM (QMQM/LIBMQM) belirtmeniz gerekir. Bunu yapmak, programınızdaki çeşitli IBM MQ yordamlarını içerir.

Örnek programlar QMQMSAMP kitaplığında QRPGLSRC üyeleri olarak sağlanır. QMQM kitaplığında sağlanan kopyalama dosyalarını kullanırlar; bu nedenle, bu kitaplığı derlerken kitaplık listesinde bulunduğundan emin olun. Örnekler, kopya dosyalarında bildirilen değişkenlerin çoğunu kullanmadığından RPG derleyicisi bilgi iletileri verir.

Örnek programların çalıştırılması

Örnekleri çalıştırırken kendi kuyruklarınızı kullanabilir ya da örnek kuyruklar yaratmak için AMQSAMP4 komutunu derleyip çalıştırabilirsiniz. Bu programın kaynağı, QMQMSAMP kitaplığındaki QCLSRC dosyasında gönderilir. CRTCLPGM komutu kullanılarak derlenebilir.

Örnek programlardan birini çağırarak için aşağıdaki gibi bir komut kullanın:

```
CALL PGM(QMQMSAMP/AMQ3PUT4) PARM('Queue_Name', 'Queue_Manager_Name')
```

Burada Queue_Name ve Queue_Manager_Name 48 karakter uzunluğunda olmalıdır; bu uzunluğu Queue_Name ve Queue_Manager_Name 'yi gerekli sayıda boşlukla doldurarak elde edebilirsiniz.

Inquire and Set örnek programları için, AMQSAMP4 tarafından yaratılan örnek tanımlar, bu örneklerin C sürümlerinin tetiklenmesine neden olur. RPG sürümlerini tetiklemek istiyorsanız, SYSTEM.SAMPLE.ECHOPROCESS ve SYSTEM.SAMPLE.INQPROCESS ve SYSTEM.SAMPLE.SETPROCESS. CHGMQMPCRC komutunu kullanabilirsiniz (Change MQ Process (CHGMQMPCRC) başlıklı konuda açıklanmıştır). Bunu yapmak için ya da diğer tanımla AMQSAMP4 'ü düzenlemek ve çalıştırmak için.

IBM i üzerinde Put örnek programı

Put örnek programı AMQ3PUT4, iletileri MQPUT çağrıyla kuyruğa koyar.

Programı başlatmak için programı çağırın ve program parametresi olarak hedef kuyruğunuzun adını verin. Program kuyruğa bir dizi sabit ileti koyar; bu iletiler, program kaynak kodunun sonundaki veri bloğundan alınır. Örnek bir koyma programı, QMQMSAMP kitaplığındaki AMQ3PUT4 programıdır.

Bu örnek program kullanılarak, komut aşağıdaki gibidir:

```
CALL PGM(QMQMSAMP/AMQ3PUT4) PARM('Queue_Name','Queue_Manager_Name')
```

Burada Queue_Name ve Queue_Manager_Name 48 karakter uzunluğunda olmalıdır; bu uzunluğu Queue_Name ve Queue_Manager_Name 'yi gerekli sayıda boşlukla doldurarak elde edebilirsiniz.

Put örnek programının tasarımı

Program, iletileri koymak üzere hedef kuyruğu açmak için OOOOUT seçeneğiyle MQOPEN çağrısıyla birlikte kullanılır. Sonuçlar, kuyruğa yollanmış bir dosyaya çıkışa yazılır. Kuyruğu açamazsa, program MQOPEN çağrısıyla döndürülen neden kodunu içeren bir hata ileti yazar. Programı basit tutmak için, bu ve sonraki MQI çağrılarında, program birçok seçenek için varsayılan değerleri kullanır.

Kaynak kodda bulunan her veri satırı için, program metni bir arabelleğe okur ve o satırın metnini içeren bir veri paketi ileti yaratmak için MQPUT çağrısı kullanır. Program, girişin sonuna ulaşınca ya da MQPUT çağrısı başarısız oluncaya kadar devam eder. Program girişin sonuna ulaşırsa, MQCLOSE çağrısı kullanarak kuyruğu kapatır.

IBM i üzerinde Göz At örnek programı

AMQ3GBR4örnek programı, MQGET çağrısı kullanarak kuyruktaki iletilere göz atmanızı sağlar.

Program, programı çağırdığınızda belirlediğiniz kuyruktaki tüm iletilerin kopyalarını alır; iletiler kuyruktaki kalır. Sağlanan SYSTEM.SAMPLE.LOCAL; kuyruğa bazı iletiler koymak için önce Put örnek programını çalıştırın. SYSTEM.SAMPLE.ALIAS. Program, kuyruğun sonuna ulaşınca ya da bir MQI çağrısı başarısız oluncaya kadar devam eder.

RPG programını çağırmak için bir komut örneği:

```
CALL PGM(QMQMSAMP/AMQ3GBR4) PARM('Queue_Name','Queue_Manager_Name')
```

Burada Queue_Name ve Queue_Manager_Name 48 karakter uzunluğunda olmalıdır; bu uzunluğu Queue_Name ve Queue_Manager_Name 'yi gerekli sayıda boşlukla doldurarak elde edebilirsiniz. Bu nedenle, SYSTEM.SAMPLE.LOCAL , 29 boş karaktere gereksinim duyarsınız.

Göz At örnek programının tasarımı

Program, OOBROW seçeneğiyle MQOPEN çağrısıyla hedef kuyruğu açar. Kuyruğu açamazsa, program kuyruk dosyasına MQOPEN çağrısıyla döndürülen neden kodunu içeren bir hata ileti yazar.

Kuyruktaki her ileti için, program iletiyi kuyruktan kopyalamak için MQGET çağrısı kullanır ve iletide bulunan verileri görüntüler. MQGET çağrısı şu seçenekleri kullanır:

GMBRWN

MQOPEN çağrısından sonra, göz atma imleci kuyruktaki ilk iletiden önce mantıksal olarak konumlandırılır; bu nedenle, arama ilk yapıldığında *first* (ilk) iletisi döndürülür.

GMNWT

Kuyrukta ileti yoksa program beklemez.

GMATM

MQGET çağrısı, değişmez büyüklükte bir arabellek belirtiyor. Bir ileti bu arabellekten uzunsa, program kesilen iletiyi, iletinin kesildiğine ilişkin bir uyarıyla birlikte görüntüler.

Çağrı, bu alanları aldığı iletide bulunan değerlere ayarladığından, her MQGET çağrısından sonra MQMD yapısının *MDMID* ve *MDCID* alanlarını nasıl temizlemeniz gerektiğini gösterir. Bu alanların temizlenmesi, ardışık MQGET çağrıları iletilerin kuyrukta tutulduğu sırada iletileri alır.

Program kuyruğun sonuna kadar devam eder; burada MQGET çağrısı RC2033 (ileti yok) neden kodunu döndürür ve program bir uyarı iletisi görüntüler. MQGET çağrısı başarısız olursa, program kuyruğa yollanmış dosyasında neden kodunu içeren bir hata iletisi yazar.

Daha sonra program, MQCLOSE çağrılarını kullanarak kuyruğu kapatır.

IBM i üzerinde Get sample programı

Örnek programı al (AMQ3GET4), MQGET çağrısını kullanarak kuyruktan ileti alır.

Program çağrıldığında, belirtilen kuyruktaki iletileri kaldırır. Sağlanan SYSTEM.SAMPLE.LOCAL; kuyruğa bazı iletiler koymak için önce Put örnek programını çalıştırın. SYSTEM.SAMPLE.ALIAS kuyruğu. Kuyruk boş oluncaya ya da bir MQI çağrısı başarısız oluncaya kadar program devam eder.

RPG programını çağırarak için bir komut örneği:

```
CALL PGM(QMQMSAMP/AMQ3GET4) PARM('Queue_Name', 'Queue_Manager_Name')
```

Burada Queue_Name ve Queue_Manager_Name 48 karakter uzunluğunda olmalıdır; Queue_Name ve Queue_Manager_Name için gereken sayıda boşluk doldurarak bu değeri elde edebilirsiniz. Bu nedenle, SYSTEM.SAMPLE.LOCAL , 29 boş karaktere gereksinim duyarsınız.

Örnek Alma programının tasarımı

Program, iletileri almak için hedef kuyruğu açar; OOINPQ seçeneğiyle MQOPEN çağrısını kullanır. Kuyruğu açamazsa, program kuyruğa yollanmış dosyasında MQOPEN çağrısıyla döndürülen neden kodunu içeren bir hata iletisi yazar.

Kuyruktaki her ileti için, program iletiyi kuyruktan kaldırmak için MQGET çağrısını kullanır; daha sonra, iletide bulunan verileri görüntüler. MQGET çağrısı, kuyrukta ileti yoksa programın bu süreyi beklemesi için 15 saniyelik bir bekleme aralığı (*GMWT*) belirterek GMWT seçeneğini kullanır. Bu aralığın süresi dolmadan hiçbir ileti gelmezse, çağrı başarısız olur ve RC2033 (ileti yok) neden kodunu döndürür.

Çağrı, bu alanları aldığı iletide bulunan değerlere ayarladığından, her MQGET çağrısından sonra MQMD yapısının *MDMID* ve *MDCID* alanlarını nasıl temizlemeniz gerektiğini gösterir. Bu alanların temizlenmesi, ardışık MQGET çağrıları iletilerin kuyrukta tutulduğu sırada iletileri alır.

MQGET çağrısı, değişmez büyüklükte bir arabellek belirtiyor. Bir ileti bu arabellekten daha uzunsa, çağrı başarısız olur ve program durur.

MQGET çağrısı RC2033 (ileti yok) neden kodunu döndürünceye ya da MQGET çağrısı başarısız oluncaya kadar program devam eder. Çağrı başarısız olursa, program neden kodunu içeren bir hata iletisi görüntüler.

Daha sonra program, MQCLOSE çağrılarını kullanarak kuyruğu kapatır.

IBM i üzerinde Request örnek programı

İstek örnek programı AMQ3REQ4, istemci/sunucu işlemlerini gösterir. Örnek, istek iletilerini bir sunucu programı tarafından işlenen bir kuyruğa koyan istemcidir. Sunucu programının yanıt kuyruğuna bir yanıt iletilisi koymasını bekler.

İstek örneği, MQPUT çağrısının kullanıldığı bir dizi istek iletilisini kuyruğa koyar. Bu iletiler SYSTEM.SAMPLE.REPLY yanıtlayın. Program yanıt iletilerini bekler ve bunları görüntüler. Yanıtlar yalnızca hedef kuyruk (*sunucu kuyruğunu* çağıracağımız kuyruk) bir sunucu uygulaması tarafından işleniyor ya da bu amaçla bir uygulama tetikleniyorsa (Inquire and Set örnek programları tetiklenecek şekilde tasarlanmıştır). Örnek, ilk yanıtın gelmesi için 5 dakika (bir sunucu uygulamasının tetiklenmesine izin vermek için) ve sonraki yanıtlar için 15 saniye bekler, ancak yanıt almadan sona erebilir.

Programı başlatmak için programı çağırın ve program parametresi olarak hedef kuyruğunuzun adını verin. Program kuyruğa bir dizi sabit ileti koyar; bu iletiler, program kaynak kodunun sonundaki veri bloğundan alınır.

İstek örnek programının tasarımı

Program, iletileri yerleştirebilmesi için sunucu kuyruğunu açar. OOOOUT seçeneğiyle MQOPEN çağrısı kullanır. Kuyruğu açamazsa, program MQOPEN çağrısıyla döndürülen neden kodunu içeren bir hata iletilisi görüntüler.

Program daha sonra SYSTEM.SAMPLE.REPLY , böylece yanıt iletileri alabilir. Bu nedenle, program OOINPX seçeneğiyle MQOPEN çağrısı kullanır. Kuyruğu açamazsa, program MQOPEN çağrısıyla döndürülen neden kodunu içeren bir hata iletilisi görüntüler.

Her giriş satırı için, program metni bir arabelleğe okur ve o satırın metnini içeren bir istek iletilisi yaratmak için MQPUT çağrısı kullanır. Bu çağrıda program, istek iletilisiyle ilgili olarak gönderilen rapor iletilerinin ileti verilerinin ilk 100 baytını içermesini istemek için ROEXCD rapor seçeneğini kullanır. Program, girişin sonuna ulaşıncaya ya da MQPUT çağrısı başarısız oluncaya kadar devam eder.

Daha sonra, program yanıt iletilerini kuyruktan kaldırmak için MQGET çağrısını kullanır ve yanıtlarda bulunan verileri görüntüler. MQGET çağrısı, ilk yanıt (sunucu uygulamasının tetiklenmesine izin vermek için) için 5 dakikalık bir bekleme aralığı (GMWI) ve sonraki yanıtlar için 15 saniye belirterek GMWT seçeneğini kullanır. Kuyruқта ileti yoksa, program bu dönemleri bekler. Bu aralığın süresi dolmadan hiçbir ileti gelmezse, çağrı başarısız olur ve RC2033 (ileti yok) neden kodunu döndürür. Çağrı ayrıca GMATM seçeneğini kullanır, bu nedenle bildirilen arabellek boyutundan daha uzun iletiler kesilir.

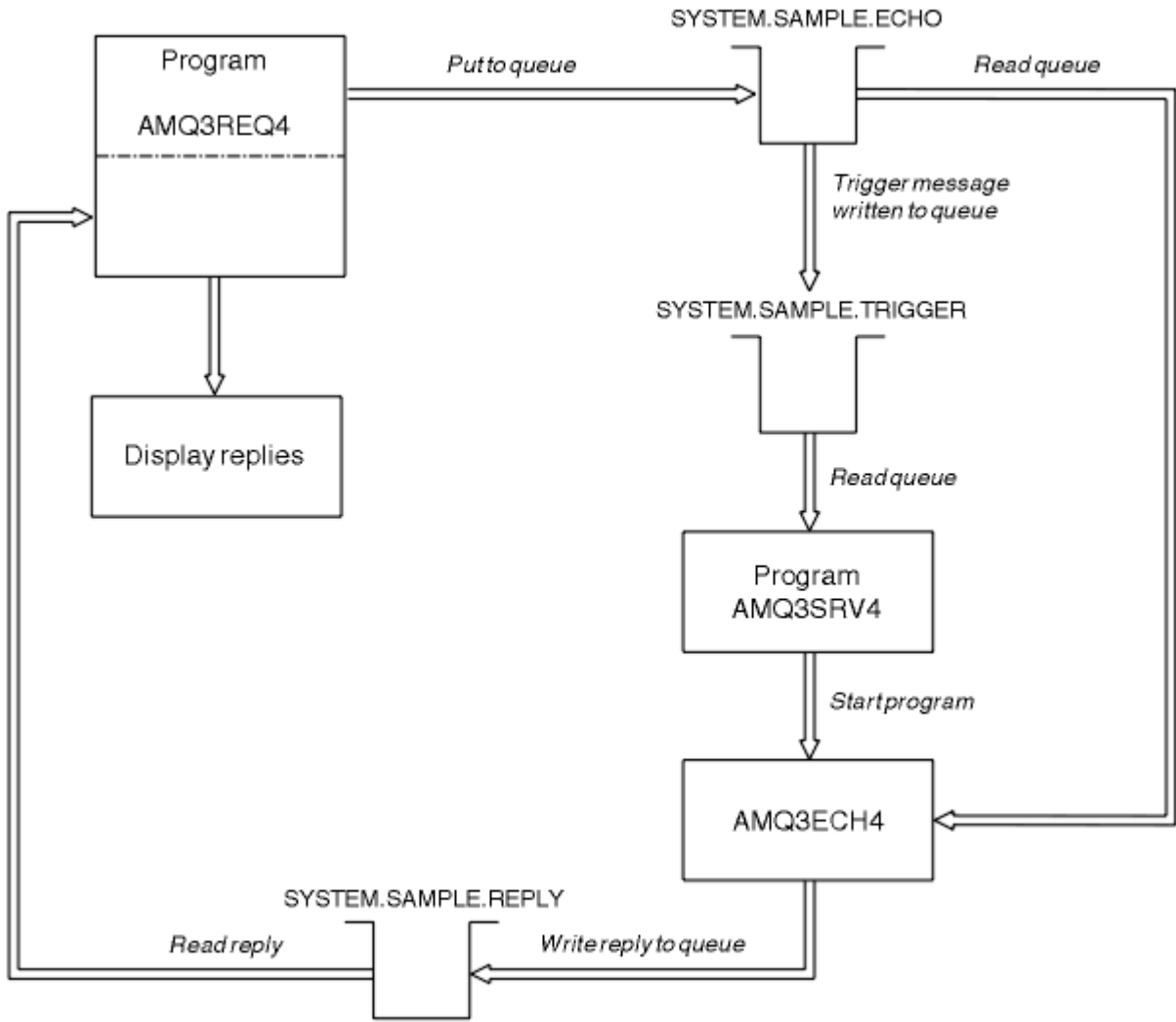
Çağrı, bu alanları aldığı iletide bulunan değerlere ayarladığından, her MQGET çağrısından sonra MQMD yapısının MDMID ve MDCOD alanlarını nasıl temizlemeniz gerektiğini gösterir. Bu alanların temizlenmesi, ardışık MQGET çağrıları iletilerin kuyruқта tutulduğu sırada iletileri alır.

MQGET çağrısı RC2033 (ileti yok) neden kodunu döndürünceye ya da MQGET çağrısı başarısız oluncaya kadar program devam eder. Çağrı başarısız olursa, program neden kodunu içeren bir hata iletilisi görüntüler.

Daha sonra program, MQCLOSE çağrısı kullanarak hem sunucu kuyruğunu hem de yanıt kuyruğunu kapatır. Çizelge 813 sayfa 1396 içinde, Inquire and Set örnek programlarını çalıştırmak için gerekli olan Echo örnek programında yapılan değişiklikler gösterilmektedir.

Not: Echo örnek programına ilişkin ayrıntılar başvuru olarak eklenmiştir.

<i>Çizelge 813. İstemci/Sunucu örnek program ayrıntıları</i>		
Program adı	SYSTEM/SAMPLE kuyruğu	Program başlatıldı
Yankı	YANKI	AMQ3ECH4
Sor	GÇÇ	AMQ3INQ4
Ayarla	Belirle	AMQ3SET4



Şekil 9. Örnek İstemci/Sunucu (Echo) programı akış şeması

IBM i IBM i üzerinde İstek örneğiyle tetikleme kullanılması

Örneği tetikleme kullanarak çalıştırmak için, AMQ3SRV4 tetikleyici sunucu programını, bir işte gerekli başlatma kuyruğuna karşı başlatın ve daha sonra, başka bir işte AMQ3REQ4 işlemini başlatın.

Bu, İstek örnek programı bir ileti gönderdiğinde tetikleyici sunucusunun hazır olduğu anlamına gelir.

Not:

1. Örnekler, SYSTEM.SAMPLE.ECHO, SYSTEM.SAMPLE.INQ ya da SYSTEM.SAMPLE.SET yerel kuyrukları. Diğer bir seçenek olarak, kendi başlatma kuyruğunuzu tanımlayabilirsiniz.
2. AMQSAMP4 tarafından yaratılan örnek tanımlar, örneğin C sürümünün tetiklenmesine neden olur. RPG sürümünü tetiklemek istiyorsanız, SYSTEM.SAMPLE.ECHOPROCESS ve SYSTEM.SAMPLE.INQPROCESS ve SYSTEM.SAMPLE.SETPROCESS. Bunu yapmak için CHGMQMPRC komutunu kullanabilir (daha fazla ayrıntı için [Change MQ Process \(CHGMQMPRC\)](#) başlıklı konuya bakın) ya da kendi AMQSAMP4 sürümünüzü düzenleyebilir ve çalıştırabilirsiniz.
3. Tetikleyici sunucu programını QMQMSAMP/QRPGLESRC içinde sağlanan kaynaktan derlemelisiniz.

Çalıştırmak istediğiniz tetikleme işlemine bağlı olarak, AMQ3REQ4 parametresi aşağıdaki örnek sunucu kuyruklarından birine yerleştirilecek istek iletilerini belirterek çağrılmalıdır:

- SYSTEM.SAMPLE.ECHO (Echo örnek programları için)
- SYSTEM.SAMPLE.INQ (Inquire örnek programları için)
- SYSTEM.SAMPLE.SET (Set örnek programları için)

SYSTEM.SAMPLE.ECHO programı Şekil 9 sayfa 1397’inde gösterilir. Bu sunucu için RPG programı isteğini yayınlamak için aşağıdaki komutu kullanın:

```
CALL PGM(QMQMSAMP/AMQ3REQ4) PARM('SYSTEM.SAMPLE.ECHO  
+ 30 blank characters','Queue_Manager_Name')
```

Çünkü kuyruk adı ve kuyruk yöneticisi adı 48 karakter uzunluğunda olmalıdır.

Not: Bu örnek kuyruk FIRST tetikleyici tipine sahiptir; bu nedenle, İstek örneğini çalıştırmadan önce kuyrukta önceden ileti varsa, sunucu uygulamaları gönderdiğiniz iletiler tarafından tetiklenmez.

Başka örnekler denemek isterseniz, aşağıdaki çeşitlemeleri deneyebilirsiniz:

- Bunun yerine işi göndermek için AMQ3SRV4 yerine AMQ3TRG4 kullanın, ancak olası iş gönderimi gecikmeleri, olan bitenin izlenmesini daha az kolaylaştıracaktır.
- SYSTEM.SAMPLE.INQ ve SYSTEM.SAMPLE.SET örnek kuyrukları. Örnek veri dosyasını kullanarak, RPG programı isteklerini bu sunuculara yayınlamak için aşağıdaki komutlar kullanılabilir:

```
CALL PGM(QMQMSAMP/AMQ3INQ4) PARM('SYSTEM.SAMPLE.INQ  
+ 31 blank characters')  
CALL PGM(QMQMSAMP/AMQ3SET4) PARM('SYSTEM.SAMPLE.SET  
+ 31 blank characters')
```

Çünkü kuyruk adı 48 karakter uzunluğunda olmalıdır.

Bu örnek kuyrukların tetikleyici tipi de FIRST ' dir.

IBM i üzerinde Yankı örnek programı

Echo örnek programları, bir yanıt kuyruğuna gönderilen iletiyi döndürür. Programın adı AMQ3ECH4

Tetikleme işleminin çalışması için, kullanmak istediğiniz Echo örnek programının SYSTEM.SAMPLE.ECHO. Bunu yapmak için, *AppLId* süreç tanımlamasının alanında SYSTEM.SAMPLE.ECHOPROCESS kullanmak istediğiniz Echo örnek programının adını belirtin. (Bunun için, [Yönetme IBM i](#) başlıklı konuda açıklanan CHGMQMPRC komutunu kullanabilirsiniz.) Örnek kuyruğun tetikleyici tipi FIRST olduğundan, İstek örneğini çalıştırmadan önce kuyrukta önceden ileti varsa, Echo örneği gönderdiğiniz iletiler tarafından tetiklenmez.

Tanımı doğru olarak ayarladığınızda, önce bir işte AMQ3SRV4 'ü başlatın, daha sonra başka bir işte AMQ3REQ4 ' ü başlatın. AMQ3SRV4 yerine AMQ3TRG4 kullanabilirsiniz, ancak olası iş gönderimi gecikmeleri, olan biteni izlemeyi daha az kolaylaştıracaktır.

SYSTEM.SAMPLE.ECHO. Echo örnek programları, istek iletisindeki verileri içeren bir yanıt iletisini istek iletisinde belirtilen yanıt kuyruğuna gönderir.

Echo örnek programının tasarımı

Program tetiklendiğinde, MQCONN çağrısı kullanılarak varsayılan kuyruk yöneticisine belirtik olarak bağlanır. Bu, IBM için gerekli olmasa da, kaynak kodu değiştirmeden aynı programı diğer platformlarda da kullanabilirsiniz.

Program, başlatılırken geçirildiği tetikleyici ileti yapısında adı belirtilen kuyruğu açar. (Daha anlaşılır olması için, buna *istek kuyruğudiyoruz*.) Program, paylaşılan giriş için bu kuyruğu açmak üzere MQOPEN çağrılmasını kullanır.

Program, iletileri bu kuyruktan kaldırmak için MQGET çağrısını kullanır. Bu çağrı, 5 saniyelik bekleme aralığıyla GMATM ve GMWT seçeneklerini kullanır. Program, bir istek iletisi olup olmadığını görmek için her iletinin tanımlayıcısını sınar; değilse, program iletiyi atar ve bir uyarı iletisi görüntüler.

İstek kuyruğundan kaldırılan her istek iletisi için, program yanıt kuyruğuna bir yanıt iletisi koymak için MQPUT çağrısını kullanır. Bu ileti, istek iletisinin içeriğini içerir.

İstek kuyruğunda kalan ileti yoksa, program o kuyruğu kapatır ve kuyruk yöneticisiyle olan bağlantısını keser.

Bu program, IBM dışındaki platformlardan kuyruğa gönderilen iletilere de yanıt verebilir, ancak bu durum için örnek sağlanmaz. ECHO programının çalışmasını sağlamak için:

- Metin isteği iletileri göndermek için *Format*, *Encoding* ve *CCSID* alanlarını doğru şekilde belirterek bir program yazın.

ECHO programı, gerekiyorsa, kuyruk yöneticisinden ileti verilerini dönüştürmesini ister.

- Yazdığınız program yanıt için benzer bir dönüştürme sağlamıyorsa, IBM MQ for IBM i gönderme kanalında CONVERT (*YES) belirleyin.

IBM i üzerinde Inquire örnek programı

AMQ3INQ4adlı Inquire örnek programı, MQINQ çağrısının kullanıldığı bir kuyruğun bazı özneliklerini sorar.

Programın tetiklenen bir program olarak çalışması amaçlanmıştır; bu nedenle, programın tek girişi MQTMC (tetikleyici ileti) yapısıdır. Bu yapı, sorulacak özneliklere sahip bir hedef kuyruğun adını içerir.

Tetikleme işleminin çalışması için, SYSTEM.SAMPLE.INQ. Ao yapmak için, SYSTEM.SAMPLE.INQPROCESS süreç tanımlamasının *AppLId* alanında Inquire örnek programının adını belirtin. (Bunun için, [Change MQ Process \(CHGMQMPRC\)](#) başlıklı konuda açıklanan CHGMQMPRC komutunu kullanabilirsiniz.) Örnek kuyruk FIRST tetikleyici tipine sahiptir; bu nedenle, İstek örneğini çalıştırmadan önce kuyrukta önceden ileti varsa, gönderdiğiniz iletiler tarafından sorgu örneği tetiklenmez.

Tanımı doğru olarak ayarladığınızda, önce bir işte AMQ3SRV4 'ü başlatın, daha sonra başka bir işte AMQ3REQ4 ' ü başlatın. AMQ3SRV4yerine AMQ3TRG4 kullanabilirsiniz, ancak olası iş gönderimi gecikmeleri, olanların izlenmesini daha az kolaylaştıracaktır.

İstek örnek programını, her biri yalnızca bir kuyruk adı içeren istek iletilerini SYSTEM.SAMPLE.INQ. Her istek ileti için, Inquire örnek programı, istek iletilisinde belirlenen kuyruğa ilişkin bilgileri içeren bir yanıt ileti gönderir. Yanıtlar, istek iletilisinde belirlenen yanıt kuyruğuna gönderilir.

Inquire örnek programının tasarımı

Program tetiklendiğinde, MQCONN çağrısı kullanılarak varsayılan kuyruk yöneticisine belirtir olarak bağlanır. IBM üzerinde gerekli olmasa da, bu tasarım özelliği, kaynak kodu değiştirmeden aynı programı diğer platformlarda da kullanabileceğiniz anlamına gelir.

Program, başlatılırken geçirildiği tetikleyici ileti yapısında adı belirtilen kuyruğu açar. (Daha anlaşılır olması için, buna *istek kuyruğudiyoruz*.) Program, paylaşılan giriş için bu kuyruğu açmak üzere MQOPEN çağrılmasını kullanır.

Program, iletileri bu kuyruktan kaldırmak için MQGET çağrısını kullanır. Bu çağrı, 5 saniyelik bekleme aralığıyla GMATM ve GMWT seçeneklerini kullanır. Program, bir istek ileti olup olmadığını görmek için her iletinin tanımlayıcısını sınar; değilse, program iletiyi atar ve bir uyarı ileti görüntüler.

İstek kuyruğundan kaldırılan her istek ileti için, program kuyruğun adını okur (*hedef kuyruk* olarak adlandırılacaktır). verilerde bulunur ve OOINQ seçeneğiyle MQOPEN çağrısıyla o kuyruğu açar. Bu program daha sonra, hedef kuyruğun **InhibitGet**, **CurrentQDepth** ve **OpenInputCount** özneliklerinin değerlerini sorgulamak için MQINQ çağrılarını kullanır.

MQINQ çağrısı başarılı olursa, program yanıt kuyruğuna bir yanıt ileti koymak için MQPUT çağrısını kullanır. Bu ileti, üç özneliğin değerlerini içerir.

MQOPEN ya da MQINQ çağrısı başarısız olursa, program yanıt kuyruğuna bir *rapor* ileti koymak için MQPUT çağrısını kullanır. Bu rapor iletilisinin ileti tanımlayıcısının *MDFB* alanında, hangisinin başarısız olduğuna bağlı olarak MQOPEN ya da MQINQ çağrısının döndürdüğü neden kodu yer alır.

MQINQ çağrısından sonra, program MQCLOSE çağrısını kullanarak hedef kuyruğu kapatır.

İstek kuyruğunda kalan ileti yoksa, program o kuyruğu kapatır ve kuyruk yöneticisiyle olan bağlantısını keser.

IBM i üzerinde Set sample programı

Set sample programı (AMQ3SET4), kuyruğun **InhibitPut** özneliğini değiştirmek için MQSET çağrısıyla kuyruğa koyma işlemlerini engeller.

Programın tetiklenen bir program olarak çalışması amaçlanmıştır; bu programın tek girişi, istenen öznelikleri olan bir hedef kuyruğun adını içeren bir MQTMC (tetikleyici iletisi) yapısıdır.

Tetikleme işleminin çalışması için, Set sample programının SYSTEM.SAMPLE.SETkuyruğuna gelen ileteler tarafından tetiklendiğinden emin olmanız gerekir. Bunu yapmak için, SYSTEM.SAMPLE.SETPROCESSsüreç tanımlamasının *AppLI*d alanında Set sample programının adını belirtin. (Bunun için, Yönetme IBM i başlıklı konuda açıklanan CHGMQMPRC komutunu kullanabilirsiniz.) Örnek kuyruğun tetikleyici tipi FIRST olduğundan, İstek örneğini çalıştırmadan önce kuyrukta önceden ileti varsa, Gönderdiğiniz ileteler Set örneğini tetiklemez.

Tanımı doğru olarak ayarladığınızda, önce bir işte AMQ3SRV4 'ü başlatın, daha sonra başka bir işte AMQ3REQ4 ' ü başlatın. AMQ3SRV4yerine AMQ3TRG4 kullanabilirsiniz, ancak olası iş gönderimi gecikmeleri, olan biteni izlemeyi daha az kolaylaştıracaktır.

Her biri yalnızca bir kuyruk adı içeren istek iletelerini SYSTEM.SAMPLE.SETkuyruğuna göndermek için İstek örnek programını kullanın. Her istek iletisi için, Set sample programı, belirlenen kuyrukta koyma işlemlerinin engellendiğini bildiren bir doğrulama içeren bir yanıt iletisi gönderir. Yanıtlar, istek iletisinde belirlenen yanıt kuyruğuna gönderilir.

Set örnek programının tasarımı

Program tetiklendiğinde, MQCONN çağrısı kullanılarak varsayılan kuyruk yöneticisine belirttik olarak bağlanır. IBM i üzerinde gerekli olmasa da, kaynak kodu değiştirmeden aynı programı diğer platformlarda da kullanabilirsiniz.

Program, başlatılırken geçirildiği tetikleyici ileti yapısında adı belirtilen kuyruğu açar. (Daha anlaşılır olması için, buna *istek kuyruğ*udiyoruz.) Program, paylaşılan giriş için bu kuyruğu açmak üzere MQOPEN çağrılmasını kullanır.

Program, ileteleri bu kuyruktan kaldırmak için MQGET çağrısını kullanır. Bu çağrı, 5 saniyelik bekleme aralığıyla GMATM ve GMWT seçeneklerini kullanır. Program, bir istek iletisi olup olmadığını görmek için her iletinin tanımlayıcısını sınar; değilse, program iletiyi atar ve bir uyarı iletisi görüntüler.

İstek kuyruğundan kaldırılan her istek iletisi için, program kuyruğun adını okur (*hedef kuyruk* olarak adlandırılacaktır). verilerde yer alır ve OQSET seçeneğiyle MQOPEN çağrısıyla o kuyruğu açar. Daha sonra program, hedef kuyruğun **InhibitPut** özneliğinin değerini QAPUTI olarak ayarlamak için MQSET çağrısını kullanır.

MQSET çağrısı başarılı olursa, program yanıt kuyruğuna yanıt iletisi koymak için MQPUT çağrısını kullanır. Bu ileti PUT inhibiteddizgisini içerir.

MQOPEN ya da MQSET çağrısı başarısız olursa, program yanıt kuyruğuna *rapor* iletisi koymak için MQPUT çağrısını kullanır. Bu rapor iletisinin ileti tanımlayıcısının *MDFB* alanında, hangisinin başarısız olduğuna bağlı olarak MQOPEN ya da MQSET çağrısının döndürdüğü neden kodu bulunur.

MQSET çağrısından sonra, program MQCLOSE çağrısını kullanarak hedef kuyruğu kapatır.

İstek kuyruğunda kalan ileti yoksa, program o kuyruğu kapatır ve kuyruk yöneticisiyle olan bağlantısını keser.

IBM i üzerinde Triggering örnek programları

IBM MQ for IBM i , ILE/RPG dilinde yazılmış iki Triggering örnek programı sağlar.

Programlar şunlardır:

AMQ3TRG4

Bu, IBM i ortamı için bir tetikleyici izleme programıdır. Uygulamanın başlatılması için bir IBM i işi gönderir, ancak bu, her tetikleyici iletiyle ilişkili ek işleme maliyetinin olduğu anlamına gelir.

AMQ3SRV4

Bu, IBM i ortamı için bir tetikleyici sunucusudur. Bu sunucu, her tetikleme iletisi için, belirlenen uygulamayı başlatmak üzere kendi işinde başlatma komutunu çalıştırır. Tetikleyici sunucu CICS hareketlerini çağırabilir.

Bu örneklerin C dili sürümleri, AMQSTRG4 ve AMQSERV4adlı QMQM kitaplığında yürütülebilir programlar olarak da kullanılabilir.

IBM i üzerinde AMQ3TRG4 örnek tetikleyici izleme programı

AMQ3TRG4 , bir tetikleyici izleyicidir. Bir parametre gerekir: hizmet vereceği başlatma kuyruğunun adı. AMQSAMP4 , örnek bir başlatma kuyruğunu (SYSTEM.SAMPLE.TRIGGERtetikleyicisi, örnek programları denerken kullanabileceğiniz bir tetikleyicidir.

AMQ3TRG4 , başlatma kuyruğundan aldığı her geçerli tetikleyici iletisi için bir IBM i işi gönderir.

Tetikleyici izleme programının tasarımı

Tetikleyici izleme programı başlatma kuyruğunu açar ve sınırsız bekleme aralığı belirterek kuyruktan ileti alır.

Tetikleyici izleme programı, tetikleyici iletisinde belirtilen uygulamayı başlatmak için bir IBM i işi sunar ve MQTMC (tetikleyici iletisinin karakter sürümü) yapısını geçirir. Tetikleyici iletideki ortam verileri, iş gönderimi parametreleri olarak kullanılır.

Son olarak, program başlatma kuyruğunu kapatır.

AMQ3SRV4 örnek tetikleyici sunucusu

AMQ3SRV4 bir tetikleyici sunucusudur. Bir parametre gerekir: hizmet vereceği başlatma kuyruğunun adı. AMQSAMP4 , örnek bir başlatma kuyruğunu (SYSTEM.SAMPLE.TRIGGERtetikleyicisi, örnek programları denerken kullanabileceğiniz bir tetikleyicidir.

Her tetikleyici iletisi için AMQ3SRV4 , belirtilen uygulamayı başlatmak için kendi işinde bir başlatma komutu çalıştırır.

Örnek tetikleyici kuyruğunun kullanılması, verilecek komutun aşağıdaki gibidir:

```
CALL PGM(QMQM/AMQ3SRV4) PARM('Queue Name')
```

Burada Queue Name , gerekli sayıda boşlukla kuyruk adını doldurarak ulaşabileceğiniz 48 karakter uzunluğunda olmalıdır. Bu nedenle, SYSTEM.SAMPLE.TRIGGER hedef kuyruğunuz olarak 28 boş karaktere gereksinim duyarsınız.

Tetikleyici sunucusunun tasarımı

Tetikleyici sunucusunun tasarımı, tetikleyici sunucusu dışında, tetikleyici izleme programının tasarımına benzer:

- CICS ve IBM i uygulamalarına izin verir
- Tetikleyici iletisindeki ortam verilerini kullanmaz
- Bir IBM i işi göndermek yerine kendi işindeki IBM i uygulamalarını çağırır (ya da CICS uygulamalarını başlatmak için STRCICSUSR kullanır)
- Paylaşılan giriş için başlatma kuyruğunu açar; böylece birçok tetikleyici sunucusu aynı anda çalışabilir

Not: AMQ3SRV4 tarafından başlatılan programlar, tetikleme sunucusunu durduracağından MQDISC çağırısı kullanmamalıdır. AMQ3SRV4 tarafından başlatılan programlar MQCONN çağırısı kullanıyorsa, RC2002 neden kodunu alır.

IBM i üzerinde Triggering örnek programlarının sonlandırılması

Tetikleyici izleme programı, sistem isteği seçeneği 2 (ENDRQS) ya da tetikleyici kuyruğundan alma işlemini engelleyerek sona erdirilebilir.

Örnek tetikleyici kuyruğu kullanılırsa, komut şöyle olur:

```
CHGMQM QNAME('SYSTEM.SAMPLE.TRIGGER') GETENBL(*NO)
```

Not: Bu kuyruğa yeniden tetikleme işlemini başlatmak için şu komutu girmeniz gerekir:

```
CHGMQM QNAME('SYSTEM.SAMPLE.TRIGGER') GETENBL(*YES)
```

IBM i üzerinde uzak kuyruklar kullanılarak örneklerin çalıştırılması

Bağlı ileti kuyruğu yöneticilerindeki örnekleri çalıştırarak uzak kuyruğa alma gösterebilirsiniz.

AMQSAMP4 programı, OTHERadlı uzak kuyruk yöneticisini kullanan uzak kuyruğun (SYSTEM.SAMPLE.REMOTE) yerel tanımlamasını sağlar. Bu örnek tanımlamayı kullanmak için OTHER değerini, kullanmak istediğiniz ikinci ileti kuyruğu yöneticisinin adına çevirin. Ayrıca, iki ileti kuyruğu yöneticiniz arasında bir ileti kanalı oluşturmanız gerekir; bunun nasıl gerçekleştirileceğine ilişkin bilgi için [İleti kanalları için kanal çıkış programları](#) başlıklı konuya bakın.

İstek örnek programı, gönderdiği iletilerin MDRM alanına kendi yerel kuyruk yöneticisi adını koyar. Sorgu ve Küme örnekleri, işledikleri istek iletilerinin MDRQ ve MDRM alanlarında adı belirtilen kuyruk ve ileti kuyruğu yöneticisine yanıt iletileri gönderir.

IBM i (ILE RPG) için dönüş kodları

Bu bilgiler, MQI ve MQAI ile ilişkili dönüş kodlarını açıklar.

Şununla ilişkili dönüş kodları:

- Programlanabilir Komut Biçimi (PCF) komutları [Programlanabilir komut biçimleri başvurusu](#) içinde listelenir.
- C++ çağrılarını C++ kullanarak içinde listelenir.

Her çağrı için, bir tamamlanma kodu ve neden kodu kuyruk yöneticisi ya da bir çıkış yordamı tarafından, aramanın başarılı ya da başarısız olduğunu belirtmek için döndürülür.

Uygulamalar, özellikle belirtildiği durumlar dışında, belirli bir sırada denetlenen hatalara bağlı olmamalıdır. Bir çağrıdan birden fazla tamamlanma kodu ya da neden kodu oluşabiliyorsa, bildirilen hata uygulamaya bağlıdır.

IBM i (ILE RPG) için tamamlama kodları

Tamamlama kodu parametresi (CMPCOD), çağırının, çağırının başarıyla tamamlanıp tamamlanmadığını, kısmen tamamlandığını ya da başarısız olup olmadığını hızlı bir şekilde görmesini sağlar.

CCOK

(Diğer altyapılarda MQCC_OK)

Başarıyla tamamlandı.

Çağrı tam olarak tamamlandı; tüm çıkış parametreleri ayarlandı. Bu durumda **REASON** parametresi her zaman RCNONE değerine sahiptir.

CCWARN

(Diğer altyapılarda MQCC_WARN)

Uyarı (kısmi tamamlama).

Arama kısmen tamamlandı. CMPCOD ve REASON çıkış parametrelerine ek olarak bazı çıkış parametreleri ayarlanmış olabilir. **REASON** parametresi, kısmi tamamlama hakkında ek bilgi verir.

CCFAIL

(Diğer altyapılarda MQCC_FAIL)

Arama başarısız oldu.

Çağrıyı işleme tamamlanmadı ve kuyruk yöneticisinin durumu olağan durumda değişmedi; kural dışı durumlar özel olarak belirtildi. *CMPCOD* ve *REASON* çıkış parametreleri ayarlandı; belirtilenler dışında diğer parametreler değiştirilmez.

Bunun nedeni, uygulama programındaki bir hata olabilir ya da programın dışındaki bir durumun (örneğin, kullanıcının yetkisi iptal edilmiş olabilir) sonucu olabilir. **REASON** parametresi, hatayla ilgili ek bilgi verir.

IBM i (ILE RPG) için neden kodları

Neden kodu parametresi (*REASON*), tamamlanma kodu parametresi (*CMPCOD*) için bir nitelik.

Raporlamak için özel bir neden yoksa, RCNONE döndürülür. Başarılı bir çağrı CCOK ve RCNONE değerini döndürür.

Tamamlanma kodu CCWARN ya da CCFAIL ise, kuyruk yöneticisi her zaman uygun bir neden bildirir; ayrıntılar her çağrı tanımı altında verilir.

Kullanıcı çıkışı yordamlarının tamamlanma kodlarını ve nedenlerini belirlediği durumlarda, bu kurallara uymaları gerekir. Ayrıca, kullanıcı çıkışları tarafından tanımlanan özel neden değerlerinin, kuyruk yöneticisi tarafından tanımlanan değerlerle çakışmamasını sağlamak için sıfırdan küçük olması gerekir. Çıkışlar, kuyruk yöneticisi tarafından tanımlanmış nedenleri belirleyebilir; bu nedenler uygun olabilir.

Neden kodları aşağıdaki durumlarda da oluşur:

- MQDLH yapısının *DLREA* alanı
- MQMD yapısının *MDFB* alanı

Neden kodlarının tam listesi için bkz. [API tamamlama ve neden kodları](#).

Bu listede IBM i neden kodunuzu bulmak için "RC" yi önden kaldırın; örneğin, RC2002 2002olur. Ayrıca, diğer platformlarda olduğu gibi gösterilen tamamlama kodları da vardır:

Çizelge 814. IBM i ve diğer altyapılarda neden kodu adları	
IBM i	Diğer platformlar
CCOK	MQCC_OK
CCWARN	MQCC_UYARISI
CCFAIL	MQCC_FAIL

IBM i (ILE RPG) için MQI seçeneklerini doğrulama kuralları

Bu konuda, bir MQOPEN, MQPUT, MQPUT1, MQGET ya da MQCLOSE çağrısından RC2046 neden kodu üreten durumlarla ilgili bilgi verilir.

IBM i üzerinde MQOPEN çağrısı

MQOPEN çağrısının seçenekleri için:

- Aşağıdaki öğelerden en az biri belirtilmelidir:
 - OOBROW
 - OOINPQ
 - OOINPX
 - OOINPS
 - OOINQ
 - OOOOUT!

- OOSSET!
- Aşağıdakilerden yalnızca *birine* izin verilir:
 - OOINPQ
 - OOINPX
 - OOINPS
- Aşağıdakilerden yalnızca *birine* izin verilir:
 - UUUUUIM
 - OOBNDN
 - OOBNDQ

Not: Daha önce listelenen seçenekler karşılıklı olarak birbirini dışlar. Ancak, OOBNDQ değeri sıfır olduğu için, bu değerin diğer iki bağ tanımlama seçeneğiyle birlikte belirtilmesi RC2046neden koduyla sonuçlanmıyor. Program belgelerine yardımcı olmak için OOBNDQ sağlanmıştır.

- OOSAVA belirtilirse, OOINP* seçeneklerinden biri de belirtilmelidir.
- OOSSET* ya da OOPAS* seçeneklerinden biri belirtilirse, OOSSET da belirtilmelidir.

IBM i üzerinde MQPUT çağırısı

Koyma iletisi seçenekleri için:

- PMSYP ve PMNSYP birleşimi kullanılamaz.
- Aşağıdakilerden yalnızca *birine* izin verilir:
 - PMDEFK
 - PMNOC
 - PMPASA
 - PMPASI
 - PMSETA
 - PMSETI
- PMALTU ' ya izin verilmez (yalnızca MQPUT1 çağırısında geçerlidir).

MQPUT1 çağırısı IBM i

Koyma iletisi seçenekleri için, aşağıdaki seçenekler dışında, kurallar MQPUT çağırısıyla aynıdır:

- PMALTU ' ya izin verilir.
- PMLOGO kullanılamaz.

IBM i üzerinde MQGET çağırısı

İleti alma seçenekleri için:

- Aşağıdaki seçeneklerden yalnızca *birine* izin verilir:
 - GMNSYP
 - GMSYP
 - GMPSYP
- Aşağıdaki seçeneklerden yalnızca *birine* izin verilir:
 - GMBRWF
 - GMBRWC
 - GMBRWN

- GMMUC
- GMSYP ' ye aşağıdaki seçeneklerden biriyle izin verilmez:
 - GMBRWF
 - GMBRWC
 - GMBRWN
 - GMLK
 - GMUNLK
- GMPSTYP ' ye aşağıdaki seçeneklerden biriyle izin verilmez:
 - GMBRWF
 - GMBRWC
 - GMBRWN
 - GMCMPM
 - GMUNLK
- GMLK belirtilirse, aşağıdaki seçeneklerden biri de belirtilmelidir:
 - GMBRWF
 - GMBRWC
 - GMBRWN
- GMUNLK belirtilirse, yalnızca aşağıdaki seçeneklere izin verilir:
 - GMNSYP
 - GMNWT

IBM i üzerinde MQCLOSE çağrısı

- MQCLOSE çağrısının seçenekleri için. CODEL ve COPURG birleşimi kullanılamaz.
- Aşağıdakilerden yalnızca birine izin verilir:
 - COKPSB
 - KORMSB

IBM i üzerinde MQSUB çağrısı

MQSUB çağrısının seçenekleri için:

- Aşağıdakilerden en az biri belirtilmelidir:
- Aşağıdakilerden en az biri belirtilmelidir:
 - SOALT
 - SıRALAR
 - SOCRT
- Aşağıdakilerden yalnızca birine izin verilir:
 - SODUR.
 - SONDUR.

Not: Daha önce listelenen seçenekler karşılıklı olarak birbirini dışlar. Ancak, SONDUR değeri sıfır olduğundan, SODUR ile belirtilmesi RC2046neden koduyla sonuçlanmaz. SONDUR, program belgelerine yardımcı olmak için sağlanır.

- SOGRP ve SOMAN birleşimine izin verilmez.
- SOGRP, SOSCID ' nin belirtilmesini gerektirir.
- Şu öğelerden yalnızca birine izin verilir: SOAUID SOFUID

- SONEWP ve SOPUBR birleşimine izin verilmez.
- SONEWP ' ye yalnızca SOCRT ile birlikte izin verilir.
- Aşağıdakilerden yalnızca birine izin verilir:
 - SOWCHR
 - SOWTOP

IBM i üzerinde makine kodlamaları

İleti tanımlayıcıdaki *MDENC* alanının yapısı hakkında bilgi edinmek için bu bilgileri kullanın.

İleti tanımlayıcı hakkında daha fazla bilgi için bkz. [“IBM i üzerinde MQMD \(İleti tanımlayıcı\)” sayfa 1093.](#)

MDENC alanı, dört ayrı alt alana bölünmüş 32 bitlik bir tamsayıdır; bu alt alanlar şunları tanımlar:

- İkili tamsayılar için kullanılan kodlama
- Paketlenmiş ondalık tamsayılar için kullanılan kodlama
- Kayan noktalı sayılar için kullanılan kodlama
- Ayrılmış bitler

Her alt alan, alt alana karşılık gelen konumlarda 1 bit ve başka bir yerde 0 bit olan bir bit maskesiyle tanımlanır. Bitler 0 biti en önemli biti, 31 biti en az anlamlı biti olacak şekilde numaralandırılır. Aşağıdaki maskeler tanımlandı:

TRIMSK

İkili tamsayı kodlaması için maske.

Bu alt alan, *MDENC* alanında 28-31 bit konumlarını kaplar.

ENDMSK

Paketlenmiş ondalık tamsayı kodlaması için maske.

Bu alt alan, *MDENC* alanında 24-27 bit konumlarını kaplar.

ENFMSK

Kayan nokta kodlaması için maske.

Bu alt alan, *MDENC* alanında 20-23 bit konumlarını kaplar.

ENRMSK

Ayrılmış bitler için maske.

Bu alt alan, *MDENC* alanında 0 ile 19 arasındaki bit konumlarını kaplar.

IBM i IBM i üzerinde ikili tamsayı kodlaması

İkili tamsayı kodlaması için geçerli değerler.

Aşağıdaki değerler ikili tamsayı kodlaması için geçerlidir:

SıKışTIRMA

Tanımlanmamış tamsayı kodlaması.

İkili tamsayılar, tanımlanmamış bir kodlama kullanılarak gösterilir.

GİRİŞ

Normal tamsayı kodlaması.

İkili tamsayılar geleneksel şekilde gösterilir:

- Sayıdaki en az anlamlı bayt, sayıdaki herhangi bir baytın en yüksek adresine sahiptir; en önemli bayt en düşük adrese sahiptir.
- Her bayttaki en az anlamlı bit, bir sonraki daha yüksek adresli baytın yanında bulunur; her bayttaki en anlamlı bit, bir sonraki alt adresli baytın yanında bulunur.

ENIREV

Tamsayı kodlaması tersine çevrildi.

İkili tamsayılar ENINOR ile aynı şekilde gösterilir, ancak byte 'lar ters sırayla düzenlenir. Her byte içindeki bitler ENINOR ile aynı şekilde düzenlenir.

IBM i

IBM i üzerinde paketlenmiş onlu tamsayı kodlaması

Paketlenmiş ondalık tamsayı kodlaması için geçerli değerler

Paketlenmiş onlu tamsayı kodlaması için aşağıdaki değerler geçerlidir:

BITIR

Tanımlanmamış paketlenmiş onlu kodlama.

Paketlenmiş ondalık tamsayılar, tanımlanmamış bir kodlama kullanılarak gösterilir.

SONA ERDIR

Olağan paketlenmiş onlu kodlama.

Paketlenmiş ondalık tamsayılar geleneksel olarak gösterilir:

- Sayının yazdırılabilir biçimindeki her ondalık basamak, X' 0 '-X' 9' aralığındaki tek bir onaltılı sayıyla paketlenmiş onlu olarak gösterilir. Her onaltılı sayı 4 bit kaplar ve bu nedenle paketlenmiş onlu sayıdaki her bayt, sayının yazdırılabilir biçiminde iki ondalık basamağı temsil eder.
- Paketlenmiş onlu sayıdaki en az anlamlı bayt, en az anlamlı ondalık basamağı içeren bayttır. Bu bayt içinde, en anlamlı 4 bit en az anlamlı ondalık basamağı içerir ve en az anlamlı 4 bit işareti içerir. İşaret X'C '(pozitif), X 'D' (negatif) ya da X'F ' (işaretsiz).
- Sayıdaki en az anlamlı bayt, sayıdaki herhangi bir baytın en yüksek adresine sahiptir; en önemli bayt en düşük adrese sahiptir.
- Her bayttaki en az anlamlı bit, bir sonraki daha yüksek adresli baytın yanında bulunur; her bayttaki en anlamlı bit, bir sonraki alt adresli baytın yanında bulunur.

ENDREV

Paketlenmiş onlu kodlama tersine çevrildi.

Paketlenmiş ondalık tamsayılar, ENDNOR ile aynı şekilde, ancak byte 'lar ters sırada düzenlenmiş olarak gösterilir. Her bayttaki bitler, ENDNOR ile aynı şekilde düzenlenir.

IBM i

IBM i üzerinde kayan nokta kodlaması

Kayan nokta kodlaması için geçerli değerler

Kayan nokta kodlaması için aşağıdaki değerler geçerlidir:

KUŞKU

Tanımlanmamış kayan nokta kodlaması.

Kayan noktalı sayılar, tanımlanmamış bir kodlama kullanılarak gösterilir.

HAKNE

Normal IEEE (Elektrik ve Elektronik Mühendisleri Enstitüsü) float kodlaması.

Kayan noktalı sayılar, standart IEEE kayan nokta biçimi kullanılarak aşağıdaki gibi düzenlenmiş olarak gösterilir:

- Mantissa 'daki en az anlamlı bayt, sayıdaki herhangi bir baytın en yüksek adresine sahiptir; üssü içeren bayt en düşük adrese sahiptir
- Her bayttaki en az anlamlı bit, bir sonraki daha yüksek adresli baytın yanında bulunur; her bayttaki en anlamlı bit, bir sonraki alt adresli baytın yanında bulunur.

IEEE float kodlamasının ayrıntıları IEEE Standard 754 'te bulunabilir.

ENFREV

IEEE kayan noktalı sayı kodlaması tersine çevrildi.

Kayan noktalı sayılar ENFNOR ile aynı şekilde temsil edilir, ancak baytlar ters sırayla düzenlenir. Her bayttaki bitler ENFNOR ile aynı şekilde düzenlenir.

ENF390

System/390 mimarisi kayan noktalı sayı kodlaması.

Kayan noktalı sayılar standart System/390 kayan noktalı biçimi kullanılarak gösterilir; bu, System/370 tarafından da kullanılır.

IBM i IBM i üzerinde kodlamalar oluşturma

MQMD 'de MDENC alanı için bir değer oluşturmak üzere gerekli kodlamaları açıklayan ilgili sabitler eklenmelidir.

ENI* kodlamalarında yalnızca birini END* kodlamalarında ve ENF* kodlamalarında biriyle birleştirdiğinizden emin olun.

IBM i IBM i üzerinde kodlamaları çözme

MDENC alanı alt alanlar içerir; bu nedenle, tamsayı, paketlenmiş ondalık ya da kayan noktalı sayı kodlamasını incelemesi gereken uygulamalar bu konuda açıklanan tekniği kullanmalıdır.

Aritmetik kullanılıyor

Tamsayı aritmetiği kullanılarak aşağıdaki adımlar gerçekleştirilmelidir:

1. Gereken kodlama tipine göre aşağıdaki değerlerden birini seçin:

- İkili tamsayı kodlaması için 1
- Paketlenmiş ondalık tamsayı kodlaması için 16
- Kayan nokta kodlaması için 256

Adeğerini çağırın.

2. MDENC alanının değerini A ile bölün; Bsonucunu çağırın.

3. B ögesini 16 'ya bölün; sonucu C olarak çağırın.

4. C 'i 16 ile çarparak B ' dan çıkarın; Dsonucunu çağırın.

5. D ' i A ile çarpma; Esonucunu çağırın.

6. E gerekli kodlamadır ve bu kodlama tipi için geçerli olan değerlerin her biriyle eşitlik için test edilebilir.

IBM i IBM i üzerindeki makine mimarisi kodlamasının özeti

Makine mimarileri için kodlamaları özetleyen bir tablo.

Makine mimarilerine ilişkin kodlamalar Çizelge 815 sayfa 1408 içinde gösterilmektedir.

Çizelge 815. Makine mimarilerine ilişkin kodlamalar özeti			
Makine mimarisi	İkili tamsayı kodlaması	Paketli ondalık tamsayı kodlaması	Kayan nokta kodlaması
IBM i	normal	normal	IEEE normal
Intel x86	Tersine Çevrilmiş	Tersine Çevrilmiş	IEEE tersine çevrildi
PowerPC	normal	normal	IEEE normal
System/390	normal	normal	System/390

Bu konu, MQGET, MQPUT ve MQPUT1 çağrılarında belirtilen ileti tanımlayıcısı MQMD ' nin bir parçası olan MDREP ve MDMFL alanlarını ilgilendirir.

İleti tanımlayıcı hakkında daha fazla bilgi için bkz. [“IBM i üzerinde MQMD \(İleti tanımlayıcı\)” sayfa 1093](#). Bu bilgiler aşağıdakileri açıklar:

- Rapor alanının yapısı ve kuyruk yöneticisinin bunu nasıl işlediği
- Bir uygulamanın rapor alanını nasıl analiz etmesi gerekir?
- İleti işaretleri alanının yapısı

Rapor alanının yapısı

MDREP alanı, üç ayrı alt alana bölünmüş 32 bitlik bir tamsayıdır.

Bu alt alanlar aşağıdakileri tanımlar:

- Yerel kuyruk yöneticisi bunları tanımazsa reddedilen rapor seçenekleri
- Yerel kuyruk yöneticisi bunları tanımasa da her zaman kabul edilen rapor seçenekleri
- Yalnızca diğer bazı koşullar karşılandıysa kabul edilen rapor seçenekleri

Her alt alan, alt alana karşılık gelen konumlarda 1 bit ve başka bir yerde 0 bit olan bir bit maskesiyle tanımlanır. Bir alt alandaki bitlerin bitişik olması gerekmediğini unutmayın. Bitler 0 biti en önemli biti, 31 biti en az anlamlı biti olacak şekilde numaralandırılır. Alt alanları tanımlamak için aşağıdaki maskeler tanımlanır:

RORUM

Reddedilen desteklenmeyen rapor seçenekleri için maske.

Bu maske, yerel kuyruk yöneticisi tarafından desteklenmeyen rapor seçeneklerinin MQPUT ya da MQPUT1 çağrısına CCFAIL tamamlanma kodu ve RC2061neden koduyla başarısız olmasına neden olduğu MDREP alanındaki bit konumlarını tanımlar.

Bu alt alan 3 ve 11 ile 13 arasındaki bit konumlarını kaplar.

ROAUM

Kabul edilen desteklenmeyen rapor seçenekleri için maske.

Bu maske, MDREP alanında, yerel kuyruk yöneticisi tarafından desteklenmeyen rapor seçeneklerinin yine de MQPUT ya da MQPUT1 çağrılarında kabul edileceği bit konumlarını tanımlar. Bu durumda, RC2104 neden koduyla tamamlama kodu CCWARN döndürülür.

Bu alt alan 0-2, 4-10 ve 24-31 arasındaki bit konumlarını kaplar.

Bu alt alana aşağıdaki rapor seçenekleri eklenmiştir:

- ROCMTC
- RODLQ
- RODISC
- SATIR
- ROEXCD
- ROEXCF
- SATIR
- YADIR
- ROEXPF
- RONAN
- RONMI
- YAYINI

- ROPAN
- ROPCI
- ROPMI

ROAUXM

Yalnızca belirli koşullarda kabul edilen desteklenmeyen rapor seçenekleri için maske.

Bu maske, *MDREP* alanındaki bit konumlarını tanıtır. Burada, yerel kuyruk yöneticisi tarafından desteklenmeyen rapor seçenekleri yine de MQPUT ya da MQPUT1 çağrılarında *kabul edilir* ve her iki koşul da karşılanır:

- İleti uzak bir kuyruk yöneticisine gönderiliyor.
- Uygulama iletiyi doğrudan yerel bir iletim kuyruğuna koymuyor (yani, MQOPEN ya da MQPUT1 çağrısında belirtilen nesne tanımlayıcısındaki *ODMN* ve *ODON* alanları tarafından tanımlanan kuyruk yerel bir iletim kuyruğu değil).

Bu koşullar yerine getirilirse, RC2104 neden koduyla tamamlama kodu CCWARN döndürülür; gelmezse, CCFAIL RC2061 neden koduyla döndürülür.

Bu alt alan, 14 ile 23 arasındaki bit konumlarını kaplar.

Bu alt alana aşağıdaki rapor seçenekleri eklenmiştir:

- ROCOA
- ROCOAD
- ROCOAF
- ROCOD
- ROCODD
- ROCODF

MDREP alanında kuyruk yöneticisinin tanımadığı seçenekler belirtilirse, kuyruk yöneticisi, *MDREP* alanını o alt alana ilişkin maskeyle birleştirmek için bit temelinde AND (VE) işlemi kullanarak her alt alanı denetler. Bu işlemin sonucu sıfır değilse, daha önce açıklanan tamamlanma kodu ve neden kodları döndürülür.

CCWARN döndürülürse, başka uyarı koşulları varsa hangi neden kodunun döndürüleceği tanımlanmaz.

Yerel kuyruk yöneticisi tarafından tanınmayan rapor seçeneklerini belirleme ve kabul etme yeteneği, *uzak* bir kuyruk yöneticisi tarafından tanınacak ve işlenecek bir rapor seçeneğiyle bir ileti göndermek gerektiğinde yararlıdır.

IBM i üzerinde rapor alanını analiz etme

MDREP alanı alt alanlar içerir. Bu nedenle, bazı uygulamaların, iletiyi gönderenin belirli bir rapor isteyip istemediğini denetlemesi gerekir. Bu uygulamalar bu konuda açıklanan tekniği kullanmalıdır.

Aritmetik kullanılıyor

Tamsayı aritmetiği kullanılarak aşağıdaki adımlar gerçekleştirilmelidir:

1. Denetlenecek rapor tipine göre aşağıdaki değerlerden birini seçin:
 - COA raporu için ROCOA
 - COD raporu için ROCOD
 - Kural dışı durum raporu için ROEXC
 - Süre sonu raporu için ROEXPAdeğerini çağırın.
2. *MDREP* alanını A ile bölün; Bsonucunu çağırın.
3. B ögesini 8 ' e bölün; Csonucunu çağırın.

4. C ögesini 8 ile çarparak B içinden çıkarın; Dsonucunu çağırın.
5. D ' i A ile çarpma; Esonucunu çağırın.
6. Bu tip bir rapor için mümkün olan her bir değerle eşitlik için E ' i test edin.

Örneğin, A ROEXC ise, iletiyi gönderen tarafından ne belirtildiğini belirlemek için aşağıdakilerden her biriyle eşitlik için E sınavasını yapın:

- YAYINI
- SATIR
- ROEXCD
- ROEXCF

Testler, uygulama mantığı için en uygun sırayla gerçekleştirilebilir.

Aşağıdaki sözde kod, kural dışı durum raporu iletileri için bu tekniği gösterir:

```
A = ROEXC
B = Report/A
C = B/8
D = B - C*8
E = D*A
```

ROPMI ya da ROPCI seçeneklerini sınamak için benzer bir yöntem kullanılabilir; A Bu iki değişmezden hangisi uygunsa, değeri seçin ve daha sonra daha önce açıklandığı gibi devam edin, ancak önceki adımlarda 8 değerini 2değeriyle değiştirin.

IBM i

IBM i üzerindeki ileti işaretleri alanının yapısı

MDMFL alanı, üç ayrı alt alana bölünmüş 32 bitlik bir tamsayıdır.

Bu alt alanlar aşağıdakileri tanımlar:

- Yerel kuyruk yöneticisi bunları tanımazsa reddedilen ileti işaretleri
- Yerel kuyruk yöneticisi bunları tanımasa da, her zaman kabul edilen ileti işaretleri
- Yalnızca diğer bazı koşullar yerine getirildiğinde kabul edilen ileti işaretleri

Not: *MDMFL* içindeki tüm alt alanlar, kuyruk yöneticisi tarafından kullanılmak üzere ayrılmıştır.

Her alt alan, alt alana karşılık gelen konumlarda 1 bit ve başka bir yerde 0 bit olan bir bit maskesiyle tanımlanır. Bitler 0 biti en önemli biti, 31 biti en az anlamlı biti olacak şekilde numaralandırılır. Alt alanları tanımlamak için aşağıdaki maskeler tanımlanır:

MFRUM

Reddedilen desteklenmeyen ileti işaretleri için maske.

Bu maske, *MDMFL* alanındaki bit konumlarını tanımlar; burada, yerel kuyruk yöneticisi tarafından desteklenmeyen ileti işaretleri MQPUT ya da MQPUT1 çağırısına CCFAIL tamamlanma kodu ve RC2249neden olur.

Bu alt alan 20-31 bit konumlarını kaplar.

Bu alt alana aşağıdaki ileti işaretleri eklenmiştir:

- MFLMIG
- MFLSEG
- MFMIG
- MFSEG
- MFSEGA
- MFSEGI

MFAUM

Kabul edilen desteklenmeyen ileti işaretleri için maske.

Bu maske, *MDMFL* alanında yerel kuyruk yöneticisi tarafından desteklenmeyen ileti işaretlerinin yine de MQPUT ya da MQPUT1 çağrılarında kabul edileceği bit konumlarını tanıtır. Tamamlanma kodu: CCOK.

Bu alt alan 0-11 arasındaki bit konumlarını kaplar.

MFAUXM

Yalnızca belirli koşullarda kabul edilen desteklenmeyen ileti işaretleri için maske.

Bu maske, *MDMFL* alanında yerel kuyruk yöneticisi tarafından desteklenmeyen ileti işaretlerinin yine de MQPUT ya da MQPUT1 çağrılarında *kabul edileceği* ve her iki koşulun da karşılanacağı bit konumlarını tanımlar:

- İleti uzak bir kuyruk yöneticisine gönderiliyor.
- Uygulama iletiyi doğrudan yerel bir iletim kuyruğuna koymuyor (yani, MQOPEN ya da MQPUT1 çağrısında belirtilen nesne tanımlayıcısındaki *ODMN* ve *ODON* alanları tarafından tanımlanan kuyruk yerel bir iletim kuyruğu değil).

Bu koşullar karşılandıysa tamamlanma kodu CCOK, karşılanmazsa neden kodu RC2249 ile CCFAIL döndürülür.

Bu alt alan 12 ile 19 arasındaki bit konumlarını kaplar.

MDMFL alanında kuyruk yöneticisinin tanımadığı işaretler belirtilirse, kuyruk yöneticisi, *MDMFL* alanını o alt alana ilişkin maskeyle birleştirmek için bit olarak AND işlemini kullanarak her alt alanı denetler. Bu işlemin sonucu sıfır değilse, daha önce açıklanan tamamlanma kodu ve neden kodları döndürülür.

IBM i

IBM i üzerinde veri dönüştürme

Bu konuda, veri dönüştürme çıkışına ilişkin arabirim ve veri dönüştürme gerektiğinde kuyruk yöneticisi tarafından gerçekleştirilen işlem açıklanmaktadır.

Veri dönüştürme çıkışı, MQGET çağrısının işlenmesinin bir parçası olarak çağrılır. Uygulama iletileri verilerini, alan uygulamanın gerektirdiği gösterimine dönüştürmek için kullanılır. Uygulama iletileri verilerinin dönüştürülmesi isteğe bağlıdır ve MQGET çağrısında GMCONV seçeneğinin belirtilmesini gerektirir.

Veri dönüştürmenin aşağıdaki yönleri açıklanmıştır:

- Kuyruk yöneticisi tarafından GMCONV seçeneğine yanıt olarak gerçekleştirilen işleme; bkz. [“IBM i üzerinde dönüştürme işlemi” sayfa 1412.](#)
- Yerleşik bir biçim işlenirken kuyruk yöneticisi tarafından kullanılan işleme kuralları; bu kurallar, kullanıcı tarafından yazılan çıkışlar için de önerilir. Bkz. [“IBM i üzerinde işleme kuralları” sayfa 1414.](#)
- Rapor iletilerinin dönüştürülmesi için dikkat edilmesi gereken özel noktalar; bkz. [“IBM i üzerinde rapor iletilerinin dönüştürülmesi” sayfa 1417.](#)
- Veri dönüştürme çıkışına geçirilen parametreler; bkz. [“IBM i üzerinde MQCONVX \(Veri dönüştürme çıkışı\)” sayfa 1428.](#)
- Karakter verilerini farklı gösterimler arasında dönüştürmek için çıkıştan kullanılacak bir çağrı; bkz. [“IBM i üzerinde MQXCNCV \(karakterleri dönüştür\)” sayfa 1423.](#)
- Çıkışa özgü veri yapısı parametresi; bkz. [“IBM i üzerinde MQDXP \(Data-conversion exit parameter\)” sayfa 1418.](#)

IBM i

IBM i üzerinde dönüştürme işlemi

Bu bilgiler, GMCONV seçeneğine yanıt olarak kuyruk yöneticisi tarafından gerçekleştirilen işlemleri açıklar.

MQGET çağrısında GMCONV seçeneği belirtilirse ve uygulamaya döndürülecek bir ileti varsa, kuyruk yöneticisi aşağıdaki işlemleri gerçekleştirir:

1. Aşağıdakilerden biri ya da birkaçı doğruysa, dönüştürme gerekmez:

- İleti verileri, MQGET çağrısı yayınlayan uygulamanın gerektirdiği karakter kümesinde ve kodlamada zaten var. Uygulama, çağrısı yayınlamadan önce, MQGET çağrısının **MSGDSC** değiştirgesindeki *MDCSI* ve *MDENC* alanlarını gerekli değerlere ayarlamalıdır.
- İleti verilerinin uzunluğu sıfır.
- MQGET çağrısının **BUFFER** değiştirgesinin uzunluğu sıfır.

Bu durumlarda ileti, MQGET çağrısının verildiği uygulamaya dönüştürülmeden döndürülür; **MSGDSC** değiştirgesindeki *MDCSI* ve *MDENC* değerleri, iletideki denetim bilgilerindeki değerlere ayarlanır ve çağrı, aşağıdaki tamamlanma kodu ve neden kodu birleşimlerinden biriyle tamamlanır:

Tamamlanma kodu

Neden Kodu

CCOK

RCNONE

CCWARN

RC2079

CCWARN

RC2080

Aşağıdaki adımlar, ileti verilerinin karakter kümesi ya da kodlaması **MSGDSC** parametresindeki karşılık gelen değerden farklıysa ve dönüştürülecek veriler varsa gerçekleştirilir:

1. İletideki denetim bilgisindeki *MDFMT* alanı *FMNONE* değerine sahipse, ileti, tamamlanma kodu *CCWARN* ve neden kodu *RC2110* ile dönüştürülmez.

Diğer tüm durumlarda dönüştürme işlemi devam eder.

2. İleti kuyruktan kaldırılır ve **BUFFER** parametresiyle aynı büyüklükte olan geçici bir arabelleğe yerleştirilir. Göz atma işlemleri için, ileti kuyruktan kaldırılmak yerine geçici arabelleğe kopyalanır.

3. İletinin arabelleğe sığması için kesilmesi gerekirse, aşağıdaki işlem yapılır:

- *GMATM* seçeneği belirtilmediyse, tamamlanma kodu *CCWARN* ve neden kodu *RC2080* ile ileti dönüştürülmez.
- *GMATM* seçeneği belirtildiyse, tamamlama kodu *CCWARN* olarak ayarlanır, neden kodu *RC2079* olarak ayarlanır ve dönüştürme işlemi devam eder.

4. İleti kesilmeden arabelleğe yerleştirilebiliyorsa ya da *GMATM* seçeneği belirtildiyse, aşağıdaki işlem yapılır:

- Biçim yerleşik bir biçimse, arabellek kuyruk yöneticisinin veri dönüştürme hizmetine geçirilir.
- Biçim yerleşik bir biçim değilse, arabellek, biçimle aynı adı taşıyan kullanıcı tarafından yazılan bir çıkışa geçirilir. Çıkış bulunamazsa, tamamlanma kodu *CCWARN* ve neden kodu *RC2110* ile ileti dönüştürülmez.

Hata ortaya çıkmazsa, veri dönüştürme hizmetinden ya da kullanıcı tarafından yazılan çıkışın çıkışı dönüştürülen iletidir ve MQGET çağrısını yayınlayan uygulamaya döndürülecek tamamlanma kodu ve neden kodu döndürülür.

5. Dönüştürme başarılı olursa, kuyruk yöneticisi dönüştürülen iletiyi uygulamaya döndürür. Bu durumda, MQGET çağrısının döndürdüğü tamamlanma kodu ve neden kodu genellikle aşağıdaki birleşimlerden biri olur:

Tamamlanma kodu

Neden Kodu

CCOK

RCNONE

CCWARN

RC2079

Ancak, dönüştürme kullanıcı tarafından yazılan bir çıkış tarafından gerçekleştirilirse, dönüştürme başarılı olsa da, diğer neden kodları döndürülebilir.

Dönüştürme başarısız olursa (herhangi bir nedenle), kuyruk yöneticisi, **MSGDSC** değiştirgesindeki **MDCSI** ve **MDENC** alanları iletilerdeki denetim bilgilerindeki değerlere ve **CCWARN** tamamlanma koduyla, dönüştürülmemiş iletiyi uygulamaya döndürür.

IBM i **IBM i üzerinde işleme kuralları**

Yerleşik bir biçimi dönüştürürken, kuyruk yöneticisi bu konuda açıklanan işleme kurallarını izler.

Kuyruk yöneticisi tarafından zorlanmamasına rağmen, bu kuralları kullanıcı tarafından yazılan çıkışlara uygulamayı düşünün. Kuyruk yöneticisi tarafından dönüştürülen yerleşik biçimler şunlardır:

- FMADMN
- FMMDE
- FMCICS
- FMPCF
- FMCMD1
- FMRMH
- FMCMD2
- FMRFH
- FMDLH
- FMRFH2
- FMDH
- FMSTR
- FMEVNT
- FMTM
- FMIMS
- FMXQH
- FMIMVS

1. İleti dönüştürme sırasında genişler ve **BUFFER** değiştirgesinin büyüklüğünü aşarsa, aşağıdaki işlem gerçekleştirilir:
 - **GMATM** seçeneği belirtilmediyse, tamamlanma kodu **CCWARN** ve neden kodu **RC2120** ile ileti dönüştürülmeden döndürülür.
 - **GMATM** seçeneği belirtildiyse, ileti kesilir, tamamlanma kodu **CCWARN** olarak ayarlanır, neden kodu **RC2079** olarak ayarlanır ve dönüştürme işlemi devam eder.
2. Kesme oluşursa (dönüştürme öncesi ya da sırasında), **BUFFER** değiştirgesinde döndürülen geçerli byte sayısı arabelleğin uzunluğundan az olabilir.
Örneğin, 4 baytlık bir tamsayı ya da bir DBCS karakteri arabelleğin sonunu takarsa bu durum oluşabilir. Eksik bilgi ögesi dönüştürülmediği için, döndürülen iletilerdeki byte 'lar geçerli bilgi içermiyor. Dönüştürme sırasında dönüştürülmeden önce kesilen bir ileti küçültülürse de bu durum oluşabilir.
Döndürülen geçerli bayt sayısı arabelleğin uzunluğundan azsa, arabelleğin sonundaki kullanılmayan bayt sayısı boş değere ayarlanır.
3. Bir dizi ya da dizgi arabelleğin sonunu tamamlarsa, verilerin çoğu dönüştürülür; yalnızca, eksik olan belirli dizi ögesi ya da DBCS karakteri dönüştürülmez; öncesinde gelen dizi ögeleri ya da karakterler dönüştürülür.
4. Kesme oluşursa (dönüştürme öncesinde ya da sırasında), **DATLEN** parametresi için döndürülen uzunluk, kesilmeden önce *unconversion* iletilisinin uzunluğudur.

5. Dizgiler tek baytlık karakter takımları (SBCS), çift baytlık karakter takımları (DBCS) ya da çok baytlık karakter takımları (MBCS) arasında dönüştürüldüğünde, dizgiler genişletilebilir ya da daralabilir.
- PCF biçimlerinde FMADMN, FMEVNT ve FMPCF, MQCFST ve MQCFSL yapılarındaki dizgiler, dönüştürmeden sonra dizgiyi sığdırmak için gereken şekilde genişletilir ya da daralır.
Dizgi listesi yapısı MQCFSL için, listedeki dizgiler farklı miktarlarda genişletilebilir ya da daraltabilir. Bu durumda, kuyruk yöneticisi kısa dizgileri, dönüşümden sonraki en uzun dizgiyle aynı uzunlukta olacak şekilde boşluklarla doldurur.
 - FMRMH biçiminde, RMSE0, RMSNO, RMDE0ve RMDNO alanları tarafından adreslenen dizgiler, dönüştürmeden sonra dizgileri sığdırmak için gerektiği şekilde genişler ya da daralır.
 - FMRFH biçiminde, RFNVS alanı, dönüştürmeden sonra ad-değer çiftlerini barındırmak için gerektiği şekilde genişletir ya da daralır.
 - Sabit alan büyüklüklü yapılarda, kuyruk yöneticisi, önemli bir bilgi kaybolmazsa, dizgilerin sabit alanları içinde genişletilmesine ya da daralmasına izin verir. Bu bağlamda, alandaki ilk boş karakteri izleyen sondaki boşluklar ve karakterler önemsiz olarak değerlendirilir.
 - Dizgi genişler, ancak alandaki dönüştürülen dizgiyi yerleştirmek için yalnızca önemsiz karakterlerin atılması gerekirse, dönüştürme başarılı olur ve çağrı CCOK ve neden kodu RCNONE ile tamamlanır (başka hata olmadığı varsayılarak).
 - Dizgi genişler, ancak dönüştürülen dizginin alana sığması için önemli karakterlerin atılmasını gerektiriyorsa, ileti dönüştürülmeden döndürülür ve çağrı CCWARN ve neden kodu RC2190ile tamamlanır.
 - Not:** Neden kodu RC2190 , bu durumda GMATM seçeneğinin belirtilip belirtilmediğini gösterir.
 - Dizilim kasılırsa, kuyruk yöneticisi dizgiyi alan uzunluğuna kadar boşluklarla doldurur.
6. Kullanıcı verilerinin izlediği bir ya da daha fazla IBM MQ üstbilgi yapısından oluşan iletiler için, bir ya da daha fazla üstbilgi yapısının dönüştürülmesi mümkündür, ancak iletinin geri kalanı dönüştürülmez. Ancak, iki istisna dışında, her üstbilgi yapısındaki MDCSI ve MDENC alanları her zaman üstbilgi yapısını izleyen verilerin karakter kümesini ve kodlamasını doğru olarak gösterir.
- Bu iki kural dışı durum, bu yapılardaki MDCSI ve MDENC alanlarındaki değerlerin önemli olmadığı MQCIH ve MQIIH yapılarıdır. Bu yapılar için, yapıyı izleyen veriler MQCIH ya da MQIIH yapısıyla aynı karakter kümesi ve kodlamadadır.
7. Alınmakta olan iletinin denetim bilgilerindeki ya da **MSGDSC** değiştirgesindeki MDCSI ya da MDENC alanları tanımlanmamış ya da desteklenmeyen değerleri belirtiyorsa, kuyruk yöneticisi, iletiyi dönüştürürken tanımlanmamış ya da desteklenmeyen değerlerin kullanılması gerekmiyorsa hatayı yoksayabilir.
- Örneğin, iletideki MDENC alanı desteklenmeyen bir kayar noktalı kodlama belirtiyorsa, ancak ileti yalnızca tamsayı verileri içeriyorsa ya da dönüştürme gerektirmeyen kayan nokta verileri içeriyorsa (kaynak ve hedef kayan nokta kodlamaları aynı olduğu için), hata tanılanabilir ya da tanılanmayabilir.
- Hata tanılanırsa, ileti dönüştürülmemiş olarak döndürülür; tamamlanma kodu CCWARN ve RC2111, RC2112, RC2113, RC2114 ya da RC2115, RC2116, RC2117, RC2118 neden kodları (uygun); **MSGDSC** parametresindeki MDCSI ve MDENC alanları, iletideki denetim bilgilerindeki değerlere ayarlanır.
- Hata tanı konmazsa ve dönüştürme başarıyla tamamlanırsa, **MSGDSC** değiştirgesindeki MDCSI ve MDENC alanlarında döndürülen değerler, MQGET çağrısı yayınlayan uygulamanın belirttiği değerlerdir.
8. Her durumda, ileti uygulamaya dönüştürülmeden döndürülürse, tamamlanma kodu CCWARN olarak ayarlanır ve **MSGDSC** parametresindeki MDCSI ve MDENC alanları, dönüştürülmemiş veriler için uygun değerlere ayarlanır. Bu, FMNONE için de yapılır.
- REASON** parametresi, iletinin de kesilmesi gerekmedikçe, dönüştürmenin neden gerçekleştirilemediğini gösteren bir koda ayarlanır; kesmeyle ilgili neden kodları, dönüştürmeyle ilgili neden kodlarından önceliklidir. (Kesilmiş bir iletinin dönüştürülüp dönüştürülmediğini saptamak için, **MSGDSC** değiştirgesindeki MDCSI ve MDENC alanlarında döndürülen değerleri denetleyin.)
- Bir hata saptandığında, belirli bir neden kodu döndürülür ya da RC2119genel neden kodu döndürülür. Döndürülen neden kodu, temel veri dönüştürme hizmetinin tanılama yeteneklerine bağlıdır.

9. CCWARN tamamlanma kodu döndürülürse ve birden çok neden kodu ilgiliyse, öncelik sırası aşağıdaki gibidir:

a. Aşağıdaki neden diğerlerinden önceliklidir:

- RC2079

b. Sonraki öncelik, aşağıdaki nedendir:

- RC2110

c. Geri kalan neden kodları içindeki öncelik sırası tanımlı değil.

10. MQGET çağrısı tamamlandığında:

• Aşağıdaki neden kodu, iletinin başarıyla dönüştürüldüğünü gösterir:

- RCNONE

• Aşağıdaki neden kodu, *Mayıs* iletisinin başarıyla dönüştürüldüğünü gösterir (öğrenmek için **MSGDSC** parametresindeki MDCSI ve MDENC alanlarını denetleyin):

- RC2079

• Diğer tüm neden kodları iletinin dönüştürülmediğini gösterir.

Aşağıdaki işlemler yerleşik biçimlere özgüdür; kullanıcı tanımlı biçimler için geçerli değildir:

1. Aşağıdaki biçimler dışında:

- FMADMN
- FMEVNT
- FMIMVS
- FMPCF
- FMSTR

Yerleşik biçimlerin hiçbiri, kuyruk adlarında geçerli olan karakterler için SBCS karakterleri olmayan karakter kümelerinden ya da karakter takımlarına dönüştürülemez. Böyle bir dönüştürme gerçekleştirme girişiminde bulunulursa, ileti, tamamlanma kodu CCWARN ve neden kodu RC2111 ya da RC2115 ile dönüştürülmeden döndürülür.

Unicode karakter kümesi UTF-16 , kuyruk adlarında geçerli olan karakterler için SBCS karakterleri olmayan bir karakter kümesi örneğidir.

2. Yerleşik bir biçime ilişkin ileti verileri kesilirse, ileti içindeki, dizgi uzunluğu ya da öge ya da yapı sayısı içeren alanlar, uygulamaya döndürülen verilerin uzunluğunu yansıtabilecek şekilde ayarlanmaz; ileti verilerinde bu tür alanlar için döndürülen değerler, kesilmeden önce ileti için geçerli olan değerlerdir.

Kesilmiş FMADMN iletisi gibi iletiler işlenirken, uygulamanın döndürülen verilerin sonundan sonraki verilere erişmeye çalışmadığından emin olmak için dikkatli olunmalıdır.

3. Biçim adı FMDLH ise, ileti verileri bir MQDLH yapısıyla başlar ve bunu sıfır ya da daha fazla byte uygulama iletisi verisi izleyebilir. Uygulama iletisi verilerinin biçimi, karakter kümesi ve kodlaması, iletinin başında MQDLH yapısındaki DLFMT, DLCSİ ve DLENC alanları tarafından tanımlanır. MQDLH yapısı ve uygulama iletisi verilerinin farklı karakter kümeleri ve kodlamaları olabileceği için, MQDLH yapısı ve uygulama iletisi verilerinden birinin, diğerinin ya da her ikisinin de dönüştürülmesi gerekebilir.

Kuyruk yöneticisi önce MQDLH yapısını gerektiği şekilde dönüştürür. Dönüştürme başarılı olursa ya da MQDLH yapısı dönüştürme gerektirmese, kuyruk yöneticisi, uygulama iletisi verilerinin dönüştürülmesinin gerekli olup olmadığını görmek için MQDLH yapısındaki DLCSİ ve DLENC alanlarını denetler. Dönüştürme gerekiyorsa, kuyruk yöneticisi, MQDLH yapısında DLFMT alanı tarafından belirtilen adla kullanıcı tarafından yazılan çıkışı çağırır ya da dönüştürmenin kendisini gerçekleştirir (DLFMT yerleşik bir biçimin adıdır).

MQGET çağrısı CCWARN ' in tamamlanma kodunu döndürürse ve neden kodu dönüştürmenin başarılı olmadığını gösteren kodlardan biriyse, aşağıdakilerden biri geçerlidir:

- MQDLH yapısı dönüştürülemedi. Bu durumda, uygulama ileti verileri de dönüştürülmez.

- MQLH yapısı dönüştürüldü, ancak uygulama iletisi verileri dönüştürülmedi.

Uygulama, **MSGDSC** değiştirgesindeki MDCSI ve MDENC alanlarında ve MQLH yapısındaki değerlerin hangilerinin geçerli olduğunu saptamak için bunları inceleyebilir.

4. Biçim adı FMXQH ise, ileti verileri bir MQXQH yapısıyla başlar ve bunu sıfır ya da daha fazla ek veri baytı takip edebilir. Bu ek veriler genellikle uygulama iletisi verileridir (sıfır uzunluklu olabilir), ancak ek verilerin başında bir ya da daha fazla IBM MQ üstbilgi yapısı da bulunabilir.

MQXQH yapısı, kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır. MQXQH yapısını izleyen verilerin biçimi, karakter kümesi ve kodlaması, MQXQH içinde bulunan MQMD yapısındaki MDFMT, MDCSI ve MDENC alanları tarafından verilir. Yapıdaki sonraki her IBM MQ üstbilgi yapısı için, MDFMT, MDCSI ve MDENC alanları söz konusu yapıyı izleyen verileri açıklar; bu veriler başka bir IBM MQ üstbilgi yapısı ya da uygulama iletisi verileridir.

Bir FMXQH iletisi için GMCONV seçeneği belirtilirse, uygulama iletisi verileri ve MQ üstbilgi yapılarının bazıları dönüştürülür, ancak MQXQH yapısındaki veriler dönüştürülmez. MQGET çağrısından döndüğünüzde:

- **MSGDSC** değiştirgesindeki MDFMT, MDCSI ve MDENC alanlarının değerleri, uygulama iletisi verilerini değil, MQXQH yapısındaki verileri tanımlar; bu nedenle değerler, MQGET çağrısının yayınlandığı uygulama tarafından belirtilenlerle aynı olmaz.

Bunun sonucu, GMCONV seçeneği belirtilen bir iletim kuyruğundan sürekli olarak ileti alan bir uygulamanın, **MSGDSC** değiştirgesindeki MDCSI ve MDENC alanlarını, her MQGET çağrısından önce, uygulama iletisi verileri için gereken değerlere geri döndürmesi gerekmesidir.

- Son MQ üstbilgi yapısındaki MDFMT, MDCSI ve MDENC alanlarının değerleri, uygulama iletisi verilerini açıklar. Başka IBM MQ üstbilgi yapısı yoksa, uygulama iletisi verileri MQXQH yapısındaki MQMD yapısında bu alanlar tarafından tanımlanır. Dönüştürme başarılı olursa, değerler, MQGET çağrısı yayınlayan uygulama tarafından **MSGDSC** değiştirgesinde belirtilenlerle aynı olur.

İleti bir dağıtım listesi iletisiyse, MQXQH yapısını bir MQDH yapısı (artı MQOR ve MQPMR kayıtları dizileri) izler; bu yapıyı, sıfır ya da daha fazla IBM MQ üstbilgi yapısı ve sıfır ya da daha fazla byte uygulama iletisi verisi izleyebilir. MQXQH yapısı gibi, MQDH yapısı da kuyruk yöneticisinin karakter kümesi ve kodlamasında olmalıdır ve GMCONV seçeneği belirtilse bile MQGET çağrısında dönüştürülmez.

Daha önce açıklanan MQXQH ve MQDH yapılarının işlenmesi öncelikle, ileti kuyruklarından ileti aldıklarında ileti kanalı araçları tarafından kullanılmak üzere tasarlanmıştır.

IBM i

IBM i üzerinde rapor iletilerinin dönüştürülmesi

Bir rapor iletisi, özgün iletiyi gönderen tarafından belirtilen rapor seçeneklerine göre çeşitli miktarlarda uygulama iletisi verisi içerebilir.

Özellikle, bir rapor iletisi aşağıdakilerden birini içerebilir:

1. Uygulama iletisi verisi yok
2. Özgün iletideki bazı uygulama iletisi verileri

Bu durum, özgün iletiyi gönderen kişi RO* D değerini belirttiğinde ve ileti 100 bayttan uzun olduğunda ortaya çıkar.

3. Özgün iletideki tüm uygulama iletisi verileri

Bu, özgün iletiyi gönderen kişi RO* F değerini belirttiğinde ya da RO* D değerini belirlediğinde ve ileti 100 bayt ya da daha kısa olduğunda ortaya çıkar.

Kuyruk yöneticisi ya da ileti kanalı aracısı bir rapor iletisi oluşturduğunda, biçim adını özgün iletiden rapor iletisindeki denetim bilgisindeki *MDFMT* alanına kopyalar. Bu nedenle, rapor iletisindeki biçim adı, rapor iletisindeki uzunluktan farklı bir veri uzunluğu anlamına gelebilir (daha önce açıklanan 1. ve 2. durumlar).

Rapor iletisi alındığında GMCONV seçeneği belirtilirse:

- Daha önce açıklanan 1. vaka için, veri dönüştürme çıkışı çağrılmaz (çünkü rapor iletisinde veri olmaz).

- Daha önce açıklanan 3. vaka için biçim adı, ileti verilerinin uzunluğunu doğru şekilde belirtir.
- Ancak, daha önce açıklanan 2. durumda, biçim adının belirttiği uzunluktan *daha kısa* bir iletiyi dönüştürmek için veri dönüştürme çıkışı çağrılır.

Ayrıca, çıkışa geçirilen neden kodu genellikle RCNONE olur (yani, neden kodu iletinin kesildiğini göstermez). Bunun nedeni, ileti verilerinin MQGET çağrısına yanıt olarak alıcının kuyruk yöneticisi tarafından değil, rapor iletisinin *gönderen* tarafından kesilmesiydi.

Bu olasılıklar nedeniyle, veri dönüştürme çıkışı, biçim adını kullanmamalıdır; bunun yerine, çıkış, sağlanan verilerin uzunluğunu denetlemeli ve biçim adının belirttiği uzunluktan daha az veri dönüştürmeye hazırlanmalıdır. Veriler başarıyla dönüştürülebiliyorsa, çıkış tarafından tamamlanma kodu CCOK ve neden kodu RCNONE döndürülmelidir. Dönüştürülecek ileti verilerinin uzunluğu, çıkışa **INLEN** değiştirgesi olarak geçirilir.

Ürüne duyarlı programlama arabirimi

Bir rapor iletisi gerçekleşen bir etkinlikle ilgili bilgi içeriyorsa, etkinlik raporu olarak bilinir. Etkinlik örnekleri şunlardır:

- bir MCA, bir kanaldan aşağı kuyruktan ileti gönderiyor
- bir MCA kanaldan bir ileti alıyor ve bir kuyruğa koyuyor
- Teslim edilemeyen bir iletiyi kuyruğa yollayan bir MCA teslim edilmeyen harf
- bir MCA kuyruktan ileti alıyor ve atılıyor
- Bir iletiyi kuyruğa geri yerleştiren bir gönderilmeyen harf işleyicisi
- Bir PCF isteğini işleyen komut sunucusu-yayınlama isteğini işleyen aracı
- kuyruktan ileti alacak kullanıcı uygulaması-kuyruktaki bir iletiye göz atan kullanıcı uygulaması

Kuyruk yöneticisi de içinde olmak üzere herhangi bir uygulama, rapor üstbilgisini izleyen etkinlik raporuna ileti verilerinin bir kısmını ekleyebilir. Bazıları gönderilirse sağlanması gereken veri miktarı düzeltilmez ve uygulama tarafından kararlaştırılır. Döndürülen bilgiler, etkinlik raporunu işleyen uygulama için yararlı olmalıdır. Kuyruk yöneticisi etkinlik raporları, özgün iletide bulunan standart IBM MQ üstbilgi yapılarını ('MQH' ile başlayan) birlikte döndürür. Bu, örneğin, özgün iletide bulunan MQRFH2 üstbilgilerini içerir. Kuyruk yöneticisi, kendisiyle ilişkili PCF değiştirgelerini değil, bulunan bir MQCFH üstbilgisini de döndürür. Bu, izleme uygulamalarına iletinin neyle ilgili olduğuna dair bir fikir verir.

IBM i

IBM i üzerinde MQDXP (Data-conversion exit parameter)

Veri dönüştürme çıkış parametresi öbeği.

Genel Bakış

Amaç: MQDXP yapısı, MQGET çağrısının işlenmesinin bir parçası olarak ileti verilerini dönüştürmek için çıkış çağrıldığında kuyruk yöneticisinin veri dönüştürme çıkışına aktardığı bir parametredir. Veri dönüştürme çıkışına ilişkin ayrıntılar için MQCONVX çağrısının tanımına bakın.

Karakter kümesi ve kodlama: MQDXP ' deki karakter verileri yerel kuyruk yöneticisinin karakter kümesinde; bu **CodedCharSetId** kuyruk yöneticisi özneteliği tarafından verilir. MQDXP ' deki sayısal veriler yerli makine kodlamasında; bu ENNAT tarafından verilir.

Kullanım: Yalnızca MQDXP ' deki *DXLEN*, *DXCC*, *DXREA* ve *DXRES* alanları çıkış tarafından değiştirilebilir; diğer alanlarda yapılan değişiklikler yoksayılr. Ancak, dönüştürülmekte olan ileti mantıksal iletinin yalnızca bir kısmını içeren bir bölümse, *DXLEN* alanı değiştirilemez.

Denetim, çıkıştan kuyruk yöneticisine döndüğünde, kuyruk yöneticisi MQDXP ' de döndürülen değerleri denetler. Döndürülen değerler geçerli değilse, kuyruk yöneticisi çıkış *DXRES* içinde XRFAIL döndürmüş gibi işlemeye devam eder; Ancak kuyruk yöneticisi, bu durumda çıkışın döndürdüğü *DXCC* ve *DXREA* alanlarının değerlerini yoksayar ve bunun yerine, bu alanların çıkışta *giriş* olarak sahip olduğu değerleri kullanır. MQDXP ' de aşağıdaki değerler bu işlemin gerçekleşmesine neden olur:

- *DXRES* alanı *XROK* ve *XRFAIL* değil
- *DXCC* alanı *CCOK* değil, *CCWARN* değil
- *DXLEN* alan sıfırdan küçük ya da *DXLEN* alanı, dönüştürülmekte olan ileti mantıksal iletinin yalnızca bir kısmını içeren bir bölüm olduğunda değişti.
- “Alanlar” sayfa 1419
- “RPG bildirim (CMQDXPH dosyasını kopyala)” sayfa 1422

Alanlar

MQDXP yapısı aşağıdaki alanları içerir; alanlar **alfabetik sırada** açıklanır:

DXAOP (10 basamaklı işaretli tamsayı)

Uygulama seçenekleri.

Bu, MQGET çağrısı yayınlayan uygulamanın belirttiği MQGMO yapısının *GMOPT* alanının bir kopyasıdır. Çıkış, GMATM seçeneğinin belirtilip belirtilmediğini saptamak için bunları incelemek zorunda olabilir.

Bu, çıkışa bir giriş alanıdır.

DXCC (10 basamaklı işaretli tamsayı)

Tamamlanma kodu.

Çıkış çağrıldığında, çıkış hiçbir şey yapmayı seçmezse, MQGET çağrısı yayınlayan uygulamaya döndürülecek tamamlanma kodunu içerir. Her zaman *CCWARN* ' dir, çünkü ileti kesildi ya da ileti dönüştürme gerektiriyor ve bu işlem henüz yapılmadı.

Çıkışta, bu alan MQGET çağrısının **CMPCOD** değiştirgesinde uygulamaya döndürülecek tamamlanma kodunu içerir; yalnızca *CCOK* ve *CCWARN* geçerlidir. Çıkışın çıkışta bu alanı nasıl ayarlayacağına ilişkin öneriler için *DXREA* alanının açıklamasına bakın.

Bu, çıkışa bir giriş/çıkış alanıdır.

DXCSI (10 basamaklı işaretli tamsayı)

Uygulamanın gerektirdiği karakter kümesi.

Bu, MQGET çağrısını yayınlayan uygulamanın gerektirdiği karakter kümesinin kodlanmış karakter kümesi tanıtıcısıdır; daha fazla ayrıntı için MQMD yapısındaki *MDCSI* alanına bakın. Uygulama MQGET çağrısında CSQM özel değerini belirtirse, kuyruk yöneticisi çıkışı çağırmadan önce kuyruk yöneticisi tarafından kullanılan karakter kümesinin gerçek karakter kümesi tanıtıcısı olarak değiştirir.

Dönüştürme başarılı olursa, çıkış bunu ileti tanımlayıcısındaki *MDCSI* alanına kopyalamalıdır.

Bu, çıkışa bir giriş alanıdır.

DXENC (10 basamaklı işaretli tamsayı)

Uygulamanın gerektirdiği sayısal kodlama.

Bu, MQGET çağrısının yayınlanması için uygulamanın gerektirdiği sayısal kodlamadır; daha fazla ayrıntı için MQMD yapısındaki *MDENC* alanına bakın.

Dönüştürme başarılı olursa, çıkış bunu ileti tanımlayıcısındaki *MDENC* alanına kopyalamalıdır.

Bu, çıkışa bir giriş alanıdır.

DXHCN (10 basamaklı işaretli tamsayı)

Bağlantı tanıtıcısı.

Bu, MQXCNC çağrısında kullanılacak bir bağlantı tanıtıcısıdır. Bu tanıtıcı, MQGET çağrısını yayınlayan uygulama tarafından belirtilen tanıtıcı ile aynı olmayabilir.

DXLEN (10 basamaklı işaretli tamsayı)

İleti verilerinin bayt cinsinden uzunluğu.

Çıkış çağrıldığında, bu alan uygulama ileti verilerinin özgün uzunluğunu içerir. İleti, uygulama tarafından sağlanan arabelleğe sığması için kesildiyse, çıkışa sağlanan iletinin boyutu *DXLEN* değerinden *daha küçük* olur. Çıkışa sağlanan iletinin boyutu, oluşabilecek herhangi bir kesmeye bakılmaksızın, her zaman çıkışın **INLEN** parametresi tarafından verilir.

Kesme, çıkışa girişte RC2079 değerini içeren *DXREA* alanıyla gösterilir.

Çoğu dönüştürmenin bu uzunluğu değiştirmesi gerekmez, ancak gerekiyorsa bir çıkış bunu yapabilir; çıkış tarafından ayarlanan değer, MQGET çağrısının **DATLEN** değiştirilmesinde uygulamaya döndürülür. Ancak, dönüştürülmekte olan ileti mantıksal iletinin yalnızca bir kısmını içeren bir bölümse, bu uzunluk değiştirilemez. Bunun nedeni, uzunluğun değiştirilmesinin mantıksal iletideki sonraki kesimlerin görelî konumlarının yanlış olmasına neden olmasıdır.

Çıkış verilerin uzunluğunu değiştirmek isterse, kuyruk yöneticisinin *dönüştürülmemiş* verilerin uzunluğuna bağlı olarak, ileti verilerinin uygulamanın arabelleğine uyup uymayacağına önceden karar verdiğini unutmayın. Bu karar, iletinin kuyruktan kaldırılıp kaldırılmayacağını (ya da göz atma imlecinin bir göz atma isteği için taşınacağını) ve dönüştürmenin neden olduğu veri uzunluğundaki herhangi bir değişiklikten etkilenmeyeceğini belirler. Bu nedenle, dönüştürme çıkışlarının uygulama ileti verilerinin uzunluğunun değişmesine neden olmaması önerilir.

Karakter dönüştürme bir uzunluk değişikliğini gösteriyorsa, bir dizilim bayt cinsinden aynı uzunlukta başka bir dizgiye dönüştürülebilir, sondaki boşluklar kesilebilir ya da gerektiğinde boşluklarla doldurulur.

İleti uygulama ileti verisi içermezse çıkış çağrılmaz; bu nedenle *DXLEN* her zaman sıfırdan büyüktür.

Bu, çıkışa bir giriş/çıkış alanıdır.

DXREA (10 basamaklı işaretli tamsayı)

Neden kodu niteleyici *DXCC*.

Çıkış çağrıldığında, çıkış hiçbir şey yapmamayı seçerse, MQGET çağrılmasını yayınlayan uygulamaya döndürülecek neden kodunu içerir. Olası değerler arasında, iletinin uygulama tarafından sağlanan arabelleğe sığması için kesildiğini belirten RC2079 ve RC2119 vardır; bu, iletinin dönüştürme gerektirdiğini, ancak bunun henüz yapılmadığını gösterir.

Çıkışın çıkışında, bu alan MQGET çağrısının **REASON** değiştirilmesinde uygulamaya döndürüleceği nedeni içerir; aşağıdakiler önerilir:

- *DXREA* ' in çıkışta RC2079 değeri varsa, dönüştürmenin başarılı olup olmadığına bakılmaksızın *DXREA* ve *DXCC* alanları değiştirilmemelidir.

(*DXCC* alanı CCOK değilse, iletiyi alan uygulama, ileti tanımlayıcısındaki *MDENC* ve *MDCSI* değerlerini istenen değerlerle karşılaştırarak bir dönüştürme başarısızlığını belirleyebilir; buna karşılık, uygulama kesilmiş bir iletiyi arabelleğe takılan bir iletiden ayırt edemez. Bu nedenle, RC2079 , dönüştürme hatasını gösteren nedenlerden biri yerine döndürülmelidir.)

- *DXREA* ' in çıkışa girişte başka bir değeri varsa:

- Dönüştürme başarılı olursa, *DXCC* CCOK olarak ayarlanmalı ve *DXREA* RCNONE olarak ayarlanmalıdır.
- Dönüştürme başarısız olursa ya da ileti genişler ve arabelleğe sığacak şekilde kesilmesi gerekirse, *DXCC* CCWARN (ya da değiştirilmeden bırakılmalıdır) olarak ayarlanmalı ve *DXREA* hatanın türünü belirtmek için aşağıdaki değerlerden birine ayarlanmalıdır.

Dönüştürme sonrası ileti arabellek için çok büyükse, ileti ancak MQGET çağrısını yayınlayan uygulama GMATM seçeneğini belirtmişse kesilmelidir:

- Bu seçenek belirtildiyse, neden RC2079 döndürülmelidir.
- Bu seçenek belirtilmediyse, ileti dönüştürülmeden döndürülmelidir; neden kodu RC2120.

Aşağıdaki listedeki neden kodları, dönüştürmenin başarısız olma nedenini belirtmek için çıkış tarafından kullanılması önerilir, ancak çıkış, uygun görülürse RC* kodları kümesinden diğer değerleri döndürebilir. Buna ek olarak, RC0900 ile RC0999 arasındaki değerler aralığı, çıkışın MQGET çağrısı

yayınlayan uygulamayla iletişim kurmak istediği koşulları belirtmek için çıkış tarafından kullanılmak üzere ayrılır.

Not: İleti başarıyla dönüştürülemezse, kuyruk yöneticisinin dönüştürülmemiş iletiyi döndürmesi için çıkışın *DXRES* alanında *XRFAIL* döndürmesi gerekir. Bu, *DXREA* alanında döndürülen neden kodundan bağımsız olarak geçerlidir.

RC0900

(900, X'384 ') Uygulama tanımlı neden kodu için daha düşük değer.

RC0999

(999, X'3E7') Uygulama tanımlı neden kodu için en yüksek değer.

RC2120

(2120, X'848 ') Dönüştürülen veriler arabellek için çok büyük.

RC2119

(2119, X'847 ') İleti verileri dönüştürülmedi.

RC2111

(2111, X'83F') Kaynak kodlanmış karakter takımı tanıtıcısı geçersiz.

RC2113

(2113, X'841 ') İletideki paketli ondalık kodlama tanınmıyor.

RC2114

(2114, X'842 ') İletide kayan nokta kodlaması tanınmıyor.

RC2112

(2112, X'840 ') Kaynak tamsayı kodlaması tanınmıyor.

RC2115

(2115, X'843 ') Hedef kodlanmış karakter takımı tanıtıcısı geçersiz.

RC2117

(2117, X'845 ') Alıcı tarafından belirtilen Paketlenmiş ondalık kodlama tanınmıyor.

RC2118

(2118, X'846 ') Alıcı tarafından belirtilen kayan nokta kodlaması tanınmıyor.

RC2116

(2116, X'844 ') Hedef tamsayı kodlaması tanınmıyor.

RC2079

(2079, X'81F') İzlenen ileti döndürüldü (işleme tamamlandı).

Bu, çıkışa bir giriş/çıkış alanıdır.

DXRES (10 basamaklı işaretli tamsayı)

Çıkıştan cevap geldi.

Bu, çıkış tarafından, dönüştürmenin başarılı olduğunu ya da başka bir şekilde gerçekleştiğini belirtmek için ayarlanır. Aşağıdakilerden biri olmalıdır:

XR0K

Dönüştürme başarılı oldu.

Çıkış bu değeri belirtiyorsa, kuyruk yöneticisi MQGET çağrısı yayınlayan uygulamaya aşağıdakileri döndürür:

- Çıkıştaki *DXCC* alanının değeri
- Çıkıştaki *DXREA* alanının değeri
- Çıkıştaki *DXLEN* alanının değeri
- Çıkışın çıkış arabelleğinin içeriği *OUTBUF*. Döndürülen bayt sayısı, çıkışın **OUTLEN** parametresinden daha küçük ve çıkıştaki *DXLEN* alanının değeridir.

Çıkışın ileti tanımlayıcı parametresindeki *MDENC* ve *MDCSI* alanları *her ikisi* değiştirilmediyse, kuyruk yöneticisi şunları döndürür:

- Çıkışa ilişkin *giriş* üzerindeki MQDXP yapısındaki *MDENC* ve *MDCSI* alanlarının değeri

Çıkışın ileti tanımlayıcı parametresindeki *MDENC* ve *MDCSI* alanlarından biri ya da her ikisi değiştirildiyse, kuyruk yöneticisi şunları döndürür:

- Çıkışta çıkışın ileti tanımlayıcı parametresindeki *MDENC* ve *MDCSI* alanlarının değeri
-

XRFAIL

Dönüştürme başarısız oldu.

Çıkış bu değeri belirtiyorsa, kuyruk yöneticisi MQGET çağrısı yayınlayan uygulamaya aşağıdakileri döndürür:

- Çıkıştaki *DXCC* alanının değeri
 - Çıkıştaki *DXREA* alanının değeri
 - Çıkışa *giriş* üzerindeki *DXLEN* alanının değeri
 - Çıkışın giriş arabelleğinin içeriği *INBUF*. **INLEN** parametresi tarafından döndürülen bayt sayısı
- Çıkış *INBUF*değiştirdiyse, sonuçlar tanımsız olur.

DXRES , çıkıştan bir çıkış alanıdır.

DXSID (4 baytlık karakter dizgisi)

Yapı tanıtıcısı.

Değer şöyle olmalıdır:

DXSIDV

Veri dönüştürme çıkış parametresi yapısına ilişkin tanıtıcı.

Bu, çıkışa bir giriş alanıdır.

DXVER (10 basamaklı işaretli tamsayı)

Yapı sürümü numarası.

Değer şöyle olmalıdır:

DXVER1

Veri dönüştürme çıkış parametresi yapısına ilişkin sürüm numarası.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

DXVERC

Veri dönüştürme çıkış parametresi yapısının yürürlükteki sürümü.

Not: Bu yapının yeni bir sürümü tanıtıldığında, var olan parçanın düzeni değişmez. Bu nedenle çıkış, *DXVER* alanının, çıkışın kullanması gereken alanları içeren en düşük sürüme eşit ya da bu sürümden büyük olup olmadığını kontrol etmelidir.

Bu, çıkışa bir giriş alanıdır.

DXXOP (10 basamaklı işaretli tamsayı)

Ayrılmış.

Bu ayrılmış bir alandır; değeri 0.

RPG bildiri (CMQDXPH dosyasını kopyala)

```
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

IBM i üzerinde MQXCNCV (karakterleri dönüştür)

MQXCNCV çağrısı, karakterleri bir karakter kümesinden diğerine dönüştürür.

Bu çağrı, IBM MQ çerçeve arabirimlerinden biri olan IBM MQ Veri Dönüştürme Arabirimi 'nin (DCI) bir parçasıdır. Not: Bu çağrı yalnızca veri dönüştürme çıkışından kullanılabilir.

- [“Sözdizimi” sayfa 1423](#)
- [“Parametreler” sayfa 1423](#)
- [“RPG çağrısı \(ILE\)” sayfa 1427](#)

Sözdizimi

MQXCNCV HCONN, OPTS, SRCCSI, SRCLEN, SRCBUF, TGTCSI, TGTLEN, TGTBUF, DATLEN, CMPCOD, REASON)

Parametreler

MQXCNCV çağrısındaki değiştirgeler şunlardır:

HCONN (10 basamaklı işaretli tamsayı)-giriş

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. Olağan durumda, MQDXP yapısının DXHCN alanındaki veri dönüştürme çıkışına geçirilen tanıtıcı olmalıdır; bu tanıtıcı, MQGET çağrısını yayınlayan uygulama tarafından belirtilen tanıtıcı ile aynı olmalıdır.

IBM üzerinde, HCONN için aşağıdaki özel değer belirtilebilir:

HCDEFH

Varsayılan bağlantı tanıtıcısı.

OPTS (10 basamaklı işaretli tamsayı)-giriş

MQXCNCV işlemini denetleyen seçenekler.

Bu bölümde daha sonra açıklanan seçeneklerden sıfır ya da daha fazlası belirtilebilir. Birden çok değer gerekiyorsa, değerler eklenebilir (aynı değişmezi bir kereden fazla eklemeyin).

Varsayılan dönüştürme seçeneği: Aşağıdaki seçenek, varsayılan karakter dönüşümünün kullanımını denetler:

DCCDEF

Varsayılan dönüştürme.

Bu seçenek, çağrıda belirtilen karakter kümelerinden biri ya da her ikisi desteklenmiyorsa, varsayılan karakter dönüşümünün kullanılabilmesini belirtir. Bu, kuyruk yöneticisinin dizgiyi dönüştürürken, belirtilen karakter kümesine yaklaşan, kuruluş tarafından belirtilen bir varsayılan karakter kümesini kullanmasını sağlar.

Not: Dizgiyi dönüştürmek için yaklaşık bir karakter kümesi kullanılmasının sonucu, bazı karakterlerin yanlış dönüştürülebilmesinden kaynaklanır. Bu, yalnızca belirtilen karakter kümesi ve varsayılan karakter kümesi için ortak olan karakterler kullanılarak önlenir.

Varsayılan karakter kümeleri, kuyruk yöneticisi kurulduğunda ya da yeniden başlatıldığında bir yapılandırma seçeneği tarafından tanımlanır.

DCCDEF belirtilmezse, kuyruk yöneticisi dizgiyi dönüştürmek için yalnızca belirtilen karakter kümelerini kullanır ve karakter kümelerinden biri ya da her ikisi desteklenmiyorsa çağrı başarısız olur.

Doldurma seçeneği: Aşağıdaki seçenek, dönüştürülen dizgiyi hedef arabelleğe sığdırmak için, kuyruk yöneticisinin dönüştürülen dizgiyi boşluklarla doldurmasını ya da anlamlı olmayan karakterleri atmasını sağlar:

DCCFIL

Hedef arabelleği doldur.

Bu seçenek, dönüştürmenin hedef arabelleğin tam olarak doldurulabileceği şekilde gerçekleştirilmesini ister:

- Dizgi dönüştürüldüğünde kasılırsa, hedef arabelleği doldurmak için sondaki boşluklar eklenir.
- Dizgi dönüştürüldüğünde genişletilirse, dönüştürülen dizginin hedef arabelleğe sığması için sondaki anlamlı olmayan karakterler atılır. Bu işlem başarılı olursa, çağrı CCOK ve neden kodu RCNONE ile tamamlanır.

Çok az sayıda sonda karakter varsa, hedef arabelleğe sığacak kadar çok dizgi yerleştirilir ve çağrı CCWARN ve neden kodu RC2120 ile tamamlanır.

Önemsiz karakterler şunlardır:

- Sondaki boşluklar
- Dizgide ilk boş karakterin ardından gelen karakterler (ancak ilk boş karakterin kendisi hariç)
- TGTCSI ve TGTLEN dizgisi, hedef arabelleğin geçerli karakterlerle tam olarak ayarlanamayacaksa, çağrı CCFAIL ile başarısız olur ve neden kodu RC2144. Bu durum, TGTCSI saf bir DBCS karakter takımı (UTF-16 gibi) olduğunda ortaya çıkabilir, ancak TGTLEN tek sayıda bayt olan bir uzunluğu belirtir.
- TGTLEN , SRCLEN değerinden küçük ya da daha büyük olabilir. MQXCNCV ' den dönüşte DATLEN , TGTLENIle aynı değere sahiptir.

Bu seçenek belirtilmezse:

- Dizginin, hedef arabellek içinde gerektiği gibi daralmasına ya da genişlemesine izin verilir. Önemsiz sondaki karakterler eklenmez ya da atılmaz.

Dönüştürülen dizgi hedef arabelleğe sığarsa, çağrı CCOK ve neden kodu RCNONE ile tamamlanır.

Dönüştürülen dizgi hedef arabellek için çok büyükse, hedef arabelleğe sığacak dizginin çoğu yerleştirilir ve çağrı CCWARN ve neden kodu RC2120 ile tamamlanır. Bu durumda TGTLEN bayttan az döndürülebilir.

- TGTLEN , SRCLEN değerinden küçük ya da daha büyük olabilir. MQXCNCV ' den dönüşte, DATLEN TGTLENI değerinden küçük ya da bu değere eşit.

Kodlama seçenekleri: Kaynak ve hedef dizgilerin tamsayı kodlamasını belirtmek için aşağıdaki seçenekler kullanılabilir. İlgili kodlama, yalnızca karşılık gelen karakter kümesi tanıtıcısı, ana bellekteki karakter kümesinin gösteriminin ikili tamsayılar için kullanılan kodlamaya bağlı olduğunu gösterdiğinde kullanılır. Bu yalnızca belirli çok baytlık karakter kümelerini etkiler (örneğin, UTF-16 karakter kümeleri).

Karakter takımı tek baytlık karakter takımı (SBCS) ya da tamsayı kodlamasına bağımlı olmayan ana saklama alanında gösterimi olan çok baytlık bir karakter takımıysa, kodlama dikkate alınmaz.

DCCT* değerlerinden biriyle birlikte, DCCS* değerlerinden yalnızca biri belirtilmelidir:

DCCSID

Kaynak kodlama, ortam ve programlama dili için varsayılan kodlamadır.

DCCSNO

Kaynak kodlaması normal.

DCCSRE

Kaynak kodlaması tersine çevrildi.

DCCSUN

Kaynak kodlaması tanımlı değil.

DCCTNA

Hedef kodlama, ortam ve programlama dili için varsayılan değerdir.

DCCTNO

Hedef kodlaması normal.

DCCTRE

Hedef kodlaması tersine çevrildi.

DCCTUN

Hedef kodlama tanımlı değil.

Önceden tanımlanan kodlama değerleri doğrudan OPTS alanına eklenebilir. Ancak, kaynak ya da hedef kodlama MQMD ' deki MDENC alanından ya da başka bir yapıdan elde edilirse, aşağıdaki işlemler yapılmalıdır:

1. Tamsayı kodlaması, kayar ve paketlenmiş ondalık kodlamaları ortadan kaldırılarak MDENC alanından çıkarılmalıdır; bunun nasıl yapılacağına ilişkin ayrıntılar için bkz. [“IBM i üzerinde kodlamaları çözümleme” sayfa 1408](#) .
2. Adım 1 'den kaynaklanan tamsayı kodlaması, OPTS alanına eklenmeden önce uygun katsayıyla çarpılmalıdır. Bu etkenler şunlardır:

DCCSFA

Kaynak kodlamaya ilişkin faktör

DCCTFA

Hedef kodlamaya ilişkin faktör

Belirtilmezse, kodlama seçenekleri varsayılan olarak tanımsız (DCC* UN) olarak ayarlanır. Çoğu durumda bu, MQXCNCV çağrısının başarıyla tamamlanmasını etkilemez. Ancak, karşılık gelen karakter kümesi, kodlamaya (örneğin, bir UTF-16 karakter kümesi) bağlı olan çok baytlık bir karakter takılıysa, çağrı RC2112 ya da RC2116 neden koduyla başarısız olur.

Varsayılan seçenek: Önceden açıklanan seçeneklerden hiçbiri belirtilmezse, aşağıdaki seçenek kullanılabilir:

DCCNON

Seçenek belirtilmedi.

DCCNON, program belgelerine yardımcı olmak için tanımlanır. Bu seçeneğin başka bir seçenikle birlikte kullanılması amaçlanmamıştır, ancak değeri sıfır olduğu için bu tür bir kullanım algılanamaz.

SRCCSI (10 basamaklı işaretli tamsayı)-giriş

Dönüştürmeden önce dizilimin kodlanmış karakter takımı tanıtıcısı.

Bu, SRCBUFiçindeki giriş dizgisinin kodlanmış karakter takımı tanıtıcısıdır.

SRCLEN (10 basamaklı işaretli tamsayı)-giriş

Dönüştürmeden önceki dizginin uzunluğu.

Bu, SRCBUF ; içindeki giriş dizgisinin bayt cinsinden uzunluğudur. Sıfır ya da daha büyük olmalıdır.

SRCBUF (1 baytlık karakter dizilimi x SRCLEN)-giriş

Dönüştürülecek dizgi.

Bu, bir karakter kümesinden diğerine dönüştürülecek dizgiyi içeren arabellektir.

TGTCSI (10 basamaklı işaretli tamsayı)-giriş

Dönüştürmeden sonra dizilimin kodlanmış karakter takımı tanıtıcısı.

Bu, SRCBUF ' in dönüştürüleceği karakter takımının kodlanmış karakter takımı tanıtıcısıdır.

TGTLEN (10 basamaklı işaretli tamsayı)-giriş

Çıkış arabelleğinin uzunluğu.

Bu, çıkış arabelleğinin bayt cinsinden uzunluğudur TGTBUF ; Sıfır ya da daha büyük olmalıdır. SRCLENDegerinden küçük ya da büyük olabilir.

TGTBUF (1 baytlık karakter dizilimi x TGTLEN)-çıkış

Dönüştürmeden sonraki dizgi.

Bu, TGTCSI tarafından tanımlanan karakter kümesine dönüştürüldükten sonraki dizedir. Dönüştürülen dizgi, dönüştürülmemiş dizgiden daha kısa ya da daha uzun olabilir. **DATLEN** parametresi, döndürülen geçerli bayt sayısını gösterir.

DATLEN (10 basamaklı işaretli tamsayı)-çıkış

Çıkış dizgisinin uzunluğu.

Bu, TGTBUF çıkış arabelleğinde döndürülen dizginin uzunluğudur. Dönüştürülen dizgi, dönüştürülmemiş dizgiden daha kısa ya da daha uzun olabilir.

CMPCOD (10 basamaklı işaretli tamsayı)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

CCOK

Başarıyla tamamlandı.

CCWARN

Uyarı (kısmi tamamlama).

CCFAIL

Arama başarısız oldu.

REASON (10 basamaklı işaretli tamsayı)-çıkış

Neden kodu niteleyici CMPCOD.

CMPCOD CCOK ise:

RCNONE

(0, X'000 ') Raporlamak için bir neden yok.

CMPCOD CCWARN ise:

RC2120

(2120, X'848 ') Dönüştürülen veriler arabellek için çok büyük.

CMPCOD CCFAIL ise:

RC2010

(2010, X'7DA') Veri uzunluğu parametresi geçersiz.

RC2150

(2150, X'866 ') DBCS dizgisi geçersiz.

RC2018

(2018, X'7E2') Bağlantı tanıtıcısı geçersiz.

RC2046

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

RC2102

(2102, X'836 ') Sistem kaynakları yetersiz.

RC2145

(2145, X'861 ') Kaynak arabellek değiştirgesi geçersiz.

RC2111

(2111, X'83F') Kaynak kodlanmış karakter takımı tanıtıcısı geçersiz.

RC2112

(2112, X'840 ') Kaynak tamsayı kodlaması tanınmıyor.

RC2143

(2143, X'85F') Kaynak uzunluğu parametresi geçersiz.

RC2071

(2071, X'817 ') Kullanılabilir bellek yetersiz.

RC2146

(2146, X'862 ') Hedef arabellek değiştirgesi geçersiz.

RC2115

(2115, X'843 ') Hedef kodlanmış karakter takımı tanıtıcısı geçersiz.

RC2116

(2116, X'844 ') Hedef tamsayı kodlaması tanınmıyor.

RC2144

(2144, X'860 ') Hedef uzunluk parametresi geçersiz.

RC2195

(2195, X'893 ') Beklenmeyen hata oluştu.

Bu neden kodlarına ilişkin ek bilgi için bkz. ["IBM i \(ILE RPG\) için dönüş kodları" sayfa 1402.](#)

RPG çağırısı (ILE)

```

C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      MQXCNV (HCONN : OPTS : SRCCSI :
C                               SRCLEN : SRCBUF : TGTCSE :
C                               TGTLEN : TGTBUF : DATLEN :
C                               CMPCOD : REASON)

```

Çağrıya ilişkin prototip tanımlaması:

```

D*..1.....2.....3.....4.....5.....6.....7..
DMQXCNV      PR          EXTPROC('MQXCNV')
D* Connection handle
D HCONN          10I 0 VALUE
D* Options that control the action of MQXCNV
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

```

IBM i **IBM i üzerinde MQCONVX (Veri dönüştürme çıkışı)**

Bu çağrı tanımlaması, veri dönüştürme çıkışına geçirilen parametreleri açıklar.

Kuyruk yöneticisi MQCONVX adlı bir giriş noktası sağlamadı (kullanım notuna bakın [“11” sayfa 1429](#)).

Bu tanım, IBM MQ çerçeve arabirimlerinden biri olan IBM MQ Veri Dönüştürme Arabirimi 'nin (DCI) bir parçasıdır.

- [“Sözdizimi” sayfa 1428](#)
- [“Kullanım notları” sayfa 1428](#)
- [“Parametreler” sayfa 1429](#)
- [“RPG çağrısı \(ILE\)” sayfa 1430](#)

Sözdizimi

MQCONVX (MQDXP, MQMD, INLEN, INBUF, OUTLEN, OUTBUF)

Kullanım notları

1. Veri dönüştürme çıkışı, bir MQGET çağrısının işlenmesi sırasında denetimi alan, kullanıcı tarafından yazılan bir çıkıştır. Veri dönüştürme çıkışı tarafından gerçekleştirilen işlev, çıkışın sağlayıcısı tarafından tanımlanır; ancak, çıkış burada açıklanan kurallara ve ilişkili değiştirge yapısına uygun olmalıdır MQDXP.

Veri dönüştürme çıkışı için kullanılacak programlama dilleri, ortam tarafından belirlenir.

2. Çıkış yalnızca aşağıdaki deyimlerin *tümü* true ise çağrılır:

- MQGET çağrısında GMCONV seçeneği belirtildi
- İleti tanımlayıcıdaki *MDFMT* alanı FMNONE değil
- İleti gerekli gösterimde değil; yani, iletinin *MDCSI* ve *MDENC* iletilerinden biri ya da her ikisi, MQGET çağrısında sağlanan ileti tanımlayıcısında uygulama tarafından belirtilen değerden farklı
- Kuyruk yöneticisi dönüştürme işlemini başarıyla gerçekleştirmedi
- Uygulamanın arabelleğinin uzunluğu sıfırdan büyük
- İleti verilerinin uzunluğu sıfırdan büyük
- MQGET işlemi sırasında şu ana kadar gelen neden kodu: RCNONE ya da RC2079

3. Bir çıkış yazılırken, çıkışı, kesilen iletileri dönüştürmesine izin verecek şekilde kodlamaya dikkat edilmelidir. Kesilmiş iletiler aşağıdaki şekillerde ortaya çıkabilir:

- Alan uygulama, iletiden küçük bir arabellek sağlar, ancak MQGET çağrısında GMATM seçeneğini belirtir.

Bu durumda, çıkışa girişteki **MQDXP** parametresindeki *DXREA* alanı RC2079 değerine sahip olur.

- İletiyi gönderen kişi, iletiyi göndermeden önce kesmiş. Bu, örneğin rapor iletilerinde gerçekleşebilir (daha fazla ayrıntı için bkz. [“IBM i üzerinde rapor iletilerinin dönüştürülmesi” sayfa 1417](#)).

Bu durumda, çıkışa girişteki **MQDXP** değiştirgesindeki *DXREA* alanı RCNONE değerine sahip olur (alan uygulama ileti için yeterli büyüklükte bir arabellek sağladıysa).

Bu nedenle, çıkışa girişteki *DXREA* alanının değeri, iletinin kesilip kesilmediğine karar vermek için her zaman kullanılamaz.

Kesilmiş bir iletinin ayırt edici özelliği, **INLEN** değiştirgesindeki çıkışa sağlanan uzunluğun, ileti tanımlayıcısındaki *MDFMT* alanında bulunan biçim adının belirttiği uzunluktan az olması olabilir. Bu

nedenle, çıkışın herhangi bir veriyi dönüştürmeye çalışmadan önce *INLEN* değerini denetlemesi gerekir; *çıkış*, biçim adı tarafından belirtilen verilerin tam miktarının sağlandığını varsaymamalıdır.

Çıkış, kesilmiş iletileri dönüştürmek için yazılmamışsa ve **INLEN** beklenen değerden küçükse, *çıkış*, **MQDXP** parametresinin *DXRES* alanında *DXCC* alanı CCWARN ve *DXREA* alanı RC2110olarak ayarlanmış olarak XRFail döndürmelidir.

Çıkış, kesilmiş iletileri dönüştürmek için yazıldıysa, *INBUF*' in sonundan sonraki verileri incelemeye ya da dönüştürmeye çalışmamaya dikkat ederek, *çıkışın* verilerin çoğunu mümkün olduğu kadar dönüştürmesi gerekir (sonraki kullanım notuna bakın). Dönüştürme başarıyla tamamlanırsa, *çıkış*, **MQDXP** değiştirgisindeki *DXREA* alanını değiştirmeden bırakmalıdır. Bu, ileti alıcının kuyruk yöneticisi tarafından kesildiye RC2079, ileti gönderen tarafından kesildiye RCNONE değerini döndürür.

Ayrıca, bir iletinin *OUTBUF* değerinden büyük olduğu noktaya *sırasında* dönüşümünü genişletmesi de mümkündür. Bu durumda, *çıkışın* iletinin kesilip kesilmeyeceğine karar vermesi gerekir; **MQDXP** parametresindeki *DXAOP* alanı, alan uygulamanın GMATM seçeneğini belirtip belirtmediğini belirtir.

4. Genellikle, *INBUF* içinde *çıkışa* sağlanan iletideki tüm verilerin dönüştürülmesi ya da hiçbirinin dönüştürülmesi önerilir. Ancak, ileti dönüştürmeden önce ya da dönüştürme sırasında kesilirse, bu durumda arabelleğin sonunda eksik bir öge olabilir (örneğin: çift baytlık bir karakterin bir baytı ya da 4 baytlık bir tamsayının 3 baytı). Bu durumda, eksik ögenin atlanması ve *OUTBUF* içindeki kullanılmayan baytların boş değere ayarlanması önerilir. Ancak, bir dizi ya da dizgi içindeki tüm öğelerin ya da karakterlerin *dönüştürülmesi* gerekir.
5. İlk kez bir *çıkış* gerektiğinde, kuyruk yöneticisi, biçimle (uzantılar dışında) aynı ada sahip bir nesneyi yüklemeye çalışır. Yüklenen nesne, bu biçim adına sahip iletileri işleyen *çıkışı* içermelidir. *Çıkış* adının ve *çıkışı* içeren nesnenin adının aynı olması önerilir; ancak, tüm ortamlar bunu gerektirmez.
6. Uygulama, kuyruk yöneticisine bağlandığından bu yana *MDFMT* ' u kullanan ilk iletiyi almaya çalıştığında *çıkışın* yeni bir kopyası yüklenir. Kuyruk yöneticisi önceden yüklenmiş bir kopyayı attıysa, başka zamanlarda da yeni bir kopya yüklenebilir. Bu nedenle, bir *çıkış*, *çıkışın* bir çağrısından diğerine bilgi iletmek için statik saklama alanını kullanma girişiminde bulunmamalıdır; *çıkış*, iki çağırma arasında boşaltılabilir.
7. Kuyruk yöneticisi tarafından desteklenen yerleşik biçimlerden biriyle aynı ada sahip kullanıcı tarafından sağlanan bir *çıkış* varsa, kullanıcı tarafından sağlanan *çıkış* yerleşik dönüştürme yordamının yerini almaz. Böyle bir *çıkışın* çağrıldığı durumlar şunlardır:
 - Yerleşik dönüştürme yordamı, ilgili *MDCSI* ya da *MDENC* ürününe ya da bu üründen gelen dönüştürmeleri işleyemezse, ya da
 - Yerleşik dönüştürme yordamı verileri dönüştüremediye (örneğin, dönüştürülemeyen bir alan ya da karakter olduğu için).
8. *Çıkışın* kapsamı ortama bağlıdır. Diğer biçimlerle çakışma riskini en aza indirmek için *MDFMT* adları seçilmelidir. Biçim adını tanımlayan uygulamayı tanıtan karakterlerle başlamaları önerilir.
9. Veri dönüştürme *çıkışı*, MQGET çağrısını yayınlayan programın ortamına benzer bir ortamda çalışır; ortam adres alanını ve kullanıcı tanıtımını içerir (varsa). Program, ileti dönüştürmeyi desteklemeyen bir hedef kuyruk yöneticisine ileti gönderen bir ileti kanalı aracısı olabilir. *Çıkış*, kuyruk yöneticisinin ortamında çalışmadığı için kuyruk yöneticisinin bütünlüğünü bozamaz.
10. *Çıkış* tarafından kullanılacak tek MQI çağrısı MQXCNCV; diğer MQI çağrılarını kullanma girişimi başarısız oluyor; neden kodu RC2219ya da öngörülemeyen diğer hatalar.
11. Kuyruk yöneticisi tarafından MQCONVX adlı bir giriş noktası sağlanmadı. *Çıkışın* adı, tüm ortamlarda gerekli olmasa da, biçim adıyla (MQMD ' deki *MDFMT* alanında bulunan ad) aynı olmalıdır.

Parametreler

MQCONVX çağrısındaki değiştirgeler şunlardır:

MQDXP (MQDXP)-giriş/çıkış

Veri dönüştürme *çıkış* parametresi öbeği.

Bu yapı, çıkışın çağrılmasıyla ilgili bilgileri içerir. Çıkış, dönüştürmenin sonucunu belirtmek için bu yapıdaki bilgileri ayarlar. Bu yapıdaki alanların ayrıntıları için bkz. [“IBM i üzerinde MQDXP \(Data-conversion exit parameter\)” sayfa 1418](#) .

MQMD (MQMD)-giriş/çıkış

İleti tanımlayıcı.

Çıkışa girişte, dönüştürme gerçekleştirilmezse uygulamaya döndürülecek ileti tanımlayıcısı budur. Bu nedenle, *INBUF* içinde bulunan dönüştürülmemiş iletinin *MDFMT*, *MDENC* ve *MDCSI* öğelerini içerir.

Not: Çıkışa geçirilen **MQMD** değiştirgesi her zaman, çıkışı çağıran kuyruk yöneticisi tarafından desteklenen en son MQMD sürümüdür. Çıkışın farklı ortamlar arasında taşınabilir olması amaçlanıyorsa, çıkışın erişmesi gereken alanların yapıda var olduğunu doğrulamak için *MQMD* içindeki *MDVER* alanını denetlemesi gerekir.

IBM üzerinde, çıkışa bir version-2 MQMD geçirilir.

Çıkışta, dönüştürme başarılı olursa, çıkış *MDENC* ve *MDCSI* alanlarını uygulama tarafından istenen değerlerle değiştirmelidir; bu değişiklikler uygulamaya geri yansıtılır. Çıkışın yapıda yaptığı diğer değişiklikler yoksayılır; bunlar uygulamaya geri yansıtılmaz.

Çıkış, MQDXP yapısının *DXRES* alanında *XROK* döndürürse, ancak ileti tanımlayıcısındaki *MDENC* ya da *MDCSI* alanlarını değiştirmezse, kuyruk yöneticisi bu alanlar için MQDXP yapısındaki ilgili alanların çıkışa girişte sahip olduğu değerleri döndürür.

INLEN (10 basamaklı işaretli tamsayı)-giriş

INBUF bayt cinsinden uzunluk.

Bu, *INBUF* giriş arabelleğinin uzunluğudur ve çıkış tarafından işlenecek bayt sayısını belirtir. *INLEN* , dönüşümden önceki ileti verilerinin uzunluğundan ve MQGET çağrısında uygulama tarafından sağlanan arabelleğin uzunluğundan daha azdır.

Değer her zaman sıfırdan büyüktür.

INBUF (1 baytlık bit dizilimi x INLEN)-giriş

Dönüştürülemeyen iletiyi içeren arabellek.

Bu, dönüştürmeden önce ileti verilerini içerir. Çıkış verileri dönüştüremezse, kuyruk yöneticisi çıkış tamamlandıktan sonra bu arabelleğin içeriğini uygulamaya döndürür.

Not: Çıkış *INBUF* ; ögesini değiştirmemelidir. Bu parametre değiştirilirse, sonuçlar tanımsız olur.

OUTLEN (10 basamaklı işaretli tamsayı)-giriş

OUTBUF bayt cinsinden uzunluk.

Bu, *OUTBUF* çıkış arabelleğinin uzunluğudur ve MQGET çağrısında uygulama tarafından sağlanan arabelleğin uzunluğuyla aynıdır.

Değer her zaman sıfırdan büyüktür.

OUTBUF (1 baytlık bit dizilimi x OUTLEN)-çıkış

Dönüştürülen iletiyi içeren arabellek.

Çıkışta, dönüştürme başarılı olduysa (**MQDXP** değiştirgesinin *DXRES* alanında *XROK* değeriyle belirtildiği gibi), **OUTBUF** istenen gösterimde uygulamaya teslim edilecek ileti verilerini içerir. Dönüştürme başarısız olduysa, çıkışın bu arabellekte yaptığı değişiklikler yoksayılır.

RPG çağrısı (ILE)

```
C*..1.....2.....3.....4.....5.....6.....7..
C          CALLP      exitname(MQDXP : MQMD : INLEN :
C                               INBUF : OUTLEN : OUTBUF)
```

Çağrıya ilişkin prototip tanımlaması:

```
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
```

Ürüne duyarlı programlama arabiriminin sonu

Kullanıcı çıkışları, API çıkışları ve kurulabilir hizmetler başvurusu

Kullanıcı çıkışlarınızı, API çıkışlarınızı ve kurulabilir hizmet uygulamalarını geliştirmenize yardımcı olması için bu bölümdeki linformation yardımcı programını kullanın:

- [“MQIEP yapısı” sayfa 1431](#)
- [“Veri dönüştürme çıkış başvurusu” sayfa 1435](#)
- [“MQ_PUBLISH_EXIT-Çıkış yayınla” sayfa 1439](#)
- [“Kanal çıkışı çağrıları ve veri yapıları” sayfa 1446](#)
- [“API çıkış başvurusu” sayfa 1533](#)
- [“Kurulabilir hizmetler arabirimi başvuru bilgileri” sayfa 1593](#)

İlgili kavramlar

[Kullanıcı çıkışları, API çıkışları ve IBM MQ kurulabilir hizmetleri](#)

İlgili görevler

[Kuyruk yöneticisi olanaklarını genişletme](#)

MQIEP yapısı

MQIEP yapısı, çıkış yapmasına izin verilen her işlev çağrısı için bir giriş noktası içerir.

Alanlar

StrucId

Tip: MQCHAR4 -giriş

Yapı tanıtıcısı. Değer aşağıdaki gibidir:

MQIEP_STRUC_ID

Sürüm

Tip: MQLONG-input

Yapı sürümü numarası. Değer aşağıdaki gibidir:

MQIEP_VERSION_1

Sürüm 1 yapısı sürüm numarası.

MQIEP_CURRENT_VERSION

Yapının yürürlükteki sürümü.

StrucLength

Tip: MQLONG

MQIEP yapısının byte cinsinden büyüklüğü. Değer aşağıdaki gibidir:

MQIEP_LENGTH_1

İşaretler

Tip: MQLONG

İşlev adreslerine ilişkin bilgi sağlar. Kitaplığın bir istemci ya da sunucu kitaplığı olup olmadığını belirtmek için bir işaretle birlikte kitaplığın kullanılıp kullanılmadığını gösteren işaret.

Kitaplık bilgisi belirtmezseniz aşağıdaki değer kullanılır:

MQIEPF_NONE

Paylaşılan kitaplığın iş parçacıklı mı, yoksa iş parçacıklı mı olduğunu belirtmek için aşağıdaki değerlerden biri kullanılır:

MQIEPF_NON_THREADED_LIBRARY

İş parçacığı kullanmayan paylaşılan kitaplık

MQIEPF_THREADED_KITAPLIĞI

İş parçacıklı paylaşılan kitaplık

Paylaşılan kitaplığın bir istemci mi, yoksa sunucu tarafından paylaşılan bir kitaplık mı olduğunu belirtmek için aşağıdaki değerlerden biri kullanılır:

MQIEPF_CLIENT_LIBRARY

İstemci paylaşılan kitaplığı

MQIEPF_LOCAL_LIBRARY

Bir sunucu paylaşılan kitaplığı

Ayrıldı

Tip: MQPTR

MQBACK_Call

Tip: PMQ_BACK_CALL

MQBACK çağrısının adresi.

MQBEGIN_Call

Tip: PMQ_BEGIN_CALL

MQBEGIN çağrısının adresi.

MQBUFMH_Call

Tip: PMQ_BUFMH_CALL

MQBUFMH çağrısının adresi.

MQCB_Call

Tip: PMQ_CB_CALL

MQCB çağrısının adresi.

MQCLOSE_Call

Tip: PMQ_CLOSE_CALL

MQCLOSE çağrısının adresi.

MQCMIT_Call

Tip: PMQ_CMIT_CALL

MQCMIT çağrısının adresi.

MQCONN_Call

Tip: PMQ_CONN_CALL

MQCONN çağrısının adresi.

MQCONNX_Call

Tip: PMQ_CONNX_CALL

MQCONNX çağrısının adresi.

MQRTMH_Call

Tip: PMQ_CRTMH_CALL

MQRTMH çağrısının adresi.

MQCTL_Call

Tip: PMQ_CTL_CALL

MQCTL çağrısının adresi.

MQDISC_Call

Tip: PMQ_DISC_CALL

MQDISC çağrısının adresi.

MQDLTMH_Call

Tip: PMQ_DLTMH_CALL

MQDLTMH çağrısının adresi.

MQDLTMP_Çağrısı

Tip: PMQ_DLTMP_CALL

MQDLTMP çağrısının adresi.

MQGET_Call

Tip: PMQ_GET_CALL

MQGET çağrısının adresi.

MQINQ_Call

Tip: PMQ_INQ_CALL

MQINQ çağrısının adresi.

MQINQMP_Call

Tip: PMQ_INQMP_CALL

MQINQMP çağrısının adresi.

MQMHBUF_Call

Tip: PMQ_MHBUF_CALL

MQMHBUF çağrısının adresi.

MQOPEN_Call

Tip: PMQ_OPEN_CALL

MQOPEN çağrısının adresi.

MQPUT_Call

Tip: PMQ_PUT_CALL

MQPUT çağrısının adresi.

MQPUT1_Call

Tip: PMQ_PUT1_CALL

MQPUT1 çağrısının adresi.

MQSET_Call

Tip: PMQ_SET_CALL

MQSET çağrısının adresi.

MQSETMP_Çağrı

Tip: PMQ_SETMP_CALL

MQSETMP çağrısının adresi.

MQSTAT_Call

Tip: PMQ_STAT_CALL

MQSTAT çağrısının adresi.

MQSUB_Call

Tip: PMQ_SUB_CALL

MQSUB çağrısının adresi.

MQSUBRQ_Call

Tip: PMQ_SUBRQ_CALL

MQSUBRQ çağrısının adresi.

MQXCNCV_Call

Tip: PMQ_XCNCV_CALL

MQXCNCV çağrısının adresi.

MQXCLWLN_Call

Tip: PMQ_XCLWLN_CALL

MQXCLWLN çağrısının adresi.

MQXDX_Call

Tip: PMQ_XDX_CALL

MQXDX çağrısının adresi.

MQXEP_Call

Tip: PMQ_XEP_CALL

MQXEP çağrısının adresi.

MQZEP Çağrısı

Tip: PMQ_ZEP_CALL

MQZEP çağrısının adresi.

C Bildirimi

```
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_CMtT_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_XCNCV_CALL MQXCNCV_Call; /* Address of MQXCNCV */
    PMQ_XDX_CALL  MQXDX_Call;    /* Address of MQXDX */
    PMQ_XEP_CALL  MQXEP_Call;    /* Address of MQXEP */
}
```

```
PMQ_ZEP_CALL      MQZEP_Call;      /* Address of MQZEP */  
};
```

Veri dönüştürme çıkış başvurusu

z/OS için, çevirici dilinde veri dönüştürme çıkışları yazmanız gerekir. Diğer platformlar için C programlama dilini kullanmanız önerilir.

Bir veri dönüştürme çıkış programı yaratmanıza yardımcı olmak için aşağıdaki kaynaklar sağlanır:

- İskelet kaynak dosyası
- Karakter dönüştürme çağrısı
- Veri tipi yapılarında veri dönüştürme gerçekleştiren bir kod parçası oluşturan yardımcı program. Bu yardımcı program yalnızca C girişini alır. z/OS üzerinde, çevirici kodu üretir.

Programların yazılması için aşağıdaki bölümlere bakın:

- **IBM i** [IBM MQ for IBM için veri dönüştürme çıkış programı yazılması](#)
- **z/OS** [IBM MQ for z/OS için veri dönüştürme çıkış programı yazılması](#)
- [IBM MQ for AIX or Linux sistemleri için veri dönüştürme çıkışı yazılması](#)
- [IBM MQ for Windows için veri dönüştürme çıkışı yazılması](#)

İskelet kaynak dosyası

Bunlar, bir veri dönüştürme çıkış programı yazarken başlangıç noktası olarak kullanılabilir.

Sağlanan dosyalar [Çizelge 816 sayfa 1435](#) içinde listelenir.

Çizelge 816. İskelet kaynak dosyaları	
Hizmet olarak sunulan	Dosya
AIX AIX	amqsvfc0.c
IBM i IBM i	QMQMSAMP/QCSRC (AMQSVFC4)
Linux Linux	amqsvfc0.c
Windows Windows sistemleri	amqsvfc0.c
z/OS z/OS	CSQ4BAX8 ("1" sayfa 1435) CSQ4BAX9 ("2" sayfa 1435) CSQ4CAX9 ("3" sayfa 1435)
Notlar: <ol style="list-style-type: none">1. MQXCVNC çağrısı gösterir.2. CICS dışındaki tüm ortamlarda kullanılmak üzere yardımcı program tarafından oluşturulan kod parçaları için bir sarıcı.3. Yardımcı program tarafından CICS ortamında kullanılmak üzere oluşturulan kod parçalarına ilişkin sarıcı.	

Karakter çağrısı dönüştür

Karakter iletisi verilerini bir karakter kümesinden diğerine dönüştürmek için bir veri dönüştürme çıkış programından MQXCNCV (karakterleri dönüştür) çağrısı kullanın. Belirli çok baytlık karakter kümeleri (örneğin, UTF-16 karakter kümeleri) için uygun seçenekler kullanılmalıdır.

Çıkış içinden başka MQI çağrısı yapılamaz; böyle bir çağrı yapma girişimi başarısız olur; neden kodu MQRC_CALL_IN_PROGRESS.

MQXCNCV çağrısı ve uygun seçeneklerle ilgili daha fazla bilgi için bkz. [“MQXCNCV-Karakterlerin dönüştürülme” sayfa 907](#).

Dönüştürme çıkış kodu yaratmak için kullanılan yardımcı program

Dönüştürme çıkış kodu yaratılmasıyla ilgili daha fazla bilgi edinmek için bu bilgileri kullanın.

Dönüştürme çıkış kodu yaratmaya ilişkin komutlar şunlardır:

IBM i **IBM i**

CVTMQMDTA (IBM MQ Veri Tipini Dönüştür)

ALW **AIX, Linux, and Windows sistemleri**

crtmqcvx (Create IBM MQ conversion-exit)

z/OS **z/OS**

CSQUCVX

Altyapınıza ilişkin komut, veri dönüştürme çıkış programınızda kullanılmak üzere veri tipi yapılarında veri dönüştürme gerçekleştiren bir kod parçası üretir. Komut, bir ya da daha çok C dil yapısı tanımı

içeren bir dosyayı alır. **z/OS** Açık z/OS, çevirici kod parçalarını ve dönüştürme işlevlerini içeren bir veri kümesi oluşturur. Diğer platformlarda, her yapı tanımlamasını dönüştürmek için C işlevli bir dosya oluşturur. z/OS işletim sistemlerinde, yardımcı program LE/370 yürütme ortamı kitaplığı SCEERUN' a erişmenizi gerektirir.

z/OS üzerinde CSQUCVX yardımcı programının çağrılması

z/OS

Şekil 10 sayfa 1436 içinde, CSQUCVX yardımcı programını çağırarak için kullanılan JCL örneği gösterilmektedir.

```
//CVX EXEC PGM=CSQUCVX
//STEPLIB DD DISP=SHR,DSN=th1qua1.SCSQANLE
// DD DISP=SHR,DSN=th1qua1.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)
```

Şekil 10. CSQUCVX yardımcı programını çağırarak için kullanılan örnek JCL

z/OS veri tanımlaması deyimleri

z/OS

CSQUCVX yardımcı programı, Çizelge 817 sayfa 1437 içinde gösterilen DD adlarını içeren DD deyimlerini gerektirir:

Çizelge 817. Veri tanımlama deyimi adları ve açıklamaları	
DD deyimi	Açıklama
SYSPRINT	Raporlar ve hata iletileri için bir veri kümesi ya da yazdırma kuyruğa yollama sınıfını belirtir.
CSQUINP	Dönüştürülecek veri yapılarının tanımlamalarını içeren bölümlenmiş veri kümesini belirtir.
CSQUOUT	Dönüştürme kodu parçalarının yazılacağı bölümlenmiş veri kümesini belirtir. Mantıksal kayıt uzunluğu (LRECL) 80 olmalı ve kayıt biçimi (RECFM) FB olmalıdır.

AIX, Linux, and Windows sistemlerinde hata iletileri

`crtmqcvx` komutu, AMQ7953 - AMQ7970 aralığındaki iletileri döndürür.

Bu iletiler [İletiler ve neden kodları IBM MQ İletileri](#) içinde listelenir.

İki ana hata tipi vardır:

- İşleme devam edilemezken sözdizimi hataları gibi önemli hatalar ortaya çıktı.

Ekranında, giriş dosyasındaki hatanın satır numarasını veren bir ileti görüntülenir. Çıkış dosyası kısmen yaratılmış olabilir.

- Bir sorunun bulunduğunu, ancak yapının ayrıştırılmasının devam edebileceğini belirten bir ileti görüntülendiğinde diğer hatalar.

Çıkış dosyası yaratıldı ve oluşan sorunlarla ilgili hata bilgilerini içeriyor. Bu hata bilgisine `#error` öneki eklenir; böylece, üretilen kod, sorunları düzeltmek için müdahale edilmeden hiçbir derleyici tarafından kabul edilmez.

Geçerli sözdizimi

Yardımcı programa ilişkin giriş kütüğünüz C dili sözdizimine uygun olmalıdır.

C ' yi tanımiyorsanız, bu konudaki [C örneğine](#) bakın.

Ayrıca, aşağıdaki kuralları da unutmayın:

- `typedef` yalnızca `struct` anahtar sözcüğünden önce tanınır.
- Yapı bildirimleriniz için bir yapı etiketi gereklidir.
- İletinin sonunda değişken uzunluklu bir diziyi ya da dizgiyi belirtmek için boş köşeli ayraç [] kullanabilirsiniz.
- Çok boyutlu diziler ve diziler desteklenmez.
- Aşağıdaki ek veri tipleri tanınır:
 - MQBOOL
 - MQBYTE
 - MQCHAR
 - MQFLOAT32
 - MQFLOAT64
 - MQSHORT
 - MQLONG
 - MQINT8
 - MQUINT8
 - MQINT16

- MQUINT16
- MQINT32
- MQUINT32
- MQINT64
- MQUINT64

MQCHAR alanları dönüştürülmüş kod sayfasıdır, ancak MQBYTE, MQINT8 ve MQUINT8 'ye dokunulmaz. Kodlama farklıysa, MQSHORT, MQLONG, MQINT16, MQUINT16, MQINT32, MQUINT32, MQINT64, MQUINT64, MQFLOAT32, MQFLOAT64 ve MQBOOL buna göre dönüştürülür.

- Aşağıdaki veri tiplerini kullanmayın:

- çift
- işaretçiler
- bit-alanlar

Bunun nedeni, dönüştürme çıkış kodu yaratma yardımcı programının bu veri tiplerini dönüştürmek için gereken olanağı sağlamamasıdır. Bunu aşmak için, kendi rutinlerinizi yazabilir ve onları çıkıştan çağırabilirsiniz.

Not etmek için diğer noktalar:

- Giriş verileri kümesinde sıra numaralarını kullanmayın.
- Kendi dönüştürme yordamlarınızı sağlamak istediğiniz alanlar varsa, bunları MQBYTE olarak bildirin ve oluşturulan CMQXCFBA makrolarını kendi dönüştürme kodunuzla değiştirin.

C Örneği

```
struct TEST { MQLONG    SERIAL_NUMBER;
              MQCHAR    ID[5];
              MQINT16   VERSION;
              MQBYTE    CODE[4];
              MQLONG    DIMENSIONS[3];
              MQCHAR    NAME[24];
            } ;
```

Bu, diğer programlama dillerinde aşağıdaki bildirimlere karşılık gelir:

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

Yalnızca z/OS üzerinde desteklenir

```
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-Çıkış yayınla

MQ_PUBLISH_EXIT çağrısı, abonelere teslim edilen iletileri inceleyebilir ve değiştirebilir.

Amaç

Abonelere gönderilen iletileri incelemek ve değiştirmek için yayınlama çıkışını kullanın:

- Her bir aboneye yayınlanan bir iletinin içeriğini inceleyin
- Her aboneye yayınlanan bir iletinin içeriğini değiştirme
- İletinin konduğu kuyruğu değiştir
- Bir iletinin aboneye tesliminin durdurulması

Bu çıkış IBM MQ for z/OS üzerinde kullanılamaz.

Sözdizimi

MQ_PUBLISH_EXIT (*ExitParms*, *PubContext*, *SubContext*)

Parametreler

ExitParms (MQPSXP) - Input/Output

ExitParms, çıkışın çağrılmasına ilişkin bilgileri içerir.

PubContext (MQPBC) - Input

PubContext, yayının yayınlayıcısını içeren bağlamsal bilgileri içerir.

SubContext (MQSBC) - Input/Output

SubContext, yayını alan aboneye ilişkin bağlamsal bilgileri içerir.

MQPSXP-Çıkış verileri yapısını yayınla

MQPSXP yapısı, yayınlama çıkışından geçirilen ve döndürülen bilgileri açıklar.

Çizelge 818 sayfa 1439, yapıdaki alanları özetler:

Çizelge 818. MQPSXP 'deki Alanlar	
Alan	Açıklama
<i>StrucID</i>	Yapı tanıtıcısı
<i>Version</i>	Yapı sürümü numarası
<i>ExitId</i>	Çağrılmakta olan çıkışın tipi
<i>ExitReason</i>	Çıkışın çağrılmasının nedeni
<i>ExitResponse</i>	Çıkıştan yanıt
<i>ExitResponse2</i>	Çıkıştan gelen ikincil yanıt
<i>Feedback</i>	Geribildirim kodu

Çizelge 818. MQPSXP 'deki Alanlar (devamı var)	
Alan	Açıklama
<u>ExitUserArea</u>	Kullanıcı alanından çık
<u>ExitData</u>	Verilerden çık
<u>QMgrName</u>	Yerel kuyruk yöneticisinin adı
<u>Hconn</u>	Bağlantı tanıtıcı
<u>MsgDescPtr</u>	İleti tanımlayıcısının (MQMD) adresi
<u>MsgHandle</u>	İleti özelliklerini işle (MQHMSG)
<u>MsgInPtr</u>	Giriş iletisinin adresi
<u>MsgInLength</u>	Giriş iletisinin uzunluğu
<u>MsgOutPtr</u>	Çıkış iletisinin adresi
<u>MsgOutLength</u>	Çıkış iletisinin uzunluğu
<u>pEntryPoints</u>	MQIEP yapısının adresi

Alanlar

StrucID (MQCHAR4)

StrucID yapı tanıtıcısıdır. Değer aşağıdaki gibidir:

MQPSXP_STRUCID

MQPSXP_STRUCID , yayınlama çıkış değiştirgesi yapısının tanıtıcısıdır. C programlama dili için, MQPSXP_STRUC_ID_ARRAY değişmezi de tanımlanır; MQPSXP_STRUC_ID ile aynı değere sahiptir, ancak dizgi yerine bir karakter dizisidir.

StrucID , çıkışa ilişkin bir giriş alanıdır.

Version (MQLONG)

Version , yapı sürümü numarasıdır. Değer aşağıdaki gibidir:

MQPSXP_VERSION_1

MQPSXP_VERSION_1 , Sürüm 1 yayınlama çıkış değiştirgesi yapısıdır. MQPSXP_CURRENT_VERSION değişmezi de aynı değerle tanımlanır.

Version , çıkışa ilişkin bir giriş alanıdır.

ExitId (MQLONG)

ExitId , çağrılmakta olan çıkışın tipidir. Değer aşağıdaki gibidir:

MQXT_PUBLISH_EXIT

Yayınlama çıkışı.

ExitId , çıkışa ilişkin bir giriş alanıdır.

ExitReason (MQLONG)

ExitReason , çıkışı çağırmanın nedenidir. Olası değerler şunlardır:

MQXR_INIT

Kullanıma hazırlama için bu bağlantıya ilişkin çıkış çağrıldı. Çıkış, gereksinim duyduğu kaynakları edinebilir ve başlatabilir; örneğin, ana saklama alanı.

MQXR_TERM

Çıkış durdurulmak üzere olduğu için bu bağlantıya ilişkin çıkış çağrıldı. Çıkış, başlatıldıktan sonra edindiği kaynakları (örneğin, ana saklama alanı) serbest bırakmalıdır.

MQXR_PUBLICATION

Çıkış, bir yayını abonenin ileti kuyruğuna koymadan önce kuyruk yöneticisi tarafından çağrılır. Çıkış iletiyi değiştirebilir, kuyruğa iletiyi yerleştiremez ya da yayını durdurabilir.

ExitReason , çıkışa ilişkin bir giriş alanıdır.

ExitResponse (MQLONG)

İşlemenin nasıl devam etmesi gerektiğini belirtmek için çıkışta *ExitResponse* değerini ayarlayın. *ExitResponse* aşağıdaki değerlerden biridir:

MQXCC_OK

İşlemeye olağan şekilde devam etmek için MQXCC_OK ayarını belirleyin. Herhangi bir *ExitReason* değerine yanıt olarak MQXCC_OK değerini ayarlayın.

ExitReason , MQXR_PUBLICATION değerine sahipse, MQSBC yapısının *DestinationQName* ve *DestinationQMgrName* alanları, iletinin gönderileceği hedefi tanımlar.

MQXCC_FAILED

Yayınlama işlemini durdurmak için MQXCC_FAILED ayarını belirleyin. MQCC_FAILED tamamlanma kodu ve neden kodu 2557 (09FD) (RC2557): MQRC_PUBLISH_EXIT_ERROR çıkıştan döndüğünüzde ayarlanır.

MQXCC_SUPPRESS_FUNCTION

İletinin normal işlenmesini durdurmak için MQXCC_SUPPRESS_FUNCTION ayarını belirleyin. Yalnızca *ExitReason* MQXR_PUBLICATION değerine sahipse MQXCC_SUPPRESS_FUNCTION değerini ayarlayın.

İleti, iletinin ileti tanımlayıcısındaki *Report* alanındaki MQRO_DISCARD_MSG seçeneğine göre kuyruk yöneticisi tarafından işlenmeye devam eder.

- MQRO_DISCARD_MSG seçeneği belirtilirse, ileti aboneye teslim edilmez.
- MQRO_DISCARD_MSG seçeneği belirtilmezse, ileti gitmeyen iletiler kuyruğuna yerleştirilir. Gitmeyen iletiler kuyruğu yoksa ya da ileti, gitmeyen iletiler kuyruğuna başarıyla yerleştirilemezse, yayın aboneye teslim edilmez. Yayının diğer abonelere teslim edilmesi, PMSGDLV ve NPMSGDLV konu nesnesi özniteliklerinin değerlerine bağlıdır. Bu özniteliklere ilişkin açıklamalar için, DEFINE TOPIC komutuna ilişkin değiştirge açıklamalarına bakın.

ExitResponse , çıkıştan bir çıkış alanıdır.

ExitResponse2 (MQLONG)

ExitResponse2 , ileride kullanılmak üzere ayrılmıştır.

Feedback (MQLONG)

Feedback , çıkış *ExitResponse* içinde MQXCC_SUPPRESS_FUNCTION değerini döndürürse kullanılacak geribildirim kodudur.

Çıkışa girişte, *Feedback* her zaman MQFB_NONE değerine sahiptir. Çıkış MQXCC_SUPPRESS_FUNCTION değerini döndürürse, kuyruk yöneticisi iletiyi gelmeyen iletiler kuyruğuna yerleştirdiğinde, ileti için kullanılacak değeri *Feedback* olarak ayarlayın. Çıkıştan döndüğünüzde, *Feedback* özgün MQFB_NONE değerine sahipse, kuyruk yöneticisi *Feedback* değerini MQFB_STOPPED_BY_PUBSUB_EXIT olarak ayarlar.

Feedback , çıkışa ilişkin bir giriş/çıkış alanıdır.

ExitUserArea (MQBYTE16)

ExitUserArea , çıkışın kullanabileceği bir alandır. Her bağlantının ayrı bir *ExitUserArea* vardır. *ExitUserArea* uzunluğu MQ_EXIT_USER_AREA_LENGTH tarafından verilir.

ExitReason alanı, çıkışın ilk çağrısında MQXR_INIT değerini gösterir. *ExitUserArea* , bir bağlantıya ilişkin çıkışın ilk çağrılmasının MQXUA_NONE olarak kullanıma hazırlanır. *ExitUserArea* üzerinde yapılan sonraki değişiklikler, çıkışın çağrılması sırasında korunur.

ExitUserArea , çıkışa ilişkin bir giriş/çıkış alanıdır.

ExitData (MQCHAR32)

ExitData , kuyruk yöneticisinin kullanıma hazırlama kütüğündeki **PublishExitData** değiştirgesiyle tanımlanan değişmez çıkış verileridir. Veriler, alanın tam uzunluğuna kadar boşluklarla doldurulur. Kullanıma hazırlama kütüğünde tanımlı sabit çıkış verisi yoksa, *ExitData* boş olur. *ExitData* uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir.

ExitData , çıkışa ilişkin bir giriş alanıdır.

QMgrName (MQCHAR48)

QMgrName , yerel kuyruk yöneticisinin adıdır. Ad, alanın tam uzunluğuna kadar boşluklarla doldurulur. Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir.

QMgrName , çıkışa ilişkin bir giriş alanıdır.

Hconn (MQHCONN)

Hconn , kuyruk yöneticisine yönelik bir bağlantıyı gösteren tanıtıcıdır. İleti özellikleriyle çalışmak için yalnızca MQSETMP, MQINQMMPya da MQDLTMP ileti özelliği işlev çağrılarında parametre olarak *Hconn* kullanın.

Hconn , çıkışa ilişkin bir giriş alanıdır.

MsgDescPtr (PMQMD)

MsgDescPtr , işlenmekte olan iletinin ileti tanımlayıcısının (MQMD) adresidir ve MQPUT çağrısından döndürülen MQMD 'nin bir kopyasıdır. Çıkış, ileti tanımlayıcısının içeriğini değiştirebilir. İleti tanımlayıcısının içeriğinde yapılacak herhangi bir değişiklik dikkatli yapılmalıdır. Özellikle, MQSBC yapısının *SubType* alanının değeri MQSUBTYPE_PROXY olduğu durumlarda, ileti tanımlayıcısındaki *CorrelId* alanı değiştirilmemelidir.

ExitReason MQXR_INIT ya da MQXR_TERM ise çıkışa ileti tanımlayıcısı iletilmez; Bu durumlarda, *MsgDescPtr* boş değerli işaretçidir.

MsgDescPtr , çıkışa ilişkin bir giriş alanıdır.

MsgHandle (MQHMSG)

MsgHandle , ileti özelliklerinin tanıtıcısıdır. İleti özellikleriyle çalışmak için yalnızca MQSETMP, MQINQMMPya da MQDLTMP ileti özelliği işlev çağrılıyla *MsgHandle* işlevini kullanın.

MsgHandle , çıkışa ilişkin bir giriş alanıdır.

MsgInPtr (PMQVOID)

MsgInPtr , giriş iletisi verilerinin adresidir. *MsgInPtr* tarafından adreslenen arabelleğin içeriği çıkışla değiştirilebilir; bkz. [MsgOutPtr](#) .

MsgInPtr , çıkışa ilişkin bir giriş alanıdır.

MsgInLength (MQLONG)

MsgInLength , çıkışa geçirilen ileti verilerinin bayt cinsinden uzunluğudur. Verilerin adresi *MsgInPtr* tarafından verilir.

MsgInLength , çıkışa ilişkin bir giriş alanıdır.

MsgOutPtr (PMQVOID)

MsgOutPtr , çıkıştan döndürülen ileti verilerini içeren bir arabelleğin adresidir. Çıkışa girişte *MsgOutPtr* boş değerli. Çıkıştan döndüğünüzde, değeri hala boşsa, kuyruk yöneticisi *MsgInPtr* ile belirtilen iletiyi, *MsgInLength* ile belirtilen uzunlukta gönderir.

Çıkış iletisi verilerini değiştirirse, aşağıdaki yordamlardan birini kullanın:

- Verilerin uzunluğu değişmezse, veriler *MsgInPtr* ile adreslenen arabellekte değiştirilebilir. Bu durumda, *MsgOutPtr* ve *MsgOutLength* değiştirmeyin.
- Değiştirilen veriler özgün verilerden kısaysa, veriler *MsgInPtr* tarafından adreslenen arabellekte değiştirilebilir. Bu durumda *MsgOutPtr* , giriş iletisi arabelleğinin adresine ayarlanmalı ve *MsgOutLength* , ileti verilerinin yeni uzunluğuna ayarlanmalıdır.
- Değiştirilen veriler özgün verilerden daha uzunsa ya da daha uzunsa, çıkışın yeni bir ileti arabelleği edinmesi gerekir. Değiştirilen verileri içine kopyalayın. *MsgOutPtr* değerini yeni arabelleğin adresine ayarlayın ve *MsgOutLength* değerini yeni ileti verilerinin uzunluğuna ayarlayın. Çıkış, çıkış bir sonraki çağrıldığında *MsgOutPtr* tarafından adreslenen arabelleği serbest bırakmaktan sorumludur.

Not: *MsgOutPtr* , daha önce elde edilen bir ileti arabelleğinin adresi değil, çıkışa giriş için her zaman boş değerli işaretçidir. Önceden elde edilen arabelleği serbest bırakmak için, çıkışın adresini ve uzunluğunu kaydetmesi gerekir. Bilgileri *ExitUserArea* içine ya da adresini *ExitUserArea* içine kaydeden bir denetim bloğuna kaydedin.

MsgOutPtr , çıkışa ilişkin bir giriş/çıkış alanıdır.

MsgOutLength (MQLONG)

MsgOutLength , çıkışın döndürdüğü ileti verilerinin bayt cinsinden uzunluğudur. Çıkışa girişte bu alan her zaman sıfırdır. Çıkıştan döndüğünüzde, *MsgOutPtr* boş değerliyse bu alan yoksayıdır. İleti verilerinin değiştirilmesine ilişkin bilgi için bkz. *MsgOutPtr* .

MsgOutLength , çıkışa ilişkin bir giriş/çıkış alanıdır.

pEntryPoints (PMQIEP)

pEntryPoints , MQI ve DCI çağrılarının yapılabileceği bir MQIEP yapısının adresidir.

C dil bildirim-MQPSXP

```
typedef struct tagMQPSXP {
    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;         /* 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 */
    PMQMD      MsgDescPtr;       /* Address of message descriptor */
    PMQVOID    MsgInPtr;         /* Address of input message data */
    MQLONG     MsgInLength;      /* Length of input message data */
    PMQVOID    MsgOutPtr;        /* Address of output message data */
    MQLONG     MsgOutLength;     /* Length of output message data */
    /* Ver:1 */
    PMQIEP     pEntryPoints;     /* Address of the MQIEP structure */
    /* Ver:2 */
} MQPSXP;
```

MQPBC-Yayın bağlamı veri yapısı

MQPBC yapısı, yayınlama çıkışına geçirilen yayının yayınlayıcısıyla ilgili bağlamsal bilgileri içerir.

Çizelge 819 sayfa 1443 , yapıdaki alanları özetler:

Çizelge 819. MQPBC içindeki alanlar	
Alan	Açıklama
<u>StrucID</u>	Yapı tanıtıcısı
<u>Version</u>	Yapı sürümü numarası
<u>PubTopicString</u>	Konu dizesini yayınla
<u>MsgDescPtr</u>	İleti tanımlayıcısının (MQMD) adresi

Alanlar

StrucID (MQCHAR4)

StrucID yapı tanıtıcısıdır. Değer aşağıdaki gibidir:

MQPBC_STRUCID

MQPBC_STRUCID , yayın bağlamı yapısının tanıtıcısıdır. C programlama dili için, MQPBC_STRUC_ID_ARRAY değişmezi de tanımlanır; MQPBC_STRUC_ID ile aynı değere sahiptir, ancak dizgi yerine bir karakter dizisidir.

StrucID , çıkışa ilişkin bir giriş alanıdır.

Version (MQLONG)

Version , yapı sürümü numarasıdır. Değer aşağıdaki gibidir:

MQPBC_VERSION_1

MQPBC_VERSION_1 , Sürüm 1 yayınlama çıkış değıştirgesi yapısıdır.

MQPBC_VERSION_2

MQPBC_VERSION_2 , Sürüm 2 yayınlama çıkış değıştirgesi yapısıdır. MQPBC_CURRENT_VERSION değışmezi de aynı değlerle tanımlanır.

Version , çıkışa ilişkin bir giriş alanıdır.

PubTopicString (MQCHARV)

PubTopicString , yayınlanmakta olan konu dizgisidir.

PubTopicString , çıkışa ilişkin bir giriş alanıdır.

MsgDescPtr (PMQMD)

MsgDescPtr , işlenmekte olan iletiye ilişkin ileti tanımlayıcısının (MQMD) bir kopyasının adresidir.

MsgDescPtr , çıkışa ilişkin bir giriş alanıdır.

C dil bildirimi-MQPBC

```
typedef struct tagMQPBC {
    MQCHAR4      StructId;          /* Structure identifier */
    MQLONG       Version;          /* Structure version number */
    MQCHARV      PubTopicString;   /* Publish topic string */
    PMQMD        MsgDescPtr;      /* Address of message descriptor */
} MQPBC;
```

MQSBC-Abonelik bağlamı veri yapısı

MQSBC yapısı, yayınlama çıkışına geçirilen, yayını alan aboneyle ilgili bağlamsal bilgileri içerir.

Çizelge 820 sayfa 1444 , yapıdaki alanları özetler:

Çizelge 820. MQSBC içindeki alanlar	
Alan	Açıklama
<u>StructID</u>	Yapı tanıtıcısı
<u>Version</u>	Yapı sürümü numarası
<u>DestinationQMgrName</u>	Hedef kuyruk yöneticisinin adı
<u>DestinationQName</u>	Hedef kuyruğun adı
<u>SubType</u>	Abonelik tipi
<u>SubOptions</u>	Abonelik seçenekleri
<u>ObjectName</u>	Nesne adı
<u>ObjectString</u>	Nesne dizgisi
<u>SubTopicString</u>	Abonelik konusu dizgisi
<u>SubName</u>	Abonelik adı
<u>SubId</u>	Abonelik tanıtıcısı
<u>SelectionString</u>	Seçim dizgisinin adresi
<u>SubLevel</u>	Abonelik düzeyi
<u>PSPProperties</u>	Özellikleri yayınl/abone ol

Alanlar

StrucID (MQCHAR4)

Yapı tanıtıcısı. Değer aşağıdaki gibidir:

MQSBC_STRUCID

MQSBC_STRUCID , yayınlama çıkış değiştirgesi yapısının tanıtıcısıdır. C programlama dili için, MQSBC_STRUC_ID_ARRAY sabiti de tanımlanır; MQSBC_STRUC_ID_ARRAY , MQSBC_STRUC_ID ile aynı değere sahiptir, ancak bir dizgi yerine bir karakter dizisidir.

StrucID , çıkışa ilişkin bir giriş alanıdır.

Version (MQLONG)

Yapı sürümü numarası. Değer aşağıdaki gibidir:

MQSBC_VERSION_1

Sürüm 1 yayınlama çıkış değiştirgesi yapısı. MQSBC_CURRENT_VERSION değışmezi de aynı değlerle tanımlanır.

Version , çıkışa ilişkin bir giriş alanıdır.

DestinationQMgrName (MQCHAR48)

DestinationQMgrName , iletinin gönderildiği kuyruk yöneticisinin adıdır. Ad, alanın tam uzunluğuna kadar boşluklarla doldurulur. Ad, çıkışla değıştirilebilir. Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir.

DestinationQMgrName , çıkışa ilişkin bir giriş/çıkış alanıdır; bkz. [not](#).

DestinationQName (MQCHAR48)

DestinationQName , iletinin gönderildiği kuyruğun adıdır. Ad, alanın tam uzunluğuna kadar boşluklarla doldurulur. Ad, çıkışla değıştirilebilir. Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

DestinationQName , çıkışa ilişkin bir giriş/çıkış alanıdır; bkz. [not](#).

SubType (MQLONG)

SubType , aboneliğin nasıl oluşturulduğunu gösterir. Geçerli değler şunlardır: MQSUBTYPE_API, MQSUBTYPE_ADMIN ve MQSUBTYPE_PROXY ; Bkz. [Abonelik Durumunu Sor \(Yanıt\)](#).

SubType , çıkışa ilişkin bir giriş alanıdır.

SubOptions (MQLONG)

SubOptions abonelik seçenekleridir; bu alanın alabileceği değlerin açıklaması için bkz. "[MQSD için seçenekler \(MQLONG\)](#)" sayfa 566 .

SubOptions , çıkışa ilişkin bir giriş alanıdır.

ObjectName (MQCHAR48)

ObjectName , yerel kuyruk yöneticisinde tanımlandığı şekilde, konu nesnesinin adıdır. Bu alanın uzunluğu MQ_TOPIC_NAME_LENGTH tarafından verilir. Nesne adı, kuyruk yöneticisinin konu dizgisiyle ilişkilendirdiği denetim konusu nesnesinin adıdır. Abone, aboneliğin bir parçası olarak bir konu nesnesi sağlasa da, *ObjectName* farklı bir konu nesnesi olabilir. Bir konu nesnesinin abonelikle ilişkilendirilmesi, *SubTopicString*' in tam çözümüne bağlıdır.

ObjectName , çıkışa ilişkin bir giriş alanıdır.

ObjectString (MQCHARV)

ObjectString , abone olunan yayının tam konu dizgisidir. Özgün abonelik dizgisindeki genel arama karakterleri çözülür. Bu, "[MQSD için ObjectString \(MQCHARV\)](#)" sayfa 576 içinde açıklanan MQSD abonelik *ObjectString* alanından farklıdır; bu alan genel arama karakterleri içerebilir ve abone tarafından sağlanan herhangi bir nesne adını içermeyebilir.

ObjectString , çıkışa ilişkin bir giriş alanıdır.

SubTopicString (MQCHARV)

SubTopicString , abone tarafından sağlanan tam konu dizgisidir. *SubTopicString* , bir konu nesnesinde tanımlanan konu dizisi ile bir konu dizgisinin birleşimidir. Bir abonenin bir konu nesnesi,

konu dizgisi ya da her ikisini birden sağlaması gerekir. Abone bir konu dizgisi sağlarsa, genel arama karakterleri içerebilir.

SubTopicString , çıkışa ilişkin bir giriş alanıdır.

SubName (MQCHARV)

SubName , abone tarafından sağlanan ya da oluşturulan bir addir.

SubName , çıkışa ilişkin bir giriş alanıdır.

SubId (MQBYTE 24)

SubId benzersiz iç abonelik tanıtıcısıdır.

SubId , çıkışa ilişkin bir giriş alanıdır.

SelectionString (MQCHARV)

SelectionString , bir konudaki iletilere abone olurken kullanılan seçim ölçütüdür; bkz. [Seçiciler](#).

SelectionString , çıkışa ilişkin bir giriş alanıdır.

SubLevel (MQLONG)

SubLevel , abonelik ile ilişkili önleme düzeyidir; daha fazla ayrıntı için bkz. [“MQSD için SubLevel \(MQLONG\)” sayfa 579](#) .

SubLevel , çıkışa ilişkin bir giriş alanıdır.

PSPProperties (MQLONG)

PSPProperties , yayınlama/abone olma özellikleridir. Bunlar, bu aboneliğe gönderilen iletilere yayınlama/abone olma ile ilgili ileti özelliklerinin nasıl ekleneceğini belirtir. Olası değerler şunlardır: MQPSPROP_NONE, MQPSPROP_COMPAT, MQPSPROP_RFH2, MQPSPROP_MSGPROP. Bu değerlerin açıklaması için bkz. [İsteğe bağlı parametreler \(Değiştir, Kopyala ve Abonelik Oluştur\)](#) .

PSPProperties , çıkışa ilişkin bir giriş alanıdır.

Not: Yetkilendirme denetimleri, yayınlama çıkışına geçirilmeden önce yalnızca *DestinationQMgrName* ve *DestinationQName* özgün değerlerinde gerçekleştirilir. Çıkış, *DestinationQMgrName* ya da *DestinationQName* değiştirilerek hedef kuyruğu değiştirdiğinde yeni yetki denetimi gerçekleştirilmez.

C dili bildirim-MQSBC

```
typedef struct tagMQSBC {
    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    PSPProperties;      /* Publish/subscribe properties */
} MQSBC;
```

Kanal çıkışı çağrıları ve veri yapıları

Bu konu derlemi, kanal çıkış programları yazarken kullanabileceğiniz özel IBM MQ çağrılarına ve veri yapılarına ilişkin başvuru bilgileri sağlar.

Bu bilgiler, ürüne duyarlı programlama arabirimi bilgileridir. IBM MQ kullanıcı çıkışlarını aşağıdaki programlama dillerinde yazabilirsiniz:

Çizelge 821. IBM MQ kullanıcı çıkışları: platformlar ve programlama dilleri	
Hizmet olarak sunulan	Programlama dilleri
IBM MQ for z/OS	Çevirici ve C (z/OS C/C++ Programlama Kılavuzu' nda açıklanan sistem çıkışları için C sistem programlama ortamına uyması gerekir.)
IBM MQ for IBM i	ILE C, ILE COBOL ve ILE RPG
Diğer tüm IBM MQ platformları	C

Yalnızca Java ve JMS uygulamalarıyla kullanmak üzere Java içinde kullanıcı çıkışlarını da yazabilirsiniz. IBM MQ classes for Java ile kanal çıkışları oluşturma ve kullanma hakkında daha fazla bilgi için bkz. [IBM MQ classes for Java içinde kanal çıkışlarının kullanılması](#) ve IBM MQ classes for JMS için bkz. [IBM MQ classes for JMS ile kanal çıkışlarının kullanılması](#).

TAL ya da Visual Basic 'te IBM MQ kullanıcı çıkışlarını yazamazsınız. Ancak, Visual Basic 'de bir IBM MQ MQI client programından MQCONNX çağrısında kullanılmak üzere MQCD yapısına ilişkin bir bildirim sağlanır.

Aşağıdaki açıklamalarda yer alan bazı durumlarda, parametreler değişmez büyüklüklü diziler ya da karakter dizilimleridir. Bu değiştirgelerde, sayısal bir değışımezi göstermek için küçük harf "n" kullanılır. Bu parametreye ilişkin bildirim kodlandığında, "n" , gerekli sayısal değerle değiştirilmelidir. Bu açıklamalarda kullanılan kurullarla ilgili daha fazla bilgi için bkz. [“Temel veri tipleri” sayfa 234](#).

Veri tanımlama dosyaları

Veri tanımlama dosyaları, desteklenen programlama dillerinin her biri için IBM MQ ile birlikte sağlanır. Bu dosyaların ayrıntıları için bkz. [Copy, header, include, and module files](#).

MQ_CHANNEL_EXIT-Kanal çıkışı

MQ_CHANNEL_EXIT çağrısı, Message Channel Agent tarafından çağrılan kanal çıkışlarının her birine geçirilen parametreleri açıklar.

Kuyruk yöneticisi tarafından MQ_CHANNEL_EXIT adlı bir giriş noktası sağlanmadı; MQCD kanal tanımlamasında kanal çıkışlarının adları sağlandığından MQ_CHANNEL_EXIT adı özel bir anlam taşıyor.

Beş tip kanal çıkışı vardır:

- Kanal güvenliği çıkışı
- Kanal ileti çıkışı
- Kanal gönderme çıkışı
- Kanal alma çıkışı
- Kanal iletisi-çıkışı yeniden dene

Parametreler her çıkış tipi için benzerdir ve burada verilen açıklama, özellikle belirtildiği durumlar dışında, hepsi için geçerlidir.

Sözdizimi

MQ_CHANNEL_EXIT (ChannelExitParms, ChannelDefinition, DataLength, AgentBufferLength, AgentBuffer, ExitBufferLength, ExitBufferAddr)

Parametreler

MQ_CHANNEL_EXIT çağrısında aşağıdaki parametreler var.

ChannelExitDeğiştirgeleri (MQCXP)-giriş/çıkış

Kanal çıkış parametresi bloğu.

Bu yapı, çıkışın çağrılmasıyla ilgili ek bilgiler içerir. Çıkış, MCA 'nın nasıl ilerlediğini belirtmek için bu yapıdaki bilgileri ayarlar.

ChannelDefinition (MQCD)-giriş/çıkış

Kanal tanımı.

Bu yapı, kanalın davranışını denetlemek için yönetici tarafından ayarlanan parametreleri içerir.

DataLength (MQLONG)-giriş/çıkış

Verilerin uzunluğu.

Veriler, çıkışın tipine bağlıdır:

- Bir kanal güvenliği çıkışı için, çıkış çağrıldığında, bu değiştirge *AgentBuffer* alanındaki herhangi bir güvenlik iletilerinin uzunluğunu (*ExitReason* MQXR_SEC_MSG ise) içerir. İleti yoksa sıfırdır. Çıkış, *ExitResponse* değerini MQXCC_SEND_SEC_MSG ya da MQXCC_SEND_AND_REQUEST_SEC_MSG olarak ayarlarsa, bu alanı ortağına gönderilecek güvenlik iletilerinin uzunluğuna ayarlamalıdır. İleti verileri *AgentBuffer* ya da *ExitBufferAddr* içinde yer almalıdır.

Güvenlik iletilerinin içeriği yalnızca güvenlik çıkışlarının sorumluluğundadır.

- Bir kanal ileti çıkışı için, çıkış çağrıldığında bu parametre iletilerin uzunluğunu (iletim kuyruğu üstbilgisi de içinde olmak üzere) içerir. Çıkış, devam etmek için bu alanı *AgentBuffer* ya da *ExitBufferAddr* içindeki iletilerin uzunluğuna ayarlamalıdır. Bu değer, iletim kuyruğu üstbilgisinin uzunluğundan (MQXQH) büyük ya da ona eşit olmalıdır.
- Kanal gönderme ya da kanal alma çıkışı için, çıkış çağrıldığında bu parametre iletim uzunluğunu içerir. Çıkış, devam etmek için bu alanı *AgentBuffer* ya da *ExitBufferAddr* içindeki iletim uzunluğuna ayarlamalıdır.

Bir güvenlik çıkışı bir ileti gönderirse ve kanalın diğer ucunda güvenlik çıkışı yoksa ya da diğer uç MQXCC_OK *ExitResponse* değerini ayarlarsa, başlatma çıkışı MQXR_SEC_MSG ve boş değerli bir yanıtla (*DataLength* = 0) yeniden çağrılır.

AgentBufferUzunluğu (MQLONG)-giriş

Aracı arabelleğinin uzunluğu.

Bu parametre, çağırma sırasında *DataLength* değerinden büyük olabilir.

Kanal ileti, gönderme ve alma çıkışları için, başlatma sırasında kullanılmayan herhangi bir alan, çıkış tarafından yerinde verileri genişletmek için kullanılabilir. Bu yapıldıysa, **DataLength** parametresi çıkış tarafından uygun şekilde ayarlanmalıdır.

C programlama dilinde, bu parametre adres tarafından iletilir.

AgentBuffer (MQBYTE x AgentBufferUzunluğu)-giriş/çıkış

Ajan arabelleği.

Bu parametrenin içeriği çıkış tipine bağlıdır:

- Bir kanal güvenliği çıkışı için, çıkış çağrıldığında *ExitReason* MQXR_SEC_MSG ise bir güvenlik ileti içerir. Bir güvenlik iletilerini geri göndermek için, çıkış bu arabelleği ya da kendi arabelleğini kullanabilir (*ExitBufferAddr*).

- Kanal ileti çıkışı için, çıkış çağrıldığında bu değiştirge şunları içerir:

- İleti tanımlayıcıyı (kendisi iletiye ilişkin bağlam bilgilerini içerir) içeren iletim kuyruğu üstbilgisi (MQXQH) hemen ardından
- İleti verileri

İleti devam edilecekse, çıkış aşağıdakilerden birini yapabilir:

- Arabelleğin içeriğine dokunmayın
- İçeriğin değiştirilmesi (*DataLength* içindeki verilerin yeni uzunluğunun döndürülmesi); bu, *AgentBufferLength* değerinden büyük olmamalıdır

– İÇeriĐi *ExitBufferAddr*' e kopyalayarak gerekli deĐiřiklikleri yapın

Çıkışın iletim kuyruĐu üstbilgisinde yaptıĐı deĐiřiklikler denetlenmez; ancak, hatalı deĐiřiklikler iletinin hedefe konamayacaĐı anlamına gelebilir.

- Bir kanal gönderme ya da alma çıkışı için, çıkış çağrıldığında bu, iletim verilerini içerir. Çıkış ařaĐıdakilerden birini yapabilir:

– ArabelleĐin iÇeriĐine dokunmayın

– İÇeriĐin deĐiřtirilmesi (*DataLength* içindeki verilerin yeni uzunluĐunun döndürülmesi); bu, *AgentBufferLength*deĐerinden büyük olmamalıdır

– İÇeriĐi *ExitBufferAddr*' e kopyalayarak gerekli deĐiřiklikleri yapın

Verilerin ilk 8 baytı çıkış tarafından deĐiřtirilmemelidir.

ExitBufferUzunluĐu (MQLONG)-giriř/çıkış

Çıkış arabelleĐinin uzunluĐu.

Çıkışın ilk çağrısında bu parametre sıfır olarak ayarlanır. Bundan sonra, çıkıştan geÇirilen her deĐer, her çağrıda, bir sonraki çağrılıřında çıkışa sunulur. DeĐer, MCA tarafından kullanılmıyor.

Not: Bu parametre, gösterge veri tipini desteklemeyen programlama dillerinde yazılan çıkışlar tarafından kullanılmamalıdır.

ExitBufferAddr (MQPTR)-giriř/çıkış

Çıkış arabelleĐinin adresi.

Bu parametre, çıkış tarafından yönetilen bir saklama alanı arabelleĐinin adresine iliřkin bir iřaretÇidir; burada, aracının arabelleĐi yeterli büyükse ya da büyükse, ileti ya da iletim verilerini (çıkış tipine baĐlı olarak) aracıya döndürmeyi seÇebilir ya da çıkışın bunu yapması için daha uygun olabilir.

Çıkışın ilk çağrısında, çıkışa geÇirilen adres boş deĐerli. Bundan sonra, çıkıştan geÇirilen her adres, her çağrıda, bir sonraki çağrılıřında çıkışa sunulur.

ExitBufferAddr boş deĐerliyse, kullanılan veriler AgentBuffer deĐiřtirgesinden alınır.

ExitBufferAddr boş deĐerli deĐilse, kullanılan veriler ExitBufferAddr deĐiřtirgesinin gösterdiĐi arabellekten alınır.

Not: Bu parametre, iřaretÇi veri tipini desteklemeyen programlama dillerinde yazılan çıkışlar tarafından kullanılmamalıdır.

C çağrısı

```
exitname (&ChannelExitParms, &ChannelDefinition,  
&DataLength, &AgentBufferLength, AgentBuffer,  
&ExitBufferLength, &ExitBufferAddr);
```

Çıkışa geÇirilen deĐiřtirgeler ařaĐıdaki gibi bildirilir:

```
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 */
```

COBOL çağrısı

```
CALL 'exitname' USING CHANNELEXITPARMS, CHANNELDEFINITION,  
DATALENGTH, AGENTBUFFERLENGTH, AGENTBUFFER,  
EXITBUFFERLENGTH, EXITBUFFERADDR.
```

Çıkışa geçirilen değıştirmeler ařađıdaki gibi bildirilir:

```
** Channel exit parameter block
01 CHANNEEXITPARMS.
   COPY CMQXCPV.
** 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.
```

RPG çağırısı (ILE)

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP      exitname(MQXCP : MQCD : DATLEN :
C                               ABUFL : ABUF : EBUFL :
C                               EBUF)
```

Çađrıya iliřkin prototip tanımlaması:

```
D*.1.....2.....3.....4.....5.....6.....7..
Dexitname      PR          EXTPROC('exitname')
D* Channel exit parameter block
D MQXCP                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                *
```

System/390 çevirici çağırısı

```
CALL EXITNAME, (CHANNEEXITPARMS, CHANNELDEFINITION, DATALENGTH, X
                AGENTBUFFERLENGTH, AGENTBUFFER, EXITBUFFERLENGTH, X
                EXITBUFFERADDR)
```

Çıkışa geçirilen değıştirmeler ařađıdaki gibi bildirilir:

```
CHANNEEXITPARMS  CMQXCPA  ,      Channel exit parameter block
CHANNELDEFINITION CMQCD   ,      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
```

Kullanım notları

1. Kanal çıkışı tarafından gerekleřtirilen iřlev, çıkışın sađlayıcısı tarafından tanımlanır. Ancak, çıkışın burada ve iliřkili denetim öbeđinde (MQXCP) tanımlanan kurallara uyması gerekir.

2. Kanal çıkışına geçirilen **ChannelDefinition** parametresi birkaç sürümden biri olabilir. Ek bilgi için MQCD yapısındaki *Version* alanına bakın.
3. Kanal çıkışı, *Version* alanı MQCD_VERSION_1 değerinden büyük bir değere ayarlanmış bir MQCD yapısı alırsa, çıkış *ShortConnectionName* alanı yerine MQCD ' deki *ConnectionName* alanını kullanmalıdır.
4. Genel olarak, kanal çıkışlarının ileti verilerinin uzunluğunu değiştirmesine izin verilir. Bu, çıkışın iletiye veri eklemesinin ya da iletiden veri kaldırmasının ya da iletiyi sıkıştırmasının ya da şifrelemesinin sonucu olarak ortaya çıkabilir. Ancak, ileti mantıksal iletinin yalnızca bir kısmını içeren bir bölümse, özel kısıtlamalar geçerlidir. Özellikle, tamamlayıcı gönderim ve alma çıkışlarının eylemlerinin bir sonucu olarak mesajın uzunluğu içinde net bir değişiklik olmamalıdır.

Örneğin, bir gönderme çıkışının iletiyi sıkıştırarak kısaltmasına izin verilebilir, ancak tamamlayıcı alıcı çıkışın, iletinin sıkıştırmasını açarak iletinin özgün uzunluğunu geri yüklemesi gerekir, böylece ileti uzunluğu boyunca net bir değişiklik olmaz.

Bu kısıtlama, bir kesimin uzunluğunun değiştirilmesinin iletideki sonraki kesimlerin görece konumlarının yanlış olmasına neden olacağı ve kuyruk yöneticisinin bölümlerin tam bir mantıksal ileti oluşturduğunu fark etme yeteneğini engelleyebilmesi nedeniyle ortaya çıkar.

MQ_CHANNEL_AUTO_DEF_EXIT-Kanal otomatik tanımlama çıkışı

MQ_CHANNEL_AUTO_DEF_EXIT çağrısı, Message Channel Agent tarafından çağrılan kanal otomatik tanımlama çıkışına geçirilen parametreleri açıklar.

Kuyruk yöneticisi tarafından MQ_CHANNEL_AUTO_DEF_EXIT adlı bir giriş noktası sağlanmadı; kuyruk yöneticisinde otomatik tanımlama çıkışlarının adları sağlandığından, MQ_CHANNEL_AUTO_DEF_EXIT adı özel bir önem taşıyor.

Sözdizimi

MQ_CHANNEL_AUTO_DEF_EXIT (ChannelExitParms, ChannelDefinition)

Parametreler

MQ_CHANNEL_AUTO_DEF_EXIT çağrısında aşağıdaki parametreler var.

ChannelExitDeğiştiricileri (MQCXP)-giriş/çıkış

Kanal çıkış parametresi bloğu.

Bu yapı, çıkışın çağrılmasıyla ilgili ek bilgiler içerir.Çıkış, MCA ' nın nasıl ilerlediğini belirtmek için bu yapıdaki bilgileri ayarlar.

ChannelDefinition (MQCD)-giriş/çıkış

Kanal tanımı.

Bu yapı, otomatik olarak oluşturulan kanalların davranışını denetlemek için yönetici tarafından ayarlanan parametreleri içerir.Çıkış, yönetici tarafından ayarlanan varsayılan davranışı değiştirmek için bu yapıdaki bilgileri ayarlar.

Listelenen MQCD alanları çıkışla değiştirilmemelidir:

- *ChannelName*
- *ChannelType*
- *StrucLength*
- *Version*

Diğer alanlar değiştirilirse, çıkışın belirlediği değer geçerli olmalıdır. Değer geçerli değilse, hata günlüğü dosyasına bir hata iletisi yazılır ya da konsolda görüntülenir (ortam için uygun olduğu şekilde).



Uyarı: Kanal otomatik tanımlama (CHAD) çıkışı tarafından oluşturulan otomatik tanımlı kanallar, kanal oluşturulduğunda TLS anlaşması gerçekleştiği için sertifika etiketini ayarlayamaz. Gelen kanallar için CHAD çıkışında sertifika etiketinin ayarlanması etkili olmaz.

C çağırısı

```
exitname (&ChannelExitParms, &ChannelDefinition);
```

Çıkışa geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
MQ_CXP ChannelExitParms; /* Channel exit parameter block */
MQ_CD ChannelDefinition; /* Channel definition */
```

COBOL çağırısı

```
CALL 'exitname' USING CHANNELEXITPARMS, CHANNELDEFINITION.
```

Çıkışa geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
** Channel exit parameter block
01 CHANNELEXITPARMS.
   COPY CMQ_CXPV.
** Channel definition
01 CHANNELDEFINITION.
   COPY CMQ_CDV.
```

RPG çağırısı (ILE)

```
C*.1.....2.....3.....4.....5.....6.....7..
C          CALLP          exitname(MQ_CXP : MQ_CD)
```

Çağrıya ilişkin prototip tanımlaması:

```
D*.1.....2.....3.....4.....5.....6.....7..
Dexitname PR          EXTPROC('exitname')
D* Channel exit parameter block
D MQ_CXP          160A
D* Channel definition
D MQ_CD          1328A
```

System/390 çevirici çağırısı

```
CALL EXITNAME, (CHANNELEXITPARMS, CHANNELDEFINITION)
```

Çıkışa geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
CHANNELEXITPARMS CMQ_CXPA , Channel exit parameter block
CHANNELDEFINITION CMQ_CDA , Channel definition
```

Kullanım notları


1. Kanal çıkışı tarafından gerçekleştirilen işlev, çıkışın sağlayıcısı tarafından tanımlanır. Ancak, çıkışın burada ve ilişkili denetim öbeğinde (MQ_CXP) tanımlanan kurallara uyması gerekir.

2. Kanal otomatik tanımlama çıkışına geçirilen **ChannelExitParms** değıştirgesi bir MQCXP yapısıdır. Geçirilen MQCXP sürümü, çıkışın çalıştığı ortama bağlıdır; ayrıntılar için [“MQCXP-Kanal çıkış değıştirgesi” sayfa 1493](#) içindeki *Version* alanının tanımına bakın.
3. Kanal otomatik tanımlama çıkışına geçirilen **ChannelDefinition** değıştirgesi bir MQCD yapısıdır. Geçirilen MQCD sürümü, çıkışın çalıştığı ortama bağlıdır; ayrıntılar için [“MQCD-Kanal tanımlaması” sayfa 1454](#) içindeki *Version* alanının tanımına bakın.

MQXWAIT-Çıkışta bekle

MQXWAIT çağırısı bir olayın oluşmasını bekler. Yalnızca z/OS üzerindeki bir kanal çıkışından kullanılabilir.

MQXWAIT kullanımı, kanal çıkışı beklemeye neden olacak bir şey yaparsa oluşabilecek performans sorunlarının önlenmesine yardımcı olur. MQXWAIT ' ın beklediği olay bir MVS ECB (olay denetim öbeği) tarafından işaretli. ECB, MQXWD denetim öbeği tanımında açıklanmıştır.

 MQXWAIT kullanımı ve kanal çıkış programlarının yazılması hakkında daha fazla bilgi için bkz. [z/OS üzerinde kanal çıkış programlarının yazılması](#)

Sözdizimi

MQXWAIT (Hconn, WaitDesc, CompCode, Reason)

Parametreler

MQXWAIT çağırısında aşağıdaki değıştirgeler var.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

Bu tanıtıcı, kuyruk yöneticisine yönelik bağlantıyı gösterir. *Hconn* değeri, çıkışın aynı ya da daha önceki çağırılmasında yayınlanan önceki bir MQCONN çağırısı tarafından döndürüldü.

WaitDesc (MQXWD)-giriş/çıkış

Bekleme tanımlayıcısı.

Bu parametre, beklenecek olayı açıklar. Bu yapıdaki alanların ayrıntıları için bkz. [“MQXWD-Çıkış bekleme tanımlayıcısı” sayfa 1507](#).

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdaki kodlardan biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

MQRC_ADAPTER_NOT_KULLANILAMIYOR

(2204, X'89C') Bağdaştırıcı kullanılmıyor.

MQRC_OPTIONS_HATA

(2046, X'7FE') Seçenekler geçerli değil ya da tutarlı değil.

MQRC_XWAIT_IPTAL edildi

(2107, X'83B') MQXWAIT çağrısı iptal edildi.

MQRC_XWAIT_HATASI

(2108, X'83C') MQXWAIT çağrısının çağrılması geçersiz.

C çağrısı

```
MQXWAIT (Hconn, &WaitDesc, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
MQHCNN  Hconn;      /* Connection handle */
MQXWD   WaitDesc;   /* Wait descriptor */
MQLONG  CompCode;   /* Completion code */
MQLONG  Reason;     /* Reason code qualifying CompCode */
```

System/390 çevirici çağrısı

```
CALL MQXWAIT, (HCONN, WAITDESC, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
HCONN    DS      F  Connection handle
WAITDESC CMQXWDA ,  Wait descriptor
COMPCODE DS      F  Completion code
REASON   DS      F  Reason code qualifying COMPCODE
```

MQCD-Kanal tanımlaması

MQCD yapısı, bir kanalın yürütülmesini denetleyen parametreleri içerir. MCA ' dan (Message Channel Agent; İleti Kanal Aracısı) çağrılan her kanal çıkışına geçirilir.

Kanal çıkışları hakkında daha fazla bilgi için bkz. “MQ_CHANNEL_EXIT-Kanal çıkışı” sayfa 1447. Bu konudaki açıklama hem ileti kanallarıyla hem de MQI kanallarıyla ilgilidir.

Ad alanlarından çık

Bir çıkış çağrıldığında, *SecurityExit*, *MsgExit*, *SendExit*, *ReceiveExit* ve *MsgRetryExit* içindeki ilgili alan, çağrılmakta olan çıkışın adını içerir. Bu alanlardaki adın anlamı, MCA ' nın çalıştığı ortama bağlıdır. Not edilenler dışında, ad alan içinde sola hizalanır ve boşluk bırakılmaz; ad, alanın uzunluğuna kadar boşluklarla doldurulur. Aşağıdaki açıklamalarda, köşeli ayraç ([]) isteğe bağlı bilgileri gösterir:

AIX and Linux

Çıkış adı, dinamik olarak yüklenebilir bir modülün ya da kitaplığın adıdır; soneki o kitaplıkta bulunan bir işlevin adıdır. İşlev adı ayraç içine alınmalıdır. Kitaplık adının başına isteğe bağlı olarak bir dizin yolu konabilir:

```
[ path ] library ( function )
```

Ad en çok 128 karakterle sınırlıdır.

z/OS

Çıkış adı, LINK ya da LOAD makrosunun EP parametresindeki belirtiler için geçerli olan bir yükleme modülünün adıdır. Ad en çok sekiz karakterle sınırlıdır.

Windows

Çıkış adı, bu kitaplıkta bulunan bir işlevin adının eklendiği dinamik bağlantı kitaplığının adıdır. İşlev adı araç içine alınmalıdır. Kitaplık adının başına isteğe bağlı olarak bir dizin yolu ve sürücü öneki konabilir:

```
[d:][ path ] library ( function )
```

Ad en çok 128 karakterle sınırlıdır.

IBM i

Çıkış adı, 10 baytlık bir program adı ve ardından 10 baytlık bir kitaplık adıdır. Adların uzunluğu 10 bayttan azsa, her ad 10 bayt olacak şekilde boşluklarla doldurulur. Bir kanal otomatik tanımlama çıkışının çağrılması dışında, kitaplık adı *LIBL olabilir; bu durumda tam olarak nitelenmiş bir ad gerekir.

Kanal çıkışında MQCD alanlarının değiştirilmesi

Bir kanal çıkışı MQCD ' deki alanları değiştirebilir. Değiştirilen değer MQCD ' de kalır ve bir çıkış zincirinde ve kanal eşgörünümünü paylaşan etkileşimde kalan çıkışlara geçerilir. Değiştirilen MQCD, MCA tarafından kanalın devam eden yaşam süresi boyunca olağan işlemesi için de kullanılır.

Aşağıdaki MQCD alanları çıkışla değiştirilmemelidir:

- ChannelName
- ChannelType
- StrucLength
- Sürüm

İlgili başvurular

[“Alanlar” sayfa 1455](#)

Bu konuda, MQCD yapısındaki tüm alanlar listelenir ve her alan açıklanır.

[“C bildirim” sayfa 1480](#)

Bu bildirim, MQCD yapısına ilişkin C bildirimini belirtir.

[“COBOL bildirim” sayfa 1482](#)

Bu bildirim, MQCD yapısına ilişkin COBOL bildirimini belirtir.

[“RPG bildirim \(ILE\)” sayfa 1485](#)

Bu bildirim, MQCD yapısına ilişkin RPG bildirimini belirtir.

[“System/390 çevirici bildirim” sayfa 1487](#)

Bu bildirim, MQCD yapısına ilişkin System/390 çevirici bildirimini gösterir.

[“Visual Basic bildirim” sayfa 1489](#)

Bu bildirim, MQCD yapısının Visual Basic bildirimini belirtir.

[“Kanal çıkışında MQCD alanlarının değiştirilmesi” sayfa 1490](#)

Bir kanal çıkışı MQCD ' deki alanları değiştirebilir. Ancak, listelenen koşullar dışında, bu değişiklikler genellikle üzerinde işlem yapılmaz.

Alanlar

Bu konuda, MQCD yapısındaki tüm alanlar listelenir ve her alan açıklanır.

BatchDataSınırı (MQLONG)

Bu alan, bir eşitleme noktası almadan önce bir kanal üzerinden gönderilebilecek veri miktarı sınırını kilobayt cinsinden belirler.

Sınıra ulaşılmamasına neden olan ileti kanal boyunca aktıktan sonra bir eşitleme noktası alınır.

Aşağıdaki koşullardan biri karşılandığında toplu iş sonlandırılır:

- **BatchSize** ileti gönderildi.

- **BatchDataLimit** bayt gönderildi.
- İletim kuyruğu boş ve **BatchInterval** aşıldı.

Değer 0-999999 aralığında olmalıdır. Varsayılan değer 5000 'dir.

Bu öznitelikteki sıfır değeri, bu kanal üzerinden toplu işlere veri sınırı uygulanmadığı anlamına gelir.

Bu parametre yalnızca *ChannelType* /MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSRCVR ya da MQCHT_CLUSSDR içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_11' den küçükse alan yoktur.

BatchHeartbeat (MQLONG)

Bu alan, kanal için toplu sağlıklı işletim bildirimini tetiklemek için kullanılan zaman aralığını belirtir.

Toplu sağlıklı işletim bildirimini, uzak kanal örneğinin belirsiz bir duruma geçmeden önce etkin olup olmadığını gönderen kanalların belirlemesine olanak sağlar. Bir gönderen kanalı, belirtilen zaman aralığı içinde uzak kanal örneğiyle iletişim kurmamışsa toplu sağlıklı işletim bildirimini oluşturur.

Değer, 0-999 999 aralığındadır; birimler milisaniyedir. Sıfır değeri, toplu iş sağlıklı işletim sinyalinin etkinleştirilmediğini gösterir.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_7' den küçükse alan yoktur.

BatchInterval (MQLONG)

Bu alan, geçerli toplu işte *BatchSize* iletilen iletiden daha az ileti varsa, bir kanalın bir toplu işi açık tuttuğu yaklaşık süreyi milisaniye cinsinden belirtir.

BatchInterval sıfırdan büyükse, toplu iş aşağıdaki olaylardan hangisi önce gerçekleşirse o şekilde sonlandırılır:

- *BatchSize* iletileri gönderildi ya da
- Toplu işin başlangıcından bu yana *BatchInterval* milisaniye geçti.

BatchInterval sıfırsa, toplu iş aşağıdaki olaylardan hangisi önce gerçekleşirse o şekilde sonlandırılır:

- *BatchSize* iletileri gönderildi ya da
- iletim kuyruğu boş olur.

BatchInterval , sıfır ile 999 999 999 999 aralığında olmalıdır.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

BatchSize (MQLONG)

Bu alan, kanal eşzamanlanmadan önce bir kanal üzerinden gönderilebilecek ileti sayısı üst sınırını belirler.

Bu alan, *ChannelType* MQCHT_SVRCONN ya da MQCHT_CLNTCONN içeren kanallar için geçerli değildir.

CertificateLabel (MQCHAR64)

Bu alan, kullanılmakta olan sertifika etiketinin ayrıntılarını verir.

IBM MQ , *CertificateLabel* alanı için varsayılan değeri boşluk olarak başlatır.

Bu, yürütme sırasında varsayılan değer olarak yorumlanır ve geriye doğru uyumludur.

Örneğin, 11 'den küçük bir MQCD sürümünün belirtilmesi ya da *CertificateLabel* alanı için varsayılan değer olan boşlukların kullanılması, bu alanın yoksayıldığı anlamına gelir.

Bu alanın uzunluğu MQ_CERT_LABEL_LENGTH tarafından verilir.

ChannelMonitoring (MQLONG)

Bu alan, kanala ilişkin yürürlükteki izleme verileri toplama düzeyini belirler.

Bu alan, MQCHT_CLNTCONN ChannelType içeren kanallar için geçerli değildir.

Aşağıdaki değerlerden biridir:

- MQMON_OFF
- MQMON_LOW
- MQMON_ORTAMI
- MQMON_YükSEKLIK

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_8' den küçükse bu yoktur.

ChannelName (MQCHAR20)

Bu alan, kanal tanımı adını belirler.

İletişim kurabilmek için uzak makinede aynı adı taşıyan bir kanal tanımlaması olmalıdır.

Ad yalnızca şu karakterleri kullanmalıdır:

- Büyük harf A-Z
- Küçük harf a-z
- 0-9 arası sayısal
- Nokta (.)
- Eğik Çizgi (/)
- Alt çizgi (_)
- Yüzde işareti (%)

Ve sağa doğru boşluklarla doldurulur. Baştaki ya da gömülü boşluklara izin verilmez.

Bu alanın uzunluğu MQ_CHANNEL_NAME_LENGTH tarafından verilir.

ChannelStatistics (MQLONG)

Bu alan, kanala ilişkin yürürlükteki istatistik verileri toplama düzeyini belirler.

Bu alan, ChannelType MQCHT_CLNTCONN ya da MQCHT_SVRCONN olan kanallar için geçerli değildir.

Aşağıdaki değerlerden biridir:

- MQMON_OFF
- MQMON_LOW
- MQMON_ORTAMI
- MQMON_YükSEKLIK

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_8' den küçükse bu yoktur.

ChannelType (MQLONG)

Bu alan, kanal tipini belirler.

Aşağıdaki değerlerden biridir:

MQCHT_SENDER

Gönderen.

MQCHT_SERVER

server.

MQCHT_ALICI

Alıcı.

MQCHT_REQUESTER

İsteyen.

MQCHT_CLNTCONN

İstemci bağlantısı.

MQCHT_SVRCONN

Sunucu bağlantısı (istemciler tarafından kullanılmak üzere).

MQCHT_CLUSSDR

Küme gönderen.

MQCHT_CLUSRCVR

Küme alıcısı.

ClientChannelAğırlığı (MQLONG)

Bu alan, hangi istemci-bağlantı kanalı tanımının kullanıldığını etkileyecek bir ağırlıklandırmayı belirtir.

ClientChannelAğırlık özneliği, birden çok uygun tanımlama olduğunda istemci kanal tanımlamalarının ağırlıklandırmalarına dayalı olarak rasgele seçilebilmeleri için kullanılır. Bir istemci, bir kuyruk yöneticisi grubuyla bağlantı isteğinde bulunan bir MQCONN verdiğiğinde, yıldız işaretiyle başlayan bir kuyruk yöneticisi adı belirtilerek ve istemci kanal tanımlama çizelgesinde (CCDT) birden çok uygun kanal tanımlaması varsa, kullanılacak tanımlama, ağırlıklandırmaya dayalı olarak rasgele seçilir ve ilgili ClientChannelAğırlık (0) tanımlamaları alfabetik sırada seçilir.

0-99 aralığında bir değer belirtin. Varsayılan, 0'dır.

0 değeri, yük dengeleme gerçekleştirilmediğini ve uygulanabilir tanımların alfabetik sırada seçildiğini gösterir. Yük dengelemeyi etkinleştirmek için 1 ile 99 arasında bir değer seçin; burada 1 en düşük ağırlıklandırma, 99 en yüksek ağırlıktır. Sıfır olmayan ağırlıklandırmaları olan iki veya daha fazla kanal arasında iletilerin dağılımı, bu ağırlıklandırmaların oranıyla orantılıdır. Örneğin, 2, 4 ve 14 ClientChannelağırlık değerlerine sahip üç kanal yaklaşık %10, %20 ve %70 oranında seçilir. Bu dağıtım garanti edilmez.

Bu öznelik yalnızca istemci-bağlantı kanalı tipi için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Sürüm* MQCD_VERSION_9' dan küçükse bu alan yoktur.

ClusterPtr (MQPTR)

Bu alan, küme adları listesinin adresini belirler.

ClustersDefined sıfırdan büyükse, bu adres küme adları listesinin adresidir. Kanal, listelenen her kümeye ait.

Bu alan yalnızca *ChannelType* MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_5' den küçükse alan yoktur.

ClustersDefined (MQLONG)

Bu alan, kanalın ait olduğu küme sayısını belirtir.

Bu alan, *ClusterPtr* ile gösterilen küme adlarının sayısıdır. Sıfır ya da daha büyük.

Bu alan yalnızca *ChannelType* MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_5' den küçükse alan yoktur.

CLWLChannelPriority (MQLONG)

Bu alan, küme iş yükü kanalı önceliğini belirtir.

İş yükü yöneticisi seçme algoritması, sıralamaya dayalı olarak seçilen hedef kümesinden en yüksek önceliğe sahip hedefi seçer. İki olası hedef kuyruk yöneticisi varsa, bu öznelik bir kuyruk yöneticisinin diğer kuyruk yöneticisine geçmesini sağlamak için kullanılabilir. Tüm iletiler, sona erinceye kadar en yüksek önceliğe sahip kuyruk yöneticisine gider, ardından iletiler bir sonraki en yüksek önceliğe sahip kuyruk yöneticisine gider.

Değer, 0-9 aralığındadır. Varsayılan, 0'dır.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_8' den küçükse alan yoktur.

Daha fazla bilgi için bkz. [Kuyruk yöneticisi kümesinin yapılandırılması](#).

CLWLChannelRank (MQLONG)

Bu alan, küme iş yükü kanal sırasını belirtir.

İş yükü yöneticisi seçme algoritması, en yüksek dereceye sahip bir hedef seçer. Son hedef farklı bir kümedeki bir kuyruk yöneticisine, ara ağ geçidi kuyruk yöneticilerinin sırasını (komşu kümelerin kesişme noktasındaki) ayarlayabilirsiniz; böylece, seçim algoritması son hedefe daha yakın bir hedef kuyruk yöneticisini doğru bir şekilde seçer.

Değer, 0-9 aralığındadır. Varsayılan, 0'dır.

Bu, çıkışa bir giriş alanıdır. *Version*, MQCD_VERSION_8' den küçükse alan yoktur.

Daha fazla bilgi için bkz. [Kuyruk yöneticisi kümesinin yapılandırılması](#).

CLWLChannelWeight (MQLONG)

Bu alan, küme iş yükü kanal ağırlığını belirtir.

Küme iş yükü kanal ağırlığı.

İş yükü yöneticisi seçim algoritması, belirli bir makineye daha fazla ileti gönderilebilmesi için, hedef seçimi sapmak için kanalın "ağırlık" özneliğini kullanır. Örneğin, büyük bir UNIX sunucusundaki bir kanala küçük masaüstü PC 'deki başka bir kanaldan daha büyük bir "ağırlık" verebilirsiniz ve seçim algoritması UNIX sunucusunu kişisel bilgisayardan daha sık seçer.

Değer, 1-99 aralığındadır. Varsayılan değer 50 'dir.

Bu, çıkışa bir giriş alanıdır. *Version*, MQCD_VERSION_8' den küçükse alan yoktur.

Daha fazla bilgi için bkz. [Kuyruk yöneticisi kümesinin yapılandırılması](#).

ConnectionAffinity (MQLONG)

Bu alan, aynı kuyruk yöneticisi adını kullanarak birden çok kez bağlanan istemci uygulamalarının aynı istemci kanalını kullanıp kullanmayacağını belirler.

Birden çok uygulanabilir kanal tanımlaması varsa bu özneliği kullanın.

Değer aşağıdakilerden biridir:

MQCAFTY_PREFERRED

Bir istemci kanal tanımlama çizelgesini (CCDT) okuyan bir süreçteki ilk bağlantı, geçerli CLNTWGHT (0) tanımlamalarına ilişkin ağırlıklandırmaya dayalı olarak geçerli tanımların bir listesini önce alfabetik sırada yaratır. İşlemdeki her bağlantı, listedeki ilk tanımlı kullanarak bağlanmayı dener. Bağlantı başarısız olursa, sonraki tanımlama kullanılır. CLNTWGHT değerleri 0 dışında başarısız olan tanımlamalar listenin sonuna taşınır. CLNTWGHT (0) tanımlamaları listenin başında kalır ve her bağlantı için önce seçilir.

Anasistem adı aynı olan her istemci işlemi her zaman aynı listeyi yaratır.

C, C + + ya da .NET programlama çerçevesi (tam olarak yönetilen .NETdahil) ile yazılan istemci uygulamaları için liste, liste yaratıldıktan sonra CCDT değiştirildiyse güncellenir.

Bu değer varsayılan değerdir.

MQCAFTY_NONE

CCDT okuyan bir işlemdeki ilk bağlantı, uygulanabilir tanımlamaların bir listesini yaratır. Bir süreçteki tüm bağlantılar, alfabetik sırayla seçilen herhangi bir uygulanabilir CLNTWGHT (0) tanımlaması ile ağırlıklandırmaya dayalı olarak uygulanabilir bir tanım seçer.

C, C + + ya da .NET programlama çerçevesi (tam olarak yönetilen .NETdahil) ile yazılan istemci uygulamaları için liste, liste yaratıldıktan sonra CCDT değiştirildiyse güncellenir.

Bu öznelik yalnızca istemci-bağlantı kanalı tipi için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Sürüm* MQCD_VERSION_9' dan küçükse bu alan yoktur.

ConnectionName (MQCHAR264)

Bu alan, kanalın bağlantı adını belirler.

Küme-alıcı kanalları için (belirtildiğinde) CONNAME, yerel kuyruk yöneticisiyle ve diğer kanallar için hedef kuyruk yöneticisiyle ilgilidir. Belirlediğiniz değer, kullanılacak iletim protokolüne (*TransportType*) bağlıdır:

- MQXPT_LU62 için bu, ortak Mantıksal Birim 'in tam olarak nitelenmiş adıdır.
- MQXPT_NETBIOS için, uzak makinede tanımlanan NetBIOS adıdır.
- MQXPT_TCP için, anasistem adı, IPv4 noktalı onlu ya da IPv6 onaltılı biçiminde belirtilen uzak makinenin ağ adresi ya da küme alıcı kanalları için yerel makinedir.
- MQXPT_SPX için, 4 baytlık bir ağ adresi, 6 baytlık bir düğüm adresi ve 2 baytlık bir yuva numarasından oluşan SPX stili bir adrestir.

Bir kanal tanımlanırken, bu alan *ChannelType* MQCHT_SVRCONN ya da MQCHT_ALICI içeren kanallarla ilgili değildir. Ancak, kanal tanımı bir çıkışa geçirildiğinde, bu alan, kanal tipi ne olursa olsun iş ortağının adresini içerir.

Bu alanın uzunluğu MQ_CONN_NAME_LENGTH tarafından verilir. *Version* , MQCD_VERSION_2' den küçükse bu alan yoktur.

DataConversion (MQLONG)

Bu alan, alan ileti kanalı aracısı bu dönüştürmeyi gerçekleştiremezse, gönderen ileti kanalı aracısının uygulama ileti verilerini dönüştürmeye çalışıp çalışmadığını belirler.

Bu alan yalnızca mantıksal ileti kesimi olmayan iletiler için geçerlidir; MCA hiçbir zaman kesim olan iletileri dönüştürmeyi denemez.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir. Aşağıdakilerden biridir:

MQCDC_SENDER_CONVERSION

Gönderen tarafından dönüştürme.

MQCDC_NO_SENDER_CONVERSION

Gönderen tarafından dönüştürme yok.

DefReconnect (MQLONG)

DefReconnect kanal özniteliği, bir istemci bağlantı kanalına ilişkin varsayılan yeniden bağlantı özniteliği değerini ayarlar.

Varsayılan otomatik istemci yeniden bağlanma seçeneği. Bir istemci uygulamasını otomatik olarak yeniden bağlamak için IBM MQ MQI client yapılandırabilirsiniz. IBM MQ MQI client , bir bağlantı hatasından sonra bir kuyruk yöneticisine yeniden bağlanmayı dener. Uygulama istemcisi bir MQCONN ya da MQCONNX MQI çağrısı yayınlamadan yeniden bağlanmayı dener.

Yeniden bağlanma bir MQCONNX seçeneğidir. DefReconnect kanal özniteliğini kullanarak, MQCONNkullanan var olan uygulamalara yeniden bağlantı davranışı ekleyebilirsiniz. MQCONNXkullanan uygulamaların yeniden bağlanma davranışını da değiştirebilirsiniz.

Yeniden bağlanma davranışını ayarlamak ya da değiştirmek için mqclient . ini dosyasından DefRecon değerini de ayarlayabilirsiniz. mqclient . ini dosyasındaki DefRecon değeri, DefReconnect kanal özniteliğinden önceliklidir.

Syntax

DefReconnect (MQRCN_NO (default) |MQRCN_YES|MQRCN_Q_MGR|MQRCN_DISABLED)

Parametreler

MQRCN_NO

MQRCN_NO varsayılan değerdir.

MQCONNX tarafından geçersiz kılınmadıkça, istemci otomatik olarak yeniden bağlanmaz.

MQRCN_YES

MQCONNX tarafından geçersiz kılınmadıkça, istemci otomatik olarak yeniden bağlanır.

MQRCN_Q_MGR

MQCONNX tarafından geçersiz kılınmadıkça, istemci otomatik olarak yeniden bağlanır, ancak yalnızca aynı kuyruk yöneticisine bağlanır. QMGR seçeneği, MQCNO_RECONNECT_Q_MGR ile aynı etkiye sahiptir.

MQRCN_DISABLED

MQCONNX MQI çağrısı kullanılarak istemci programı tarafından istense de, yeniden bağlanma geçersiz kılınır.

IBM MQ classes for Java otomatik istemci yeniden bağlantısını desteklemez.

Çizelge 822. Otomatik yeniden bağlanma, uygulamada ve kanal tanımında ayarlanan değerlere bağlıdır

DefReconnect	Uygulamada ayarlanan yeniden bağlanma seçenekleri			
	MQCNO_RECONNE CT	MQCNO_RECONNE CT_Q_MGR	MQCNO_RECONNE CT_AS_DEF	MQCNO_RECONNE CT_DISABLED
MQRCN_NO	EVET	QMGR	HAYIR	HAYIR
MQRCN_YES	EVET	QMGR	EVET	HAYIR
MQRCN_Q_MGR	EVET	QMGR	QMGR	HAYIR
MQRCN_DISABLED	HAYIR	HAYIR	HAYIR	HAYIR

İlgili kavramlar

[Otomatik istemci yeniden bağlantısı](#)

[Kanal ve istemci yeniden bağlantısı](#)

[İstemci yapılandırma dosyasının KANAL kısmı](#)

İlgili başvurular

[“MQCNO için seçenekler \(MQLONG\)” sayfa 321](#)

MQCONNX işlemini denetleyen seçenekler.

Tanım (MQCHAR64)

Bu alan, açıklayıcı açıklama için kullanılabilir.

Alanın içeriği, İleti Kanalı Aracıları için önemli değildir. Ancak, yalnızca görüntülenebilecek karakterleri içermelidir. Boş karakter içeremez; gerekirse, boşluk ile sağa doğru doldurulur. Bir DBCS kuruluşunda, alan DBCS karakterleri içerebilir (alan uzunluğu üst sınırı 64 bayttir).

Not: Bu alan, kuyruk yöneticisinin karakter kümesinde (**CodedCharSetId** kuyruk yöneticisi özniteliğinde tanımlandığı şekilde) olmayan karakterler içeriyorsa, bu alan başka bir kuyruk yöneticisine gönderilirse, bu karakterler yanlış çevrilebilir.

Bu alanın uzunluğu MQ_CHANNEL_DESC_LENGTH tarafından verilir.

DiscInterval (DiscInterval) (MQLONG)

Bu alan, kanal sonlandırılmadan önce, kanalın iletim kuyruğuna ulaşması için bekleyeceği sürenin üst sınırını saniye cinsinden belirler.

Başka bir deyişle, bağlantı kesme aralığını belirtir.

Sıfır değeri, MCA ' nın süresiz olarak beklemesine neden olur.

TCP protokolünü kullanan sunucu bağlantısı kanalları için aralık, saniye cinsinden belirtilen istemci etkinlik dışı bağlantı kesme değerini gösterir. Bir sunucu bağlantısı bu süre boyunca ortak istemcisinden herhangi bir iletişim almazsa, bağlantıyı sonlandırır. Sunucu bağlantısı boşta durma aralığı yalnızca bir istemciden gelen IBM MQ API çağrıları arasında geçerlidir; bu nedenle, uzun süreli MQGET ve bekleme çağrısı sırasında hiçbir istemci bağlantısı kesilmez.

Bu öznitelik, TCP dışındaki protokolleri kullanan sunucu bağlantısı kanalları için geçerli değildir.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSSDR, MQCHT_CLUSRCVR ya da MQCHT_SVRCONN içeren kanallar için geçerlidir.

ExitDataUzunluğu (MQLONG)

Bu alan, *MsgUserDataPtr*, *SendUserDataPtr* ve *ReceiveUserDataPtr* alanları tarafından adreslenen çıkış kullanıcısı veri öğeleri listelerindeki her bir kullanıcı veri öğesinin bayt cinsinden uzunluğunu belirtir.

Bu uzunluğun MQ_EXIT_DATA_LENGTH ile aynı olması gerekmez.

Bu, çıkışa bir giriş alanıdır. *Version*, MQCD_VERSION_4' den küçükse alan yoktur.

ExitNameUzunluğu (MQLONG)

Bu alan, *MsgExitPtr*, *SendExitPtr* ve *ReceiveExitPtr* alanları tarafından adreslenen çıkış adları listelerindeki her bir adın bayt cinsinden uzunluğunu belirtir.

Bu uzunluğun MQ_EXIT_NAME_LENGTH ile aynı olması gerekmez.

Bu, çıkışa bir giriş alanıdır. *Version*, MQCD_VERSION_4' den küçükse alan yoktur.

HdrCompListesi [2] (MQLONG)

Bu alan, kanal tarafından desteklenen üstbilgi veri sıkıştırma tekniklerinin listesini belirtir.

Liste aşağıdaki değerlerden birini ya da birkaçını içerir:

MQCOMPRESS_NONE

Üstbilgi veri sıkıştırması gerçekleştirilmez.

MQCOMPRESS_SYSTEM

Üstbilgi veri sıkıştırması gerçekleştirilir.

MQCOMPRESS_NOT_VAR

Listede kullanılmayan değerler bu değere ayarlanır.

Bu, çıkışa bir giriş alanıdır. *Version*, MQCD_VERSION_8' den küçükse alan yoktur.

HeartbeatInterval (MQLONG)

Bu alan, sağlıklı işletim bildirim akışları arasındaki süreyi saniye cinsinden belirtir.

Bu alanın yorumu, kanal tipine bağlıdır:

- MQCHT_SENDER, MQCHT_SERVER, MQCHT_ALICI MQCHT_REQUESTER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR kanal tipi için bu alan, iletim kuyruğunda ileti olmadığına gönderen MCA ' dan geçirilen sağlıklı işletim bildirim akışları arasındaki saniye cinsinden süredir. Bu, alan MCA ' ya kanalı susturma fırsatı verir. Yararlı olması için *HeartbeatInterval*, *DiscInterval* değerinden küçük olmalıdır.
- MQCD Paylaşım Sohbetleri alanı sıfır olarak ayarlanmış bir MQCHT_CLNTCONN ya da MQCHT_SVRCONN kanal tipi için, bu alan, MCA bir istemci uygulaması adına MQGMO_WAIT seçeneğiyle bir MQGET çağrısı yayınladığında sunucu MCA ' dan geçirilen sağlıklı işletim bildirim akışları arasındaki saniye cinsinden süreyi gösterir. Bu, sunucu MCA ' nın MQGMO_WAIT ile bir MQGET sırasında istemci bağlantısının başarısız olduğu durumları işlemlerini sağlar.
- MQCD Paylaşım Etkileşimi alanı sıfır dışında bir değere ayarlanmış MQCHT_CLNTCONN ya da MQCHT_SVRCONN kanal tipi için bu alan, gönderilen ya da alınan veri akışı olmadığına sağlıklı işletim bildirim akışı arasındaki saniye cinsinden süreyi gösterir. Bu, kanalın verimli bir şekilde susturulmasını sağlar.

Değer, 0-999 aralığındadır. Kullanılan değer, her iki tarafta da 0 değeri belirtilmedikçe, gönderen tarafında ve alıcı tarafında belirtilen değerlerden daha büyüktür; bu durumda sağlıklı işletim bildirim değış tokuşu gerçekleşmez.

Bu, çıkışa bir giriş alanıdır. *Version*, MQCD_VERSION_4' den küçükse alan yoktur.

KeepAliveAralığı (MQLONG)

Bu alan, kanala ilişkin canlı tutma zamanlaması için iletişim yığınına geçirilen değeri belirler.

Değer, TCP/IP ve SPX iletişim protokolleri için geçerlidir, ancak tüm uygulamalar bu parametreyi desteklemez.

Değer, 0-99 999 aralığındadır; birimler saniyedir. Sıfır değeri, kanal canlı tutma özelliğinin etkinleştirilmediğini gösterir; ancak, TCP/IP canlı tutma (kanal canlı tutma değil) etkinleştirildiyse, canlı tutma işlemi yine de gerçekleşebilir. Aşağıdaki özel değer de geçerlidir:

MQKAI_AUTO

Otomatik.

Bu değer, canlı tutma aralığının kararlaştırılan sağlıklı işletim bildirim aralığından hesaplandığını gösterir:

- Kararlaştırılan sağlıklı işletim bildirim aralığı sıfırdan büyükse, kullanılan canlı tutma aralığı sağlıklı işletim bildirim aralığı artı 60 saniyedir.
- Kararlaştırılan sağlıklı işletim bildirim aralığı sıfırsa, kullanılan canlı tutma aralığı sıfırdır.
- z/OS' da, kuyruk yöneticisi nesnesinde TCPKEEP (YES) belirtildiğinde TCP/IP canlı tutma durumu oluşur.
- Diğer ortamlarda, dağıtılmış kuyruğa alma yapılarındaki TCP kısmı için **KEEPALIVE=YES** değıştirgesi belirtildiğinde TCP/IP canlı tutma (Keepalive) oluşur.

Bu alan yalnızca *TransportType* MQXPT_TCP ya da MQXPT_SPX içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_7' den küçükse alan yoktur.

LocalAddress (MQCHAR48)

Bu alan, giden iletişim için kanal için tanımlanan yerel TCP/IP adresini belirler.

Giden iletişim için belirli bir adres tanımlanmadıysa bu alan boştur. Adres, isteğe bağlı olarak bir kapı numarası ya da kapı numarası aralığı içerebilir. Bu adresin biçimi şöyledir:

```
[ip-addr] [(low-port[, high-port])]
```

Burada köşeli ayraç ([]) isteğe bağlı bilgileri gösterir; ip-addr , IPv4 noktalı onlu, IPv6 onaltılı ya da alfasayısal biçimde ve low-port ve high-port , parantez içine alınmış kapı numaralarıdır. Hepsisi isteğe bağlıdır.

Giden iletişim için belirli bir IP adresi, kapı ya da kapı aralığı, bir kanalın farklı bir TCP/IP yığnında yeniden başlatıldığı kurtarma senaryolarında kullanışlıdır.

LocalAddress , formda *ConnectionName* ile benzer, ancak bununla karıştırılmamalıdır.

LocalAddress , yerel iletişimin özelliklerini belirtirken, *ConnectionName* uzak kuyruk yöneticisine nasıl erişileceğini belirtir.

IBM MQ 9.3.0olanağından, Java İleti Kuyruğa Alma Arabirimi (JMQUI), bir kanal yönetim ortamı yaratıldıktan ve bir kuyruk yöneticisine bağlandıktan sonra bir MQCD nesnesinde yerel adres alanının ayarlandığından emin olmak için güncellenmiştir. Bu, Java içinde yazılan bir kanal çıkışı yöntemi MQCD.getLocalAddress() çağırıldığında, yöntemin kanal örneğinin kullandığı yerel adresi döndürdüğü anlamına gelir. IBM MQ 9.3.0öncesinde, kanal güvenlik çıkışı kanal örneği tarafından kullanılan yerel adrese erişemedi ve MQCD.getLocalAddress() yöntemi boş değer döndürdü.

Bu alan yalnızca *TransportType* MQXPT_TCP içeren kanallar ve MQCHT_SENDER, MQCHT_SERVER, MQCHT_REQUESTER, MQCHT_CLNTCONN, MQCHT_CLUSR ya da MQCHT_CLUSRCVR içeren *ChannelType* kanalları için geçerlidir.

Bu alanın uzunluğu MQ_LOCAL_ADDRESS_LENGTH tarafından verilir. *Version* , MQCD_VERSION_7' den küçükse bu alan yoktur.

LongMCAUserIdLength (MQLONG)

Bu alan, *LongMCAUserIdPtr* ile gösterilen tam MCA kullanıcı kimliğinin bayt cinsinden uzunluğunu belirtir.

Bu alan, *ChannelType* MQCHT_CLNTCONN içeren kanallar için uygun değildir.

Bu, çıkışa bir giriş/çıkış alanıdır. *Version* , MQCD_VERSION_6' dan küçükse alan yoktur.

LongMCAUserIdPtr (MQPTR)

Bu alan, uzun MCA kullanıcı kimliğinin adresini belirtir.

LongMCAUserIdLength sıfırdan büyükse, bu alan tam MCA kullanıcı kimliğinin adresidir. Tam tanıtıcının uzunluğu *LongMCAUserIdLength* tarafından verilir. MCA kullanıcı kimliğinin ilk 12 baytı da *MCAUserIdentifier* alanında bulunur.

MCA kullanıcı kimliğinin ayrıntıları için *MCAUserIdentifier* alanının açıklamasına bakın.

Bu alan, *ChannelType* MQCHT_SDR, MQCHT_SVR, MQCHT_CLNTCONN ya da MQCHT_CLUSSDR içeren kanallar için anlamlı değildir.

Bu, çıkışa bir giriş/çıkış alanıdır. *Version* , MQCD_VERSION_6' dan küçükse alan yoktur.

LongRemoteUserIdLength (MQLONG)

Bu alan, *LongRemoteUserIdPtr* ile gösterilen tam uzak kullanıcı kimliğinin bayt cinsinden uzunluğunu belirtir.

Bu alan yalnızca *ChannelType* MQCHT_CLNTCONN ya da MQCHT_SVRCONN içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_6' dan küçükse alan yoktur.

LongRemoteUserIdPtr (MQPTR)

Bu alan, uzun uzak kullanıcı kimliğinin adresini belirler.

LongRemoteUserIdLength sıfırdan büyükse, bu işaret tam uzak kullanıcı kimliğinin adresidir. Tam tanıtıcının uzunluğu *LongRemoteUserIdLength* tarafından verilir. Uzak kullanıcı kimliğinin ilk 12 baytı da *RemoteUserIdentifier* alanında bulunur.

Uzak kullanıcı kimliğinin ayrıntıları için *RemoteUserIdentifier* alanının açıklamasına bakın.

Bu alan yalnızca *ChannelType* MQCHT_CLNTCONN ya da MQCHT_SVRCONN içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_6' dan küçükse alan yoktur.

LongRetrySayısı (MQLONG)

Bu alan, *ShortRetryCount* tarafından belirtilen sayı tükendikten sonra kullanılan sayıyı belirtir.

Bir hatayı işletmene kaydetmeden önce, *LongRetryInterval* tarafından belirtilen aralıklarla uzak makineye bağlanmak için yapılacak deneme sayısı üst sınırını belirler.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

LongRetryAralığı (MQLONG)

Bu alan, uzak makineyle bağlantıyı yeniden denemeden önce beklenecek saniye sayısı üst sınırını belirler.

Kanal etkin duruma gelmek için beklemek zorunda kaldıysa, yeniden denemeler arasındaki aralık uzatılabilir.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

MaxInstances (MQLONG)

Bu alan, başlatılabilecek tek bir sunucu bağlantısı kanalının koşut zamanlı eşgörünüm sayısı üst sınırını belirler.

Bu alan yalnızca sunucu bağlantısı kanallarında kullanılır.

Alan, 0-999 999 999 aralığında bir değere sahip olabilir. Sıfır değeri, tüm istemci erişimini engeller.

Bu alanın varsayılan değeri 999 999 999 'dur.

Bu alanın değeri, çalışmakta olan sunucu bağlantısı kanalının eşgörünümlerinin sayısından küçük bir sayıya düşürülürse, çalışmakta olan eşgörünümler bundan etkilenmez. Ancak, çalışmakta olan eşgörünümlerinin sayısının alanın değerinden az olması için, varolan yönetim ortamlarının yeterli sayıda çalışması duruncaya kadar yeni yönetim ortamları başlatılamaz.

MaxInstancesPerClient (MQLONG)

Bu alan, tek bir istemciden başlatılabilecek tek bir sunucu bağlantısı kanalının koştuzamanlı yönetim ortamı sayısı üst sınırını belirler.

Bu bağlamda, aynı uzak ağ adresinden gelen bağlantılar aynı istemciden gelen bağlantılar olarak kabul edilir.

Bu alan yalnızca sunucu bağlantısı kanallarında kullanılır.

Alan, 0-999 999 999 aralığında bir değere sahip olabilir. Sıfır değeri, tüm istemci erişimini engeller.

Bu alanın varsayılan değeri 999 999 999 'dur.

Bu alanın değeri, tek tek istemcilerden çalışmakta olan sunucu bağlantısı kanalının eşgörünümlerinin sayısından az bir sayıya indirgenirse, çalışmakta olan eşgörünümler bundan etkilenmez. Ancak, bu istemcilerden herhangi birinin yeni eşgörünümleri, çalışmakta olan eşgörünümlerinin sayısı (istemcinin yeni bir eşgörünüm başlatmayı denemesinden kaynaklanmakta olan) alanın değerinden az olacak şekilde yeterli sayıda var olan yönetim ortamı duruncaya kadar başlatılamaz.

MaxMsgUzunluğu (MQLONG)

Bu alan, kanalda iletilebilecek ileti uzunluğu üst sınırını belirler.

Bu, uzak kanal değeri ile karşılaştırılır ve gerçek üst sınır, iki değerinki alt sınırındır.

MCAName (MQCHAR20)

Bu alan ayrılmış bir alandır.

Bu alanın değeri boş.

Bu alanın uzunluğu MQ_MCA_NAME_LENGTH tarafından verilir.

MCASecurityId (MQBYTE40)

Bu alan, MCA için güvenlik tanıtıcısını belirtir.

Bu alan, *ChannelType* MQCHT_CLNTCONN içeren kanallar için uygun değildir.

Aşağıdaki özel değer, güvenlik tanıtıcısı olmadığını gösterir:

MQSID_YOK

Güvenlik tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQSID_NONE_ARRAY değişmezi de tanımlanır; bu değişmez MQSID_NONE ile aynı değere sahiptir, ancak dizgi yerine bir karakter dizisidir.

Bu, çıkışa bir giriş/çıkış alanıdır. Bu alanın uzunluğu MQ_SECURITY_ID_LENGTH tarafından verilir. *Version* , MQCD_VERSION_6' dan küçükse bu alan yoktur.

MCAType (MQLONG)

Bu alan, ileti kanalı aracısı programının tipini belirler.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_REQUESTER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Değer aşağıdakilerden biridir:

MQMCAST_PROCESS

Süreç.

İleti kanalı aracısı ayrı bir işlem olarak çalışır.

MQMCAT_THREAD

İş parçacığı (Çoklu platformlar).

İleti kanalı aracısı ayrı bir iş parçacığı olarak çalışır.

Sürüm MQCD_VERSION_2' den küçükse bu alan yoktur.

MCAUserIdentifier (MQCHAR12)

Bu alan, ileti kanalı aracısı (MCA) için kullanıcı kimliğini belirler.

Bu alan, MCA kullanıcı kimliğinin ilk 12 baytını kullanır ve bir güvenlik aracısı tarafından ayarlanabilir.

MCA kullanıcı kimliğini içeren iki alan vardır:

- *MCAUserIdentifier*, MCA kullanıcı kimliğinin ilk 12 baytını içerir ve tanıtıcı 12 bayttan kısaysa boşluklarla doldurulur. *MCAUserIdentifier* boş olabilir.
- *LongMCAUserIdPtr*, 12 bayttan uzun olabilen tam MCA kullanıcı kimliğini gösterir. Uzunluğu *LongMCAUserIdLength* tarafından verilir. Tam tanıtıcı sonda boşluk içermiyor ve boş sonlandırılmış değil. Tanıtıcı boşsa, *LongMCAUserIdLength* sıfırdır ve *LongMCAUserIdPtr* değeri tanımlıdır.

Not: *Version*, MQCD_VERSION_6' dan küçükse *LongMCAUserIdPtr* yoktur.

MCA kullanıcı kimliği boş değilse, ileti kanalı aracısı tarafından IBM MQ kaynaklarına erişim yetkisi için kullanılacak kullanıcı kimliğini belirtir. MQCHT_REQUESTER, MQCHT_ALICI ve MQCHT_CLUSRCVR kanal tipleri için, PutAuthority MQPA_DEFAULT ise, hedef kuyruklara koyma işlemine ilişkin yetkilendirme denetimleri için kullanılan kullanıcı kimliği budur.

MCA kullanıcı kimliği boşsa, ileti kanalı aracısı varsayılan kullanıcı kimliğini kullanır.

MCA kullanıcı kimliği, ileti kanalı aracısının kullanması gereken kullanıcı kimliğini belirtmek için bir güvenlik çıkışı tarafından ayarlanabilir. Çıkış, *MCAUserIdentifier* değerini ya da *LongMCAUserIdPtr* ile gösterilen dizgiyi değiştirebilir. Her ikisi de değişirse, ancak birbirlerinden farklıysa, MCA, *MCAUserIdentifier* tercinde *LongMCAUserIdPtr* değerini kullanır. Çıkış, *LongMCAUserIdPtr* ile adreslenen dizginin uzunluğunu değiştirirse, *LongMCAUserIdLength* uygun şekilde ayarlanmalıdır. Çıkış, tanıtıcının uzunluğunu artırırsa, çıkışın gereken uzunlukta bir saklama alanı ayırması, bu saklama alanını gerekli tanıtıcıya ayarlaması ve bu saklama alanının adresini *LongMCAUserIdPtr* içine yerleştirmesi gerekir. Çıkış, daha sonra MQXR_TERM nedeniyle çağrıldığında o depolamanın serbest bırakılmasından sorumludur.

ChannelType MQCHT_SVRCONN içeren kanallar için, kanal tanımlamasında *MCAUserIdentifier* boşsa, istemciden aktarılan kullanıcı kimliği bu kanala kopyalanır. Bu kullanıcı kimliği (sunucudaki güvenlik çıkışında yapılan herhangi bir değişiklikten sonra), istemci uygulamasının altında çalıştığı varsayılır.

MCA kullanıcı kimliği, *ChannelType* MQCHT_SDR, MQCHT_SVR, MQCHT_CLNTCONN, MQCHT_CLUSSDR içeren kanallar için uygun değil.

Bu, çıkışa bir giriş/çıkış alanıdır. Bu alanın uzunluğu MQ_USER_ID_LENGTH tarafından verilir. *Version*, MQCD_VERSION_2' den küçükse bu alan yoktur.

ModeName (MQCHAR8)

Bu alan LU 6.2 kip adını belirtir.

Bu alan yalnızca iletim protokolü (*TransportType*) MQXPT_LU62 ise ve *ChannelType* MQCHT_SVRCONN ya da MQCHT_ALICI değilse geçerlidir.

Bu alan her zaman boştur. Bilgiler iletişim tarafı nesnesinde bulunur.

Bu alanın uzunluğu MQ_MODE_NAME_LENGTH tarafından verilir.

MsgCompListesi [16] (MQLONG)

Bu alan, kanal tarafından desteklenen ileti verileri sıkıştırma tekniklerinin listesini belirtir.

Liste aşağıdaki değerlerden birini ya da birkaçını içerir:

MQCOMPRESS_NONE

İleti veri sıkıştırması gerçekleştirilmez.

MQCOMPRESS_RLE

İleti veri sıkıştırması, çalıştırma uzunluğu kodlaması kullanılarak gerçekleştirilir.

MQCOMPRESS_ZLIBFAST

İleti veri sıkıştırması, zlib sıkıştırma tekniği kullanılarak gerçekleştirilir. Hızlı sıkıştırma süresi tercih edilir.

MQCOMPRESS_ZLIBHIGH

İleti veri sıkıştırması, zlib sıkıştırma tekniği kullanılarak gerçekleştirilir. Yüksek düzeyde sıkıştırma tercih edilir.

MQCOMPRESS_ANY

Kuyruk yöneticisi tarafından desteklenen herhangi bir sıkıştırma tekniği, ileti sıkıştırması için kullanılabilir. MQCOMPRESS_ANY yalnızca alıcı, istekte bulunan ve sunucu bağlantısı kanalları için geçerlidir.

MQCOMPRESS_NOT_VAR

Listede kullanılmayan değerler bu değere ayarlanır.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_8' den küçükse alan yoktur.

MsgExit (MQCHARn)

Bu alan, kanal iletisi çıkış adını belirler.

Bu ad boş değilse, aşağıdaki zamanlarda çıkış çağrılır:

- İletim kuyruğundan (gönderen ya da sunucu) bir ileti alındıktan hemen sonra ya da bir ileti hedef kuyruğa (alıcı ya da istekte bulunan) konmadan hemen önce.

Çıkışa, tüm uygulama iletisinin ve iletim kuyruğu üstbilgisinin değiştirilmesi verilir.

- Kanal başlatılırken ve sonlandırılırken.

Bu alan, *ChannelType* MQCHT_SVRCONN ya da MQCHT_CLNTCONN içeren kanallarla ilgili değildir; bu tür kanallar için hiçbir zaman ileti çıkışı çağrılmaz.

Bu alanın çeşitli ortamlardaki içeriğinin açıklaması için bkz. [“MQCD-Kanal tanımlaması” sayfa 1454](#) .

Bu alanın uzunluğu MQ_EXIT_NAME_LENGTH tarafından verilir.

Not: Bu değişmez değeri ortama özgüdür.

MsgExitPtr (MQPTR)

Bu alan, ilk *MsgExit* alanının adresini belirtir.

MsgExitsDefined sıfırdan büyükse, bu adres zincirdeki her kanal ileti çıkışının ad listesinin adresidir.

Her ad, *ExitNameLength* uzunluğundaki bir alandır ve sağa boşluklarla doldurulur. Her çıkış için bir tane olmak üzere birbirine bitişik *MsgExitsDefined* alanları vardır.

Bu adlarda çıkışla yapılan değişiklikler korunur; ancak, ileti kanalı çıkışı belirttik bir işlem gerçekleştirmez; hangi çıkışların çağrılacağını değiştirmez.

MsgExitsDefined sıfırsa, bu alan boş gösterge olur.

Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

MsgExitsTanımlı (MQLONG)

Bu alan, zincirde tanımlanan kanal iletisi çıkışlarının sayısını belirtir.

Sıfırdan büyük ya da sıfıra eşittir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

MsgRetrySayısı (MQLONG)

Bu alan, ilk girişim başarısız olduktan sonra MCA ' nın iletiyi kaç kez yerleştirmeyi denediğini belirtir.

Bu alan, ilk MQOPEN ya da MQPUT, MQCC_FAILED tamamlanma koduyla başarısız olursa, MCA ' nın açma ya da koyma işlemini kaç kez deneyeceğini gösterir. Bu özneliğin etkisi, *MsgRetryExit* ' un boş mu, yoksa boş mu olduğuna bağlıdır:

- *MsgRetryExit* boşsa, **MsgRetryCount** özneliği MCA ' nın yeniden denenip denenmeyeceğini denetler. Öznelik değeri sıfır, yeniden deneme girişiminde bulunulmaz. Öznelik değeri sıfırdan büyükse, yeniden denemeler **MsgRetryInterval** özneliği tarafından belirtilen aralıklarla denir.

Yeniden denemeler yalnızca aşağıdaki neden kodları için denir:

- MQRC_PAGESET_FULL
- MQRC_PUT_INENGELLEME
- MQRC_Q_FULL

Diğer neden kodları için, MCA, başarısız olan iletiyi yeniden denemeden hemen olağan hata işlemesine devam eder.

- *MsgRetryExit* boş değilse, **MsgRetryCount** özneliği MCA ' yı etkilemez; bunun yerine, yeniden deneme işleminin kaç kez deneneceğini ve hangi aralıklarla deneneceğini belirleyen ileti yeniden deneme çıkışıdır; **MsgRetryCount** özneliği sıfır olsa bile çıkış çağrılır.

MsgRetryCount özneliği MQCD yapısındaki çıkışın kullanımına sunulur, ancak bunu yerine getirmesi gerekli değildir-çıkış MQCXP ' nin *ExitResponse* alanında MQXCC_SUPPRESS_FUNCTION ögesini döndürünceye kadar süresiz olarak devam eder.

Bu alan yalnızca *ChannelType* MQCHT_REQUESTER, MQCHT_CLUSRCVR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Version , MQCD_VERSION_3' den küçükse bu alan yoktur.

MsgRetryÇıkış (MQCHARn)

Bu alan, kanal iletisi yeniden deneme çıkış adını belirtir.

İleti yeniden deneme çıkışı, MCA bir MQOPEN ya da MQPUT çağrısından MQCC_FAILED ' in tamamlanma kodunu aldığı anda MCA tarafından çağrılan bir çıkıştır. Çıkışın amacı, MCA ' nın MQOPEN ya da MQPUT işlemini yeniden denemeden önce bekleyeceği bir zaman aralığı belirtmektir. Diğer bir seçenek olarak, çıkış işlemi yeniden denemeyecek şekilde ayarlanabilir.

Çıkış, tamamlama kodu MQCC_FAILED olan tüm neden kodları için çağrılır-çıkışın ayarları, MCA ' nın hangi neden kodlarını yeniden denemesini istediğini, kaç girişimde bulunacağını ve zaman aralıklarını belirler.

İşlem artık denenmediğinde, MCA olağan hata işlemesini gerçekleştirir; bu işleme, bir kural dışı durum raporu iletisi oluşturulmasını (gönderen tarafından belirtildiyse) ve özgün iletinin gitmeyen iletiler kuyruğuna yerleştirilmesini ya da iletinin atılmasına (gönderen tarafından belirtilen MQRO_DEAD_LETTER_Q ya da MQRO_DISCARD_MSG ' nin belirlenip belirlenmediğine göre) dahildir. Gitmeyen iletiler kuyruğunu içeren hatalar (örneğin, gitmeyen iletiler kuyruğu dolu), ileti yeniden deneme çıkışının çağrılmasına neden olmaz.

Çıkış adı boş değilse, aşağıdaki zamanlarda çıkış çağrılır:

- Bir iletiyi yeniden teslim etmeye çalışmadan önce bekleme gerçekleştirilmeden hemen önce
- Kanal başlatılırken ve sonlandırılırken

Bu alanın çeşitli ortamlardaki içeriğinin açıklaması için bkz. [“MQCD-Kanal tanımlaması” sayfa 1454](#) .

Bu alan yalnızca *ChannelType* MQCHT_REQUESTER, MQCHT_CLUSRCVR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Bu alanın uzunluğu MQ_EXIT_NAME_LENGTH tarafından verilir.

Not: Bu değişimin değeri ortama özgüdür.

Version , MQCD_VERSION_3' den küçükse bu alan yoktur.

MsgRetryAralığı (MQLONG)

Bu alan, açma ya da koyma işleminin yeniden denenmesinden sonraki milisaniye cinsinden aralık alt sınırını belirtir.

Bu özneliğin etkisi, *MsgRetryExit* ' un boş mu, yoksa boş mu olduğuna bağlıdır:

- *MsgRetryExit* boşsa, ilk MQOPEN ya da MQPUT MQCC_FAILED tamamlanma koduyla başarısız olursa, **MsgRetryInterval** özneliği MCA ' nın bir iletiyi yeniden denemeden önce bekleyeceği süre alt sınırını belirtir. Sıfır değeri, önceki denemeden sonra yeniden denemenin mümkün olan en kısa sürede gerçekleştirileceği anlamına gelir. Yeniden denemeler yalnızca *MsgRetryCount* sıfırdan büyükse gerçekleştirilir.

Bu öznelik, ileti yeniden deneme çıkışı MQCXP ' deki *MsgRetryInterval* alanında geçersiz bir değer döndürürse bekleme süresi olarak da kullanılır.

- *MsgRetryExit* boş değilse, **MsgRetryInterval** özneliği MCA 'yı etkilemez; bunun yerine MCA 'nın ne kadar bekleyeceğini belirleyen ileti yeniden deneme çıkışıdır. **MsgRetryInterval** özneliği MQCD yapısındaki çıkışın kullanımına sunulur, ancak bunu yerine getirmesi için çıkış gerekli değildir.

Değer, 0-999 999 999 999 aralığındadır.

Bu alan yalnızca *ChannelType* MQCHT_REQUESTER, MQCHT_CLUSRCVR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Version , MQCD_VERSION_3' den küçükse bu alan yoktur.

Version MQCD_VERSION_4' den küçükse, bu yapıdaki aşağıdaki alanlar yoktur.

MsgRetryUserData (MQCHAR32)

Bu alan, kanal iletişi yeniden deneme çıkışı kullanıcı verilerini belirtir.

Bu veriler, **ChannelExitParms** değiştirgesinin *ExitData* alanındaki kanal iletişine geçirilir (bkz. MQ_CHANNEL_EXIT).

Bu alan başlangıçta kanal tanımında ayarlanan verileri içerir. Ancak, bu MCA örneğinin yaşam süresi boyunca, bu alanın içeriğinde herhangi bir tip çıkışla yapılan değişiklikler MCA tarafından korunur ve bu MCA örneğine ilişkin çıkışların sonraki çağrılmasında (türüne bakılmaksızın) görünür hale getirilir. Bu tür değişiklikler, diğer MCA eşgörünümleri tarafından kullanılan kanal tanımını etkilemez. Herhangi bir karakter (ikili veriler de içinde olmak üzere) kullanılabilir.

Bu alan yalnızca *ChannelType* MQCHT_REQUESTER, MQCHT_CLUSRCVR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

Bu alanın uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir. *Version* , MQCD_VERSION_3' den küçükse bu alan yoktur.

Bu alan IBM MQ for IBM i ile ilgili değildir.

MsgUserVerileri (MQCHAR32)

Bu alan, kanal iletişi çıkışı kullanıcı verilerini belirtir.

Bu veriler, **ChannelExitParms** değiştirgesinin *ExitData* alanındaki kanal iletişi çıkışına geçirilir (MQ_CHANNEL_EXIT konusuna bakın).

Bu alan başlangıçta kanal tanımında ayarlanan verileri içerir. Ancak, bu MCA örneğinin yaşam süresi boyunca, bu alanın içeriğinde herhangi bir tip çıkışla yapılan değişiklikler MCA tarafından korunur ve bu MCA örneğine ilişkin çıkışların sonraki çağrılmasında (türüne bakılmaksızın) görünür hale getirilir. Bu tür değişiklikler, diğer MCA eşgörünümleri tarafından kullanılan kanal tanımını etkilemez. Herhangi bir karakter (ikili veriler de içinde olmak üzere) kullanılabilir.

Bu alanın uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir.

Bu alan IBM MQ for IBM i ile ilgili değildir.

MsgUserDataPtr (MQPTR)

Bu alan, ilk *MsgUserData* alanının adresini belirtir.

MsgExitsDefined sıfırdan büyükse, bu adres zincirdeki her kanal ileti çıkışı için kullanıcı veri öğeleri listesinin adresidir.

Her kullanıcı veri öğesi, *ExitDataLength* uzunluğundaki bir alandır ve sağa doğru boşluklarla doldurulur. Her çıkış için bir tane olmak üzere birbirine bitişik *MsgExitsDefined* alanları vardır. Tanımlanan kullanıcı veri öğesi sayısı çıkış adı sayısından azsa, tanımlanmamış kullanıcı veri öğeleri boş olarak ayarlanır. Ters durumda, tanımlanan kullanıcı veri öğesi sayısı çıkış adı sayısından fazlaysa, fazla kullanıcı veri öğeleri yoksayıılır ve çıkışa sunulmaz.

Bu değerlerde bir çıkış tarafından yapılan değişiklikler korunur. Bu, bir çıkışın bilgileri başka bir çıkışa geçirmesini sağlar. Herhangi bir değişiklik üzerinde geçerlilik denetimi gerçekleştirilmez; örneğin, gerekirse ikili veriler bu alanlara yazılabilir.

MsgExitsDefined sıfırsa, bu alan boş gösterge olur.

Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir.

Bu, çıkışa bir giriş alanıdır. *Version* , M0CD_VERSION_4' den küçükse alan yoktur.

NetworkPriority (MQLONG)

Bu alan, kanala ilişkin ağ bağlantısının önceliğini belirler.

Belirli bir hedefe giden birden çok yol varsa, en yüksek önceliğe sahip yol seçilir. Değer 0-9 aralığındadır; 0 en düşük önceliklidir.

Bu alan yalnızca *ChannelType* M0CHT_CLUSSDR ya da M0CHT_CLUSRCVR içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , M0CD_VERSION_5' den küçükse alan yoktur.

Version , M0CD_VERSION_6' dan küçükse, bu yapıdaki aşağıdaki alanlar yoktur.

NonPersistentMsgSpeed (MQLONG)

Bu alan, kalıcı olmayan iletilerin kanal üzerinden geçiş hızını belirtir.

Bu alan yalnızca *ChannelType* M0CHT_SENDER, M0CHT_SERVER, M0CHT_ALICI, M0CHT_REQUESTER, M0CHT_CLUSSDR ya da M0CHT_CLUSRCVR içeren kanallar için geçerlidir.

Değer aşağıdakilerden biridir:

M0NPMS_NORMAL

Normal hız.

Bir kanal M0NPMS_NORMAL olarak tanımlanırsa, kalıcı olmayan iletiler kanal üzerinden normal hızda hareket eder. Bu, bir kanal hatası oluşursa bu iletilerin kaybolmaması avantajına sahiptir. Ayrıca, aynı iletim kuyruğundaki kalıcı ve kalıcı olmayan iletiler, sıralarını birbirlerine göre korur.

M0NPMS_FAST

Hızlı hızlı.

Bir kanal M0NPMS_FAST olarak tanımlanırsa, kalıcı olmayan iletiler kanal üzerinden hızlı bir şekilde seyahat eder. Bu, kanalın verimini artırır, ancak bir kanal hatası oluşursa kalıcı olmayan iletilerin kaybolduğu anlamına gelir. Ayrıca, kalıcı olmayan iletilerin aynı iletim kuyruğunda bekleyen kalıcı iletilerin önüne atılması mümkündür; yani, kalıcı olmayan iletilerin sırası kalıcı iletilere göre korunmaz. Ancak, kalıcı olmayan iletilerin birbiriyle göreceli sırası korunur. Benzer şekilde, birbiriyle göreceli kalıcı iletilerin sırası korunur.

Parola (M0CHAR12)

Bu alan, uzak bir ileti kanalı aracısıyla güvenli bir SNA oturumu başlatma girişimi sırasında ileti kanalı aracısı tarafından kullanılan parolayı belirtir.

Bu alan yalnızca AIX, Linux, and Windows üzerinde boş bırakılabilir ve yalnızca *ChannelType* M0CHT_SENDER, M0CHT_SERVER, M0CHT_REQUESTER ya da M0CHT_CLNTCONN içeren kanallar için geçerlidir. z/OS' ta bu alan ilgili değildir.

Bu alanın uzunluğu M0_PASSWORD_LENGTH tarafından verilir. Ancak, yalnızca ilk 10 karakter kullanılır.

Version , MQCD_VERSION_2' den küçükse bu alan yoktur.

PropertyControl (MQLONG)

Bu alan, ileti bir V6 ya da önceki bir kuyruk yöneticisine (bir özellik tanımlayıcısı kavramını anlamayan bir kuyruk yöneticisi) gönderilmek üzereyken iletilerin özelliklerine ne olacağını belirler.

Değer, aşağıdaki değerlerden herhangi biri olabilir:

MQPROP_COMPATIBILITY

İleti **mcd.**, **jms.**, **usr.** ya da **mqext.** öneğine sahip bir özellik içeriyorsa, tüm ileti özellikleri bir MQRFH2 üstbilgisinde uygulamaya teslim edilir. Ters durumda, ileti tanımlayıcısında (ya da uzantıda) yer alan özellikler dışında, iletinin tüm özellikleri atılır ve artık uygulama bunlara erişemez.

Bu değer varsayılan değerdir; JMS ile ilgili özelliklerin ileti verilerindeki bir MQRFH2 üstbilgisinde olmasını beklenen uygulamaların değiştirilmeden çalışmaya devam etmesini sağlar.

MQPROP_NONE

İleti tanımlayıcıdaki (ya da uzantıdaki) özellikler dışında, iletinin tüm özellikleri, ileti uzak kuyruk yöneticisine gönderilmeden önce iletiden kaldırılır.

MQPROP_Tümü

İletinin tüm özellikleri, uzak kuyruk yöneticisine gönderildiğinde iletiye eklenir. İleti tanımlayıcısındaki (ya da uzantısındaki) özellikler dışında özellikler, ileti verilerindeki bir ya da daha çok MQRFH2 üstbilgisine yerleştirilir.

Bu öznitelik, Gönderen, Sunucu, Küme Gönderen ve Küme Alıcı kanalları için geçerlidir.

“MQIA_ * (Tamsayı Öznitelik Seçicileri)” sayfa 128

“MQPROP_ * (Kuyruk ve Kanal Özellik Denetimi Değerleri ve Özellik Uzunluğu Üst Sınırı)” sayfa 168

PutAuthority (MQLONG)

Bu alan, bir iletiyle ilişkili bağlam bilgilerindeki kullanıcı kimliğinin, iletiyi hedef kuyruğa koyma yetkisi oluşturmak için kullanılıp kullanılmayacağını belirtir.

Bu alan yalnızca *ChannelType* MQCHT_REQUESTER, MQCHT_CLUSRCVR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir. Aşağıdakilerden biridir:

MQPA_DEFAULT

Varsayılan kullanıcı kimliği kullanılır.

MQPA_CONTEXT

Bağlam kullanıcı kimliği kullanıldı.

MQPA_ALTERNATE_OR_MCA

İleti tanımlayıcısının UserIdentifier alanındaki kullanıcı kimliği kullanılır. Ağdan alınan kullanıcı kimliği kullanılmaz. Bu değer yalnızca z/OS üzerinde desteklenir.

MQPA_ONLY_MCA

Varsayılan kullanıcı kimliği kullanılır. Ağdan alınan kullanıcı kimliği kullanılmaz. Bu değer yalnızca z/OS üzerinde desteklenir.

QMgrName (MQCHAR48)

Bu alan, çıkışın bağlanabileceği kuyruk yöneticisinin adını belirtir.

MQCHT_CLNTCONN dışında bir *ChannelType* içeren kanallar için bu alan, bir çıkışın bağlanabileceği kuyruk yöneticisinin adıdır; AIX, Linux, and Windows üzerinde bu ad her zaman boş değildir.

Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir.

ReceiveExit (MQCHARn)

Bu alan, kanal alma çıkış adını belirler.

Bu ad boş değilse, aşağıdaki zamanlarda çıkış çağrılır:

- Alınan ağ verileri işlenmeden hemen önce.

Çıkışa, alınan tüm iletim arabelleği verilir. Arabelleğin içeriği gerektiği gibi değiştirilebilir.

- Kanal başlatılırken ve sonlandırılırken.

Bu alanın çeşitli ortamlardaki içeriğinin açıklaması için bkz. "[MQCD-Kanal tanımlaması](#)" sayfa 1454 .

Bu alanın uzunluğu MQ_EXIT_NAME_LENGTH tarafından verilir.

Not: Bu değişimin değeri ortama özgüdür.

ReceiveExitPtr (MQPTR)

Bu alan, ilk *ReceiveExit* alanının adresini belirtir.

ReceiveExitsDefined sıfırdan büyükse, bu adres zincirdeki her bir kanal alma çıkışının ad listesinin adresidir.

Her ad, *ExitNameLength* uzunluğundaki bir alandır ve sağa boşluklarla doldurulur. Her çıkış için bir tane olmak üzere birbirine bitişik *ReceiveExitsDefined* alanları vardır.

Bu adlarda çıkışla yapılan değişiklikler korunur; ancak, ileti kanalı çıkışı belirtik bir işlem gerçekleştirmez; hangi çıkışların çağrılacağını değiştirmez.

ReceiveExitsDefined sıfırsa, bu alan boş gösterge olur.

Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizisi olarak bildirilir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

ReceiveExitsTanımlı (MQLONG)

Bu alan, zincirde tanımlanan kanal alma çıkışlarının sayısını belirtir.

Sıfırdan büyük ya da sıfıra eşittir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

ReceiveUserVerileri (MQCHAR32)

Bu kanal, kanal alma çıkışı kullanıcı verilerini belirtir.

Bu veriler, **ChannelExitParms** değiştirgesinin *ExitData* alanındaki kanal alma çıkışına geçirilir (bkz. MQ_CHANNEL_EXIT).

Bu alan başlangıçta kanal tanımında ayarlanan verileri içerir. Ancak, bu MCA örneğinin yaşam süresi boyunca, bu alanın içeriğinde herhangi bir tip çıkışla yapılan değişiklikler MCA tarafından korunur ve bu MCA örneğine ilişkin çıkışların sonraki çağrılmasında (türüne bakılmaksızın) görünür hale getirilir. Bu, farklı konuşmalardaki çıkışlar için geçerlidir. Bu tür değişiklikler, diğer MCA eşgörünümleri tarafından kullanılan kanal tanımını etkilemez. Herhangi bir karakter (ikili veriler de içinde olmak üzere) kullanılabilir.

Bu alanın uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir.

Bu alan IBM MQ for IBM i ile ilgili değildir.

Version , MQCD_VERSION_2' den küçükse, bu yapıda aşağıdaki alanlar yoktur.

ReceiveUserDataPtr (MQPTR)

Bu alan, ilk *ReceiveUserData* alanının adresini belirtir.

ReceiveExitsDefined sıfırdan büyükse, bu adres zincirdeki her bir kanal alma çıkışı için kullanıcı verileri ögesi listesinin adresidir.

Her kullanıcı veri ögesi, *ExitDataLength* uzunluğundaki bir alandır ve sağa doğru boşluklarla doldurulur. Her çıkış için bir tane olmak üzere birbirine bitişik *ReceiveExitsDefined* alanları vardır. Tanımlanan kullanıcı veri ögesi sayısı çıkış adı sayısından azsa, tanımlanmamış kullanıcı veri öğeleri boş olarak ayarlanır. Ters durumda, tanımlanan kullanıcı veri ögesi sayısı çıkış adı sayısından fazlaysa, fazla kullanıcı veri öğeleri yoksayılr ve çıkış sunulmaz.

Bu değerlerde bir çıkış tarafından yapılan değişiklikler korunur. Bu, bir çıkışın bilgileri başka bir çıkışa geçirmesini sağlar. Herhangi bir değişiklik üzerinde geçerlilik denetimi gerçekleştirilmez; örneğin, gerekirse ikili veriler bu alanlara yazılabilir.

ReceiveExitsDefined sıfırsa, bu alan boş gösterge olur.

Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizisi olarak bildirilir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

Version , MQCD_VERSION_5' den küçükse, bu yapıdaki aşağıdaki alanlar yoktur.

RemotePassword (MQCHAR12)

Bu alan, bir iş ortağından gelen parolayı belirtir.

Bu alan yalnızca *ChannelType* MQCHT_CLNTCONN ya da MQCHT_SVRCONN ise geçerli bilgiler içerir.

- MQCHT_CLNTCONN kanalındaki bir güvenlik çıkışı için bu parola, ortamdaki alandan alınan bir paroladır. Çıkış, sunucunun güvenlik çıkışına göndermeyi seçebilir.
- MQCHT_SVRCONN kanalındaki bir güvenlik çıkışında, istemci güvenlik çıkışı yoksa, bu alanda istemcideki ortamdaki elde edilen bir parola bulunabilir. Çıkış, *RemoteUserIdentifier* içindeki kullanıcı kimliğini doğrulamak için bu parolayı kullanabilir.

İstemcide bir güvenlik çıkışı varsa, bu bilgiler istemciden bir güvenlik akışında alınabilir.

Bu alanın uzunluğu MQ_PASSWORD_LENGTH tarafından verilir. *Version* , MQCD_VERSION_2' den küçükse bu alan yoktur.

RemoteSecurityTanıtıcısı (MQBYTE40)

Bu alan, uzak kullanıcıya ilişkin güvenlik tanıtıcısını belirler.

Bu alan yalnızca *ChannelType* MQCHT_CLNTCONN ya da MQCHT_SVRCONN içeren kanallar için geçerlidir.

Aşağıdaki özel değer, güvenlik tanıtıcısı olmadığını gösterir:

MQSID_YOK

Güvenlik tanıtıcısı belirtilmedi.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQSID_NONE_ARRAY değişmezi de tanımlanır; bu değişmez MQSID_NONE ile aynı değere sahiptir, ancak dizgi yerine bir karakter dizisidir.

Bu, çıkışa bir giriş alanıdır. Bu alanın uzunluğu MQ_SECURITY_ID_LENGTH tarafından verilir. *Version* , MQCD_VERSION_6' dan küçükse bu alan yoktur.

Version , MQCD_VERSION_7' den küçükse, bu yapıdaki aşağıdaki alanlar yoktur.

RemoteUserTanıtıcısı (MQCHAR12)

Bu alan, bir iş ortağından gelen kullanıcı kimliğinin ilk 12 baytını belirler.

Uzak kullanıcı kimliğini içeren iki alan vardır:

- *RemoteUserIdentifier* uzak kullanıcı kimliğinin ilk 12 baytını içerir ve tanıtıcı 12 bayttan kısaysa boşluklarla doldurulur. *RemoteUserIdentifier* boş olabilir.
- *LongRemoteUserIdPtr* , 12 bayttan uzun olabilen tam uzak kullanıcı kimliğini gösterir. Uzunluğu *LongRemoteUserIdLength* tarafından verilir. Tam tanıtıcı sonda boşluk içermiyor ve boş sonlandırılmış değil. Tanıtıcı boşsa, *LongRemoteUserIdLength* sıfırdır ve *LongRemoteUserIdPtr* değeri tanımlıdır.

Version , MQCD_VERSION_6' dan küçükse *LongRemoteUserIdPtr* yoktur.

Uzak kullanıcı kimliği yalnızca *ChannelType* MQCHT_CLNTCONN ya da MQCHT_SVRCONN içeren kanallar için geçerlidir.

- MQCHT_CLNTCONN kanalındaki bir güvenlik çıkışı için bu değer, ortamdan elde edilen bir kullanıcı kimliğidir. Çıkış, sunucunun güvenlik çıkışına göndermeyi seçebilir.
- MQCHT_SVRCONN kanalındaki bir güvenlik çıkışında, istemci güvenlik çıkışı yoksa, bu alanda istemcideki ortamdan elde edilen bir kullanıcı kimliği bulunabilir. Çıkış, bu kullanıcı kimliğini doğrulayabilir (büyük olasılıkla *RemotePassword* içindeki parolayla) ve *MCAUserIdentifier* içindeki değeri güncelleyebilir.

İstemcide bir güvenlik çıkışı varsa, bu bilgiler istemciden bir güvenlik akışında alınabilir.

Bu alanın uzunluğu MQ_USER_ID_LENGTH tarafından verilir. *Version* , MQCD_VERSION_2' den küçükse bu alan yoktur.

SecurityExit (MQCHARn)

Bu alan, kanal güvenliği çıkış adını belirler.

Bu ad boş değilse, aşağıdaki zamanlarda çıkış çağrılır:

- Bir kanal kurduktan hemen sonra.
Herhangi bir ileti aktarılmadan önce, çıkışa bağlantı yetkilendirmesini doğrulamak için güvenlik akışlarını teşvik etme fırsatı verilir.
- Bir güvenlik iletisi akışına yanıt alındıktan sonra.
Uzak makinedeki uzak işlemciden alınan güvenlik iletisi akışları çıkışa verilir.
- Kanal başlatılırken ve sonlandırılırken.

Bu alanın çeşitli ortamlardaki içeriğinin açıklaması için bkz. [“MQCD-Kanal tanımlaması” sayfa 1454](#) .

Bu alanın uzunluğu MQ_EXIT_NAME_LENGTH tarafından verilir.

Not: Bu değişmez değeri ortama özgüdür.

SecurityUserVerileri (MQCHAR32)

Bu alan, kanal güvenliği çıkışı kullanıcı verilerini belirler.

Bu veriler, **ChannelExitParms** değiştirgesinin *ExitData* alanındaki kanal güvenliği çıkışına geçirilir (bkz. MQ_CHANNEL_EXIT).

Bu alan başlangıçta kanal tanımında ayarlanan verileri içerir. Ancak, bu MCA örneğinin yaşam süresi boyunca, bu alanın içeriğinde herhangi bir tip çıkışla yapılan değişiklikler MCA tarafından korunur ve bu MCA örneğine ilişkin çıkışların sonraki çağrılmasında (türüne bakılmaksızın) görünür hale getirilir. Bu, farklı konuşmalardaki çıkışlar için geçerlidir. Bu tür değişiklikler, diğer MCA eşgörünümleri tarafından kullanılan kanal tanımlaması üzerinde etkili olmaz. Herhangi bir karakter (ikili veriler de içinde olmak üzere) kullanılabilir.

Bu alanın uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir.

Bu alan IBM MQ for IBM i ile ilgili değildir.

SendExit (MQCHARn)

Bu alan, kanal gönderme çıkış adını belirler.

Bu ad boş değilse, aşağıdaki zamanlarda çıkış çağrılır:

- Veriler ağa gönderilmeden hemen önce.
Çıkışa, iletilmeden önce tam iletim arabelleği verilir. Arabelleğin içeriği gerektiği gibi değiştirilebilir.
- Kanal başlatılırken ve sonlandırılırken.

Bu alanın çeşitli ortamlardaki içeriğinin açıklaması için bkz. [“MQCD-Kanal tanımlaması” sayfa 1454](#) .

Bu alanın uzunluğu MQ_EXIT_NAME_LENGTH tarafından verilir.

Not: Bu değişmez değeri ortama özgüdür.

SendExitPtr (MQPTR)

Bu alan, ilk *SendExit* alanının adresini belirtir.

SendExitsDefined sıfırdan büyükse, bu adres zincirdeki her kanal gönderme çıkışının ad listesinin adresidir.

Her ad, *ExitNameLength* uzunluğundaki bir alandır ve sağa boşluklarla doldurulur. Her çıkış için bir tane olmak üzere birbirine bitişik *SendExitsDefined* alanları vardır.

Bu adlarda bir çıkış tarafından yapılan değişiklikler korunur; ancak, ileti gönderme çıkışı belirttik bir işlem gerektirmez; hangi çıkışların çağrılacağını değiştirmez.

SendExitsDefined sıfırsa, bu alan boş gösterge olur.

Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

SendExitsTanımlı (MQLONG)

Bu alan, zincirde tanımlanan kanal gönderme çıkışlarının sayısını belirtir.

Sıfırdan büyük ya da sıfıra eşittir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

SendUserData (MQCHAR32)

Bu alan, kanal gönderme çıkış kullanıcı verilerini belirler.

Bu veriler, **ChannelExitParms** değiştirgesinin *ExitData* alanındaki kanal gönderme çıkışına geçirilir (bkz. MQ_CHANNEL_EXIT).

Bu alan başlangıçta kanal tanımında ayarlanan verileri içerir. Ancak, bu MCA örneğinin yaşam süresi boyunca, bu alanın içeriğinde herhangi bir tip çıkışla yapılan değişiklikler MCA tarafından korunur ve bu MCA örneğine ilişkin çıkışların sonraki çağrılmasında (türüne bakılmaksızın) görünür hale getirilir. Bu, farklı konuşmalardaki çıkışlar için geçerlidir. Bu tür değişiklikler, diğer MCA eşgörünümleri tarafından kullanılan kanal tanımını etkilemez. Herhangi bir karakter (ikili veriler de içinde olmak üzere) kullanılabilir.

Bu alanın uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir.

Bu alan IBM MQ for IBM i ile ilgili değildir.

SendUserDataPtr (MQPTR)

Bu alan, *SendUserData* alanının adresini belirtir.

SendExitsDefined sıfırdan büyükse, bu adres zincirdeki her kanal ileti çıkışı için kullanıcı veri öğeleri listesinin adresidir.

Her kullanıcı veri öğesi, *ExitDataLength* uzunluğundaki bir alandır ve sağa doğru boşluklarla doldurulur. Her çıkış için bir tane olmak üzere birbirine bitişik *MsgExitsDefined* alanları vardır. Tanımlanan kullanıcı veri öğesi sayısı çıkış adı sayısından azsa, tanımlanmamış kullanıcı veri öğeleri boş olarak ayarlanır. Ters durumda, tanımlanan kullanıcı veri öğesi sayısı çıkış adı sayısından fazlaysa, fazla kullanıcı veri öğeleri yoksayıılır ve çıkışa sunulmaz.

Bu değerlerde bir çıkış tarafından yapılan değişiklikler korunur. Bu, bir çıkışın bilgileri başka bir çıkışa geçirmesini sağlar. Herhangi bir değişiklik üzerinde geçerlilik denetimi gerçekleştirilmez; örneğin, gerekirse ikili veriler bu alanlara yazılabilir.

SendExitsDefined sıfırsa, bu alan boş gösterge olur.

Programlama dilinin gösterge veri tipini desteklemediği platformlarda, bu alan uygun uzunlukta bir bayt dizgisi olarak bildirilir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_4' den küçükse alan yoktur.

SeqNumberKaydır (MQLONG)

Bu alan, izin verilen en yüksek ileti sıra numarasını belirtir.

Bu değere ulaşıldığında, sıra numaraları 1 'den yeniden başlayacak şekilde kaydırılır.

Bu değer kararlaştırılmaz ve hem yerel hem de uzak kanal tanımlarında eşleşmesi gerekir.

Bu alan, *ChannelType* MQCHT_SVRCONN ya da MQCHT_CLNTCONN içeren kanallar için geçerli değildir.

SharingConversations (MQLONG)

Bu alan, bu kanalla ilişkilendirilmiş bir kanal örneğini paylaşabilecek etkileşim sayısı üst sınırını belirtir.

Bu alan, istemci bağlantısı ve sunucu bağlantısı kanallarında kullanılır.

0 değeri, kanalın aşağıdaki özniteliklerle ilgili olarak IBM WebSphere MQ 7.0 ' den önceki sürümlerde olduğu gibi çalıştığı anlamına gelir:

- Etkileşim paylaşımı
- Önden oku
- STOP CHANNEL(*channelname*) MODE(QUIESCE)
- Sağlıklı işletim bildirimi
- İstemci zamanuyumsuz tüketimi

1 değeri, IBM MQ davranışı için alt sınır değeridir. Kanal örneğinde yalnızca bir etkileşimin olmasına izin verilse de, önden okuma, zamanuyumsuz tüketim ve CLNTCONN-SVRCONN sağlıklı işletim bildirimi ve sessiz kanal durdurma davranışı kullanılabilir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_9' dan küçükse bu yoktur.

Bu alanın varsayılan değeri 10 'dur.

Not: Bir kanala uygulanan *MaxInstances* ve *MaxInstancesPerClient* sınırları, bu eşgörünümleri paylaşan etkileşimlerin sayısını değil, kanal eşgörünümlerinin sayısını sınırlar.

ShortConnectionAdı (MQCHAR20)

Bu alan, bir bağlantı adının ilk 20 baytını belirler.

Version alanı MQCD_VERSION_1ise, *ShortConnectionName* tam bağlantı adını içerir.

Version alanı MQCD_VERSION_2 ya da daha büyükse, *ShortConnectionName* bağlantı adının ilk 20 karakterini içerir. Tam bağlantı adı *ConnectionName* alanı tarafından verilir; *ShortConnectionName* ve ilk 20 *ConnectionName* karakteri aynıdır.

Bu alanın içeriğinin ayrıntıları için bkz. *ConnectionName* .

Not: Bu alanın adı MQCD_VERSION_2 ve sonraki MQCD sürümleri için değiştirildi; alan daha önce *ConnectionName*olarak adlandırıldı.

Bu alanın uzunluğu MQ_SHORT_CONN_NAME_LENGTH tarafından verilir.

ShortRetrySayısı (MQLONG)

Bu alan, uzak bir makineye bağlanmak için yapılan deneme sayısı üst sınırını belirler.

Bu alan, (normalde daha uzun) *LongRetryCount* ve *LongRetryInterval* kullanılmadan önce, *ShortRetryInterval*tarafından belirtilen aralıklarla uzak makineye bağlanmak için yapılan deneme sayısı üst sınırınıdır.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

ShortRetryAralığı (MQLONG)

Bu alan, uzak makineyle bağlantıyı yeniden denemeden önce beklenecek saniye sayısı üst sınırını belirler.

Kanalın etkinleşmesi için beklemesi gerekirse, yeniden denemeler arasındaki aralık uzatılabilir.

Bu alan yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_CLUSSDR ya da MQCHT_CLUSRCVR içeren kanallar için geçerlidir.

► z/OS *SPLProtection (MQLONG)*

Bu alan, AMS güvenlik ilkesi korumasının değerini belirtir.

Değer aşağıdakilerden biridir:

MQSPL_PASSTHRU

Bu kanal için MCA tarafından gönderilen ya da alınan iletilerin üzerinden geçin, değiştirilmedi.

Bu değer yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_ALICI ya da MQCHT_REQUESTER içeren kanallar için geçerlidir ve varsayılan değerdir.

MQSPL_REMOVE

MCA tarafından iletim kuyruğundan alınan iletilerden AMS korumasını kaldırın ve iletileri iş ortağına gönderin.

Bu değer yalnızca *ChannelType* MQCHT_SENDER ya da MQCHT_SERVER içeren kanallar için geçerlidir.

MQSPL_ ASPOLICY

Hedef kuyruk için tanımlanan ilkeye dayalı olarak, hedef kuyruğa yerleştirmeden önce gelen iletilere AMS korumasını uygulayın.

Bu değer yalnızca *ChannelType* MQCHT_ALICI ya da MQCHT_REQUESTER içeren kanallar için geçerlidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_12' den küçükse bu alan yoktur.

SSLCipherSpec (MQCHAR32)

Bu alan, TLS kullanılırken kullanılmakta olan Şifre Belirtimini belirtir.

SSLCipherSpec boşsa, kanal TLS kullanmaz. Boş değilse, bu alan kullanılmakta olan CipherSpec öğesini belirten bir dizgi içerir.

Bu parametre tüm kanal tipleri için geçerlidir. Aşağıdaki altyapılarda desteklenir:

- AIX AIX
- IBM i IBM i
- Linux Linux
- Windows Windows
- z/OS z/OS

Yalnızca TCP iletim tipinin (TRPTYPE) kanal tipleri için geçerlidir.

Bu, çıkışa bir giriş alanıdır. Bu alanın uzunluğu MQ_SSL_CIPHER_SPEC_LENGTH tarafından verilir. *Version* , MQCD_VERSION_7' den küçükse alan yoktur.

SSLClientAuth (MQLONG)

Bu alan, TLS istemci kimlik doğrulamasının gerekli olup olmadığını belirtir.

Bu alan yalnızca SVRCONN kanal tanımlamaları için geçerlidir.

Aşağıdaki değerlerden biridir:

MQSCA_REQUIRED

İstemci kimlik doğrulaması gerekiyor.

MQSCA_OPTIONAL

İstemci kimlik doğrulaması isteğe bağlıdır.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_7' den küçükse alan yoktur.

SSLPeerNameUzunluğu (MQLONG)

Bu alan, *SSLPeerNamePtr* ile gösterilen TLS eş adının bayt cinsinden uzunluğunu belirtir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_7' den küçükse alan yoktur.

SSLPeerNamePtr (MQPTR)

Bu alan TLS eş adının adresini belirtir.

Başarılı bir TLS anlaşması sırasında bir sertifika alındığında, sertifikanın konusunun Ayırt Edici Adı, sertifikayı alan kanalın sonundaki SSLPeerNamePtr ile erişilen MQCD alanına kopyalanır. Bu değer, yerel kullanıcının kanal tanımında varsa, kanala ilişkin SSLPeerName değerinin üzerine yazar. Kanalın bu ucunda bir güvenlik çıkışı belirtilirse, MQCD ' deki eşdüzey sertifikadan Ayırt Edici Adı alır.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCD_VERSION_7' den küçükse alan yoktur.

Not: IBM WebSphere MQ 7.1 yayın düzeyinden önce oluşturulan güvenlik çıkışı uygulamalarının güncellenmesi gerekebilir. Daha fazla bilgi için bkz. [Kanal güvenliği çıkış programları](#).

StrucLength (MQLONG)

Bu alan, MQCD yapısının bayt cinsinden uzunluğunu belirtir.

Uzunluk, yap içinde yer alan g " sterge alanlarıyla adreslenen dizilimleri içermez. Değer aşağıdakilerden biridir:

MQCD_LENGTH_4

version-4 kanal tanımlama yapısının uzunluğu.

MQCD_LENGTH_5

version-5 kanal tanımlama yapısının uzunluğu.

MQCD_LENGTH_6

version-6 kanal tanımlama yapısının uzunluğu.

MQCD_LENGTH_7

version-7 kanal tanımlama yapısının uzunluğu.

MQCD_LENGTH_8

version-8 kanal tanımlama yapısının uzunluğu.

MQCD_LENGTH_9

version-9 kanal tanımlama yapısının uzunluğu.

MQCD_LENGTH_10

version-10 kanal tanımlama yapısının uzunluğu.

MQCD_LENGTH_11

version-11 kanal tanımlama yapısının uzunluğu.

z/OS MQCD_LENGTH_12

version-12 kanal tanımlama yapısının uzunluğu.

Aşağıdaki değişmez, yürürlükteki sürümün uzunluğunu belirtir:

MQCD_CURRENT_LENGTH

Kanal tanımlama yapısının yürürlükteki sürümünün uzunluğu.

Not: Bu sabitlerin ortama özgü değerleri vardır.

Version , MQCD_VERSION_4' den küçükse alan yoktur.

TpName (MQCHAR64)

Bu alan, LU 6.2 hareket programı adını belirtir.

Bu alan yalnızca iletim protokolü (*TransportType*) MQXPT_LU62ise ve *ChannelType* MQCHT_SVRCONN ya da MQCHT_ALICI değilse geçerlidir.

Bu alan, bilgilerin iletişim tarafı nesnesinde bulunduğu altyapılarda her zaman boştur.

Bu alanın uzunluğu MQ_TP_NAME_LENGTH tarafından verilir.

TransportType (MQLONG)

Bu alan, kullanılacak iletim protokolünü belirler.

Kanal diđer uçtan başlatıldıysa, bu deđer denetlenmez.

Aşağıdaki deđerlerden biridir:

MQXPT_LU62

LU 6.2 iletim protokolü.

MQXPT_TCP

TCP/IP iletim protokolü.

MQXPT_NETBIOS

NetBIOS aktarım iletişim kuralı.

Bu deđer şu ortamlarda desteklenir: Windows.

MQXPT_SPX

SPX taşıma protokolü.

Bu deđer, şu ortamlarda desteklenir: Windowsve bu sistemlere bađlı IBM MQ istemcileri.

UseDLQ (MQLONG)

Bu alan, iletiler kanallar tarafından teslim edilemediğinde, teslim edilmeyen ileti kuyruğunun (ya da teslim edilmeyen ileti kuyruğunun) kullanılıp kullanılmayacağını belirler.

Aşağıdaki deđerlerden birini içerebilir:

MQUSEDLQ_NO

Bir kanal tarafından teslim edilemeyen iletiler hata olarak kabul edilir. Kanal, NPMSPEED ayarına uygun olarak iletiyi atar ya da kanal sona erer.

MQUSEDLQ_YES

DEADQ kuyruk yöneticisi özniteliđi bir kullanılmayan ileti kuyruğunun adını sağladığında, bu işlem HAYIR deđeriyle ilgili olur. YES varsayılan deđerdir.

UserIdentifier (MQCHAR12)

Bu alan, uzak ileti kanalı aracısıyla güvenli bir SNA oturumu başlatma girişimi sırasında ileti kanalı aracısı tarafından kullanılan kullanıcı kimliğini belirtir.

Bu alan yalnızca AIX, Linux, and Windowsüzerinde boş bırakılabilir ve yalnızca *ChannelType* MQCHT_SENDER, MQCHT_SERVER, MQCHT_REQUESTER ya da MQCHT_CLNTCONN içeren kanallar için geçerlidir. z/OS' ta bu alan ilgili deđildir.

Bu alanın uzunluđu MQ_USER_ID_LENGTH tarafından verilir. Ancak, yalnızca ilk 10 karakter kullanılır.

Version , MQCD_VERSION_2' den küçükse bu alan yoktur.

Sürüm (MQLONG)

Version alanı, yapı için ayarlayabileceğiniz en yüksek sürüm numarasını belirtir.

Deđer ortama bađlıdır:

MQCD _VERSION_1

Sürüm 1 kanal tanımlama yapısı.

MQCD _VERSION_2

Sürüm 2 kanal tanımlama yapısı.

MQCD _VERSION_3

Sürüm 3 kanal tanımlama yapısı.

MQCD _VERSION_4

Sürüm 4 kanal tanımlama yapısı.

MQCD _VERSION_5

Sürüm 5 kanal tanımlama yapısı.

MQCD _VERSION_6

Sürüm 6 kanal tanımlama yapısı.

MQCD _VERSION_7

Sürüm 7 kanal tanımlama yapısı.

MQCD _VERSION_8

Sürüm 8 kanal tanımlama yapısı.

MQCD _VERSION_9

Sürüm 9 kanal tanımlama yapısı.

MQCD _VERSION_10

Sürüm 10 kanal tanımlama yapısı.

MQCD _VERSION_11

Sürüm 11 kanal tanımlama yapısı.

Sürüm 11, alanı tüm platformlarda IBM MQ 8.0 üzerinde ayarlayabileceğiniz en yüksek düzeydir.

z/OS MQCD _VERSION_12

Sürüm 12 kanal tanımlama yapısı.

Sürüm 12, alanı IBM MQ 9.1.3 üzerinde ayarlayabileceğiniz en yüksek düzeydir.

Yalnızca yapının daha yeni sürümlerinde var olan alanlar, alanların tanımlarında bu şekilde tanımlanır.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQCD_CURRENT_VERSION

MQCD_CURRENT_VERSION içinde ayarlanan değer, kullanılmakta olan kanal tanımlama yapısının geçerli sürümüdür.

MQCD_CURRENT_VERSION değeri ortama bağlıdır. Platform tarafından desteklenen en yüksek değeri içerir.

MQCD_CURRENT_VERSION , üstbilgide sağlanan varsayılan yapıları kullanıma hazırlamak, kopyalamak ve farklı programlama dilleri için sağlanan dosyaları eklemek için kullanılmaz. 'Version' in varsayılan kullanıma hazırlanması, platforma ve yayına bağlıdır.

Üstbilgi, kopyalama ve içerme dosyalarındaki MQCD bildirimleri MQCD_VERSION_6 olarak başlatılır. Ek MQCD alanlarını kullanmak için uygulamaların sürüm numarasını MQCD_CURRENT_VERSION olarak ayarlaması gerekir. Birkaç ortam arasında taşınabilir bir uygulama yazıyorsanız, tüm ortamlarda desteklenen bir sürüm seçmeniz gerekir.

İpucu: MQCD yapısının yeni bir sürümü kullanıma sunulduğunda, var olan parçanın düzeni değişmez. Çıkışın sürüm numarasını denetlemesi gerekir. Çıkışın kullanması gereken alanları içeren en düşük sürüme eşit ya da bu sürümden büyük olmalıdır.

XmitQName (MQCHAR48)

Bu alan, iletilerin alındığı iletim kuyruğunun adını belirler.

Bu alan yalnızca *ChannelType* MQCHT_SENDER ya da MQCHT_SERVER içeren kanallar için geçerlidir.

Bu alanın uzunluğu MQ_Q_NAME_LENGTH tarafından verilir.

C bildirim

Bu bildirim, MQCD yapısına ilişkin C bildirimini belirtir.

```
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 */

```

```

MQLONG    NetworkPriority;          /* which the channel belongs */
/* Ver:5 */                          /* Network priority */
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 */
/* instances */
MQLONG    MaxInstancesPerClient;  /* Limit on SVRCONN channel */
/* instances per client */
MQLONG    ClientChannelWeight;    /* Client channel weight */
MQLONG    ConnectionAffinity;     /* Connection affinity */
/* Ver:9 */
MQLONG    BatchDataLimit;         /* Batch data limit */
MQLONG    UseDLQ;                /* Use Dead Letter Queue */
MQLONG    DefReconnect;          /* Default client reconnect */
/* option */
/* Ver:10 */
MQCHAR64  CertificateLabel;       /* Certificate label */
/* Ver:11 */
MQLONG    SPLProtection           /* AMS Security policy protection */
/* Ver:12 */
};

```

COBOL bildirim

Bu bildirim, MQCD yapısına ilişkin COBOL bildirimini belirtir.

```

** 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-NONPERSISTENTMSGSPEED PIC S9(9) BINARY.
** Length of MQCD structure
  15 MQCD-STRUCLength 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-MCASESECURITYID PIC X(40).
** Remote security identifier
15 MQCD-REMOTESECURITYID 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 **

```

RPG bildirim (ILE)

Bu bildirim, MQCD yapısına ilişkin RPG bildirimini belirtir.

```

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 CDLML 1097 1100I 0
D* Length of long remote user identifier
D CDLRL 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 **

```

System/390 çevirici bildirimi

Bu bildirim, MQCD yapısına ilişkin System/390 çevirici bildirimini gösterir.

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
*			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_NONPERSISTENTMSGSPPEED	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_REMOTESECURITYID	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 Affinity
MQCD_BATCHDATALIMIT	DS	F	Batch data limit
MQCD_USEDLDQ	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)	

Visual Basic bildirimi

Bu bildirim, MQCD yapısının Visual Basic bildirimini belirtir.

Visual Basic 'de, MQCD yapısı MQCONNX çağrısında MQCNO yapısıyla birlikte kullanılabilir.

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' 'identifier'
LongMCAUserIdPtr	As MQPTR	'Address of long MCA user' 'identifier'
LongRemoteUserIdPtr	As MQPTR	'Address of long remote user' 'identifier'
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		

Kanal çıkışında MQCD alanlarının değiştirilmesi

Bir kanal çıkışı MQCD 'deki alanları değiştirebilir. Ancak, listelenen koşullar dışında, bu değişiklikler genellikle üzerinde işlem yapılmaz.

Bir kanal çıkış programı MQCD veri yapısındaki bir alanı değiştirirse, yeni değer genellikle IBM MQ kanal işlemi tarafından yoksayılır. Ancak, yeni değer MQCD 'de kalır ve bir çıkış zincirinde kalan çıkışlara ve kanal yönetim ortamını paylaşan sohbetlere geçirilir.

MQCXP yapısında SharingConversations FALSE olarak ayarlanırsa, çıkış programının tipine, tipine ve çıkış neden koduna bağlı olarak belirli alanlarda yapılan değişiklikler gerçekleştirilebilir. Aşağıdaki tabloda, kanalın davranışını ve hangi koşullarda değiştirilebileceği ve etkileyebileceği alanlar gösterilmektedir. Bir çıkış programı bu alanlardan birini başka bir koşulda ya da listede yer almayan herhangi bir alanda değiştirirse, yeni değer kanal işlemi tarafından dikkate alınmaz. Yeni değer MQCD 'de kalır ve bir çıkış zincirinde kalan çıkışlara ve kanal yönetim ortamını paylaşan herhangi bir etkileşimde iletilir.

Kullanıma hazırlama çağrıldığında (MQXR_INIT) herhangi bir çıkış programı tipi, MQCXP SharingConverstions FALSE olarak ayarlandığı sürece, herhangi bir kanal tipinin ChannelName alanını değiştirebilir. MQCXP SharingConverstionsdeğerinden bağımsız olarak yalnızca bir güvenlik çıkışı MCAUserIdentifier alanını değiştirebilir.

<i>Çizelge 823. Değiştirilebilecek ve kanalın davranışını etkileyebilecek alanlar</i>			
Alan	Çıkış neden kodu	Çıkış tipi	Kanal tipi
ChannelName	MQXR_INIT	Tümü	Tümü
TransportType	MQXR_INIT	Tümü	Tümü
XmitQName	MQXR_INIT	Tümü	SDR, RCVR
ModeName	MQXR_INIT	Tümü	Tümü
TpName	MQXR_INIT	Tümü	Tümü
BatchSize	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
DiscInterval	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
ShortRetrySayısı	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
ShortRetryInterval (Kısayol Yeniden Deneme Aralığı)	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
LongRetrySayısı	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
LongRetryAralığı	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
SeqNumberKaydır	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
MaxMsgUzunluğu	MQXR_INIT	Tümü	Tümü

Çizelge 823. Değiştirilebilecek ve kanalın davranışını etkileyebilecek alanlar (devamı var)

Alan	Çıkış neden kodu	Çıkış tipi	Kanal tipi
PutAuthority	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
DataConversion	MQXR_INIT	Tümü	Tümü
MCAUserIdentifier	MQXR_INIT, MQXR_INIT_SEC, MQXR_SEC_MSG, MQXR_SEC_PARMS	Durumu	RCVR, RQSTR, SVRCONN, CLUSRCVR
ConnectionName	MQXR_INIT	Tümü	SDR, SVR, RQSTR, CLNTCONN, CLUSSDR, CLUSRCVR
MsgRetryUserData	MQXR_INIT	Tümü	RCVR, RQSTR, CLUSRCVR
MsgRetrySayısı	MQXR_INIT	Tümü	RCVR, RQSTR, CLUSRCVR
MsgRetryAralığı	MQXR_INIT	Tümü	RCVR, RQSTR, CLUSRCVR
HeartbeatInterval	MQXR_INIT	Tümü	Tümü
BatchInterval	MQXR_INIT	Tümü	SDR, SVR, CLUSSDR, CLUSRCVR
NonPersistentMsgSpeed	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
MCASecurityId	MQXR_INIT, MQXR_INIT_SEC, MQXR_SEC_MSG, MQXR_SEC_PARMS	Durumu	SDR, SVR, RCVR, RQSTR, SVRCONN, CLUSSDR, CLUSRCVR
SSLCipherSpec	MQXR_INIT	Tümü	Tümü
SSLPeerNamePtr	MQXR_INIT	Tümü	Tümü
SSLPeerNameUzunluğu	MQXR_INIT	Tümü	Tümü
SSLClientAuth	MQXR_INIT	Tümü	SVR, RCVR, RQSTR, SVRCONN, CLUSRCVR

Çizelge 823. Değiştirilebilecek ve kanalın davranışını etkileyebilecek alanlar (devamı var)			
Alan	Çıkış neden kodu	Çıkış tipi	Kanal tipi
KeepAliveAralığı	MQXR_INIT	Tümü	Tümü
LocalAddress	MQXR_INIT	Tümü	SDR, SVR, RQSTR, CLNTCONN, CLUSSDR, CLUSRCVR
BatchHeartbeat	MQXR_INIT	Tümü	SDR, SVR, CLUSSDR, CLUSRCVR
HdrCompListesi	MQXR_INIT	Tümü	Tümü
MsgCompListesi	MQXR_INIT	Tümü	Tümü
ChannelMonitoring	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, SVRCONN, CLUSSDR, CLUSRCVR
ChannelStatistics	MQXR_INIT	Tümü	SDR, SVR, RCVR, RQSTR, CLUSSDR, CLUSRCVR
SharingConversations	MQXR_INIT	Tümü	SVRCONN, CLNTCONN
PropertyControl	MQXR_INIT	Tümü	SDR, SVR, CLUSSDR, CLUSRCVR

MQCXP-Kanal çıkış değiştirgesi

MQCXP yapısı, bir Message Channel Agent (MCA), istemci-bağlantı kanalı ya da sunucu bağlantısı kanalı tarafından çağrılan her çıkış tipine geçirilir.

Bkz. MQ_CHANNEL_EXIT.

Aşağıdaki açıklamalarda "çıkışa giriş" olarak tanımlanan alanlar, çıkış kanala denetimi döndürdüğünde kanal tarafından yoksayılır. Kanal çıkış değiştirgesi öbeğinde çıkışın değiştiği giriş alanları, sonraki çağrılışında korunmaz. Giriş/çıkış alanlarında (örneğin, *ExitUserArea* alanı) yapılan değişiklikler, yalnızca o çıkış eşgörünümünün çağrılması için korunur. Bu tür değişiklikler, aynı kanalda tanımlanan farklı çıkışlar arasında ya da farklı kanallarda tanımlanan aynı çıkış arasında veri geçirmek için kullanılamaz.

İlgili başvurular

["Alanlar" sayfa 1494](#)

Bu konuda, MQCXP yapısındaki tüm alanlar listelenir ve her alan açıklanır.

["C bildirim" sayfa 1504](#)

Bu bildirim, MQCXP yapısına ilişkin C bildirimini gösterir.

["COBOL bildirim" sayfa 1505](#)

Bu bildirim, MQCXP yapısına ilişkin COBOL bildirimini gösterir.

["RPG bildirim \(ILE\)" sayfa 1506](#)

Bu bildirim, MQCXP yapısına ilişkin RPG bildirimini gösterir.

“System/390 çevirici bildirimini” sayfa 1507

Bu bildirim, MQCXP yapısına ilişkin System/390 çevirici bildirimini gösterir.

Alanlar

Bu konuda, MQCXP yapısındaki tüm alanlar listelenir ve her alan açıklanır.

StrucId (MQCHAR4)

Bu alan, yapı tanıttıcısını belirtir.

Değer şöyle olmalıdır:

MQCXP_STRUC_ID

Kanal çıkış parametresi yapısına ilişkin tanıttıcı.

C programlama dili için, değişmez MQCXP_STRUC_ID_ARRAY de tanımlanır; bu değişmezın değeri MQCXP_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi.

Bu, çıkışa bir giriş alanıdır.

Sürüm (MQLONG)

Bu alan, yapı sürümü numarasını belirtir.

Değer ortama bağlıdır:

MQCXP_VERSION_1

Version-1 kanal çıkış parametresi yapısı.

MQCXP_VERSION_2

Version-2 kanal çıkış parametresi yapısı.

MQCXP_VERSION_3

Version-3 kanal çıkış parametresi yapısı.

 Alan, başka bir yerde listelenmeyen AIX and Linux sistemlerinde bu değeri

içerir.

MQCXP_VERSION_4

Version-4 kanal çıkış değıştirtgesi yapısı.

MQCXP_VERSION_5

Version-5 kanal çıkış değıştirtgesi yapısı.

MQCXP_VERSION_6

Version-6 kanal çıkış değıştirtgesi yapısı.

MQCXP_VERSION_8

Version-8 kanal çıkış değıştirtgesi yapısı.

 Alan, z/OSiçinde bu değere sahiptir.

MQCXP_VERSION_9

Version-9 kanal çıkış değıştirtgesi yapısı.

Alan ařağıdaki ortamlarda bu değere sahiptir:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

Yalnızca yapının daha yeni sürümlerinde bulunan alanlar, alanların tanımlarında bu şekilde tanımlanır. Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQ_CXP_CURRENT_VERSION

Kanal çıkış parametresi yapısının yürürlükteki sürümü.

Değer ortama bağlıdır.

Not: MQ_CXP yapısının yeni bir sürümü kullanıma sunulduğunda, var olan kısmın düzeni değişmez. Bu nedenle çıkışın, sürüm numarasının, çıkışın kullanması gereken alanları içeren en düşük sürüme eşit ya da bu sürümden büyük olup olmadığını denetlemesi gerekir.

Bu, çıkışa bir giriş alanıdır.

ExitId (MQLONG)

Bu alan, çağrılmakta olan çıkışın tipini belirtir ve çıkış yordamının girişinde ayarlanır.

Aşağıdaki değerler kullanılabilir:

MQXT_CHANNEL_SEC_EXIT

Kanal güvenliği çıkışı.

MQXT_CHANNEL_MSG_EXIT

Kanal mesaj çıkışı.

MQXT_CHANNEL_SEND_EXIT

Kanal gönderme çıkışı.

MQXT_CHANNEL_RCV_EXIT

Kanal alma çıkışı.

MQXT_CHANNEL_MSG_RETRY_EXIT

Kanal iletisi-çıkışı yeniden deneyin.

MQXT_CHANNEL_AUTO_DEF_EXIT

Kanal otomatik tanımlama çıkışı.

z/OS üzerinde bu çıkış tipi yalnızca MQ_CHT_CLUSSDR ve MQ_CHT_CLUSRCVR tipindeki kanallar için desteklenir.

Bu, çıkışa bir giriş alanıdır.

ExitReason (MQLONG)

Bu alan, çıkışın çağrılmasının ve çıkış yordamının girişinde ayarlanmasının nedenini belirtir.

Otomatik tanımlama çıkışı tarafından kullanılmaz. Aşağıdaki değerler kullanılabilir:

MQXR_INIT

Başlatma işleminden çık.

Bu değer, çıkışın ilk kez çağrıldığını gösterir. Çıkışın, gereksinim duyduğu kaynakları (örneğin: bellek) edinmesini ve kullanıma hazırlamasını sağlar.

MQXR_TERM

Çıkış sonlandırması.

Bu değer, çıkışın sonlandırılacağını gösterir. Çıkış, başlatıldığından bu yana edindiği kaynakları serbest bırakmalıdır (örneğin: bellek).

MQXR_MSG

Bir iletiyi işleyin.

Bu değer, bir iletiyi işlemek için çıkışın çağrıldığını gösterir. Bu değer, yalnızca kanal iletisi çıkışları için oluşur.

MQXR_XMIT

Bir iletimi işleyin.

Bu değer, yalnızca kanal gönderme ve alma çıkışları için geçerlidir.

MQXR_SEC_MSG

Güvenlik iletisi alındı.

Bu değer, yalnızca kanal güvenliği çıkışları için geçerlidir.

MQXR_INIT_SEC

Güvenlik değişimini başlatın.

Bu değer, yalnızca kanal güvenliği çıkışları için geçerlidir.

Alıcının güvenlik çıkışı, MQXR_INIT ile çağrıldıktan hemen sonra, güvenlik değiş tokuşu başlatma fırsatı vermek için her zaman bu nedenle çağrılır. Fırsatı reddederseniz (MQXCC_SEND_SEC_MSG ya da MQXCC_SEND_AND_REQUEST_SEC_MSG yerine MQXCC_OK döndürülürse), gönderenin güvenlik çıkışı MQXR_INIT_SEC ile çağrılır.

Alıcının güvenlik çıkışı bir güvenlik değiş tokuşu başlatırsa (MQXCC_SEND_SEC_MSG ya da MQXCC_SEND_AND_REQUEST_SEC_MSG döndürerek), gönderenin güvenlik çıkışı hiçbir zaman MQXR_INIT_SEC ile çağrılmaz; bunun yerine, alıcının iletisini işlemek için MQXR_SEC_MSG ile çağrılır. (Her iki durumda da ilk olarak MQXR_INIT ile çağrılır.)

Güvenlik isteklerinden biri kanalın sonlandırılmasını istemezse (*ExitResponse* MQXCC_SUPPRESS_FUNCTION ya da MQXCC_CLOSE_CHANNEL olarak ayarlanarak), güvenlik değiş tokuşu, değiş tokuşu başlatan tarafta tamamlanmalıdır. Bu nedenle, MQXR_INIT_SEC ile bir güvenlik çıkışı çağrılırsa ve bir değiş tokuş başlatırsa, çıkış bir sonraki çağrılışında MQXR_SEC_MSG ile birlikte olur. Bu durum, çıkışın işlenmesine ya da işlenmemesine ilişkin bir güvenlik iletisi olup olmadığını belirler. Ortak MQXCC_SEND_SEC_MSG ya da MQXCC_SEND_AND_REQUEST_SEC_MSG döndürürse, ancak ortak MQXCC_OK döndürürse ya da ortakta güvenlik çıkışı yoksa, bir güvenlik iletisi görüntülenir. İşlenecek bir güvenlik iletisi yoksa, başlatma ucundaki güvenlik çıkışı sıfır *DataLength* ile yeniden çağrılır.

MQXR_RETRY

Bir iletiyi yeniden deneyin.

Bu değer ileti için oluşur-yalnızca yeniden deneme çıkışları.

MQXR_AUTO_CLUSSDR

Küme gönderen kanalının otomatik tanımı.

Bu değer, yalnızca kanal otomatik tanımlaması çıkışları için oluşur.

MQXR_AUTO_ALICI

Alıcı kanalının otomatik tanımı.

Bu değer, yalnızca kanal otomatik tanımlaması çıkışları için oluşur.

MQXR_AUTO_SVRCONN

Sunucu bağlantısı kanalının otomatik tanımı.

Bu değer, yalnızca kanal otomatik tanımlaması çıkışları için oluşur.

MQXR_AUTO_CLUSRCVR

Bir küme-alıcı kanalının otomatik tanımı.

Bu değer, yalnızca kanal otomatik tanımlaması çıkışları için oluşur.

MQXR_SEC_PARMS

Güvenlik parametreleri

Bu değer yalnızca güvenlik çıkışları için geçerlidir ve çıkışa bir MQCSP yapısının geçirildiğini gösterir. Daha fazla bilgi için bkz. [“MQCSP-Güvenlik değiştiricileri” sayfa 334](#)

Not:

1. Bir kanal için tanımlanmış birden fazla çıkışın varsa, MCA kullanıma hazırlandığında bunların her biri MQXR_INIT ile çağrılır. Ayrıca, MCA sonlandırıldığında bunların her biri MQXR_TERM ile çağrılır.
2. Kanal otomatik tanımlama çıkışı için, *Version* MQXCP_VERSION_4' den küçükse *ExitReason* ayarlanmaz. Bu durumda MQXR_AUTO_SVRCONN değeri örtük olarak belirtilmiştir.

Bu, çıkışa bir giriş alanıdır.

ExitResponse (MQLONG)

Bu alan, çıkıştan gelen yanıtı belirtir.

Bu alan, MCA ile iletişim kurmak için çıkış tarafından ayarlanır. Aşağıdaki değerlerden biri olmalıdır:

MQXCC_OK

Çıkış başarıyla tamamlandı.

- Kanal güvenliği çıkışı için bu değer, ileti aktarımının artık olağan bir şekilde devam edebileceğini gösterir.
- Kanal iletişi yeniden deneme çıkışı için bu değer, MCA 'nın MQCXP' deki *MsgRetryInterval* alanındaki çıkışın döndürdüğü zaman aralığını beklemesi gerektiğini gösterir ve iletiyi yeniden deneyin.

ExitResponse2 alanı ek bilgi içerebilir.

MQXCC_SUPPRESS_FUNCTION

İşlevi engelle.

- Kanal güvenlik çıkışı için bu değer, kanalın sonlandırılması gerektiğini gösterir.
- Kanal ileti çıkışı için bu değer, iletinin hedefine daha fazla ilerleyemeyeceğini gösterir. Bunun yerine, MCA bir kural dışı durum raporu iletişi oluşturur (özgün iletiyi gönderen tarafından istendiye) ve özgün arabellekte bulunan iletiyi gönderenin belirttiği MQRO_DEAD_LETTER_Q ise, gönderenin belirttiği MQRO_DISCARD_MSG ' ye yerleştirir).

Kalıcı iletiler için, gönderen MQRO_DEAD_LETTER_Q belirttiyse, ancak gitmeyen iletiler kuyruğuna koyma başarısız olursa ya da gitmeyen iletiler kuyruğu yoksa, özgün ileti iletim kuyruğuna bırakılır ve rapor iletişi oluşturulmaz. Rapor iletişi başarıyla oluşturulamazsa, özgün ileti de iletim kuyruğunda bırakılır.

İleti gönderilmeyen iletiler kuyruğundaki iletinin başındaki MQDLH yapısındaki *Feedback* alanı, iletinin neden gönderilmeyen iletiler kuyruğuna konduğunu gösterir; bu geribildirim kodu, kural dışı durum raporu iletişinin ileti tanımlayıcısında da (gönderen tarafından istendiye) kullanılır.

- Kanal iletişi yeniden deneme çıkışı için bu değer, MCA 'nın iletiyi beklemediğini ve yeniden denemediğini gösterir; bunun yerine, MCA olağan hata işlemesiyle hemen devam eder (ileti, iletiyi gönderen tarafından belirtilen, gitmeyen iletiler kuyruğuna konur ya da atılır).
- Kanal otomatik tanımlama çıkışı için MQXCC_OK ya da MQXCC_SUPPRESS_FUNCTION belirtilmelidir. Bu değerlerin hiçbiri belirtilmezse, MQXCC_SUPPRESS_FUNCTION varsayılan olarak varsayılır ve otomatik tanımlama bırakılır.

Bu yanıt, kanal gönderme ve alma çıkışları için desteklenmez.

MQXCC_SEND_SEC_MSG

Güvenlik iletişi gönder.

Bu değer yalnızca bir kanal güvenlik çıkışı tarafından ayarlanabilir. Çıkışın, iş ortağına iletilmesi gereken bir güvenlik iletişi sağladığını gösterir.

MQXCC_SEND_AND_REQUEST_SEC_MSG

Yanıt gerektiren bir güvenlik iletişi gönderin.

Bu değer yalnızca bir kanal güvenlik çıkışı tarafından ayarlanabilir. Bu gösteriyor ki ...

- Çıkışın iş ortağına iletebilecek bir güvenlik iletişi sağladığını ve
- çıkış için iş ortağından yanıt alınması gerekir. Herhangi bir yanıt alınmazsa, çıkış henüz iletişimin devam edip edemeyeceğine karar vermediği için kanal sonlandırılmalıdır.

MQXCC_SUPPRESS_EXIT

Çıkışı engelle.

- Bu değer, güvenlik çıkışı ya da otomatik tanımlama çıkışı dışındaki tüm kanal çıkışı tipleriyle belirlenebilir. Çıkış yeniden MQXR_TERM *ExitReason* ile çağrılıncaya kadar, kanal sonlanıncaya kadar, bu çıkışın herhangi bir çağrısını (kanal tanımında adı boş gibi) gizler.
- Bir ileti yeniden deneme çıkışı bu değeri döndürürse, sonraki iletilere ilişkin yeniden denemeler *MsgRetryCount* ve *MsgRetryInterval* kanal öznitelikleri tarafından normal olarak denetlenir. For the current message, the MCA performs the number of outstanding retries, at intervals given by the *MsgRetryInterval* channel attribute, but only if the reason code is one that the MCA would normally retry (see the *MsgRetryCount* field described in “MQCD-Kanal tanımlaması” sayfa 1454). Bekleyen yeniden deneme sayısı **MsgRetryCount** özniteliğinin değeridir; çıkışın yürürlükteki ileti için kaç kez MQXCC_OK döndürdüğünü gösterir; bu sayı negatifse, yürürlükteki ileti için MCA tarafından başka yeniden deneme gerçekleştirilmez.

MQXCC_CLOSE_CHANNEL

Kanalı kapat.

Bu değer, otomatik tanımlama çıkışı dışında herhangi bir kanal çıkışı tipiyle belirlenebilir.

Paylaşım etkileşimleri etkinleştirilmezse, bu değer kanalı kapatır.

Paylaşım etkileşimleri etkinleştirildiyse, bu değer etkileşimi sona erdirir. Kanaldaki tek sohbet bu ise, kanal da kapanır.

Bu alan, çıkıştan bir giriş/çıkış alanıdır.

ExitResponse2 (MQLONG)

Bu alan, çıkıştan gelen ikincil yanıtı belirtir.

Bu alan, çıkış yordamına ilişkin girişte sıfır olarak ayarlanır. IBM MQ kanal işlevlerine daha fazla bilgi sağlamak için çıkış tarafından ayarlanabilir. Otomatik tanımlama çıkışı tarafından kullanılmaz.

Çıkış aşağıdaki değerlerden birini ya da birkaçını ayarlayabilir. Birden çok değer gerekiyorsa, değerler eklenir. Geçerli olmayan birleşimler not edildi; diğer birleşimler izin verildi.

MQXR2_PUT_WITH_DEF_ACTION

Varsayılan işlemle koy.

Bu değer, alıcının kanal ileti çıkışı tarafından belirlenir. İletinin MCA 'nın varsayılan işlemiyle (MCA' nın varsayılan kullanıcı kimliği) ya da iletinin MQMD 'sindeki (ileti tanımlayıcısı) *UserIdentifier* bağlamıyla birlikte konacağını gösterir.

Çıkış çağrıldığında ayarlanan ilk değere karşılık gelen değer sıfırdır. Sabit, belgeleme amacıyla sağlanır.

MQXR2_PUT_WITH_DEF_USERID

Varsayılan kullanıcı kimliğine koyun.

Bu değer yalnızca alıcının kanal ileti çıkışı tarafından ayarlanabilir. İletinin MCA 'nın varsayılan kullanıcı kimliğiyle birlikte konacağını gösterir.

MQXR2_PUT_WITH_MSG_USERID

İletinin kullanıcı kimliğiyle birlikte koyun.

Bu değer yalnızca alıcının kanal ileti çıkışı tarafından ayarlanabilir. İletinin, iletinin MQMD (ileti tanımlayıcısı) içindeki *UserIdentifier* bağlamıyla birlikte konacağını gösterir (bu, çıkış tarafından değiştirilmiş olabilir).

MQXR2_PUT_WITH_DEF_ACTION, MQXR2_PUT_WITH_DEF_USERID ve MQXR2_PUT_WITH_MSG_USERID öğelerinden yalnızca biri ayarlanmalıdır.

MQXR2_USE_AGENT_BUFFER

Aracı arabelleğini kullan.

Bu değer, iletilecek verilerin *ExitBufferAddr* değil, *AgentBuffer* içinde olduğunu gösterir.

Çıkış çağrıldığında ayarlanan ilk değere karşılık gelen değer sıfırdır. Sabit, belgeleme amacıyla sağlanır.

MQXR2_USE_EXIT_BUFFER

Çıkış arabelleğini kullan.

Bu değer, iletilecek verilerin *AgentBuffer* değil, *ExitBufferAdd* içinde olduğunu gösterir. Yalnızca bir `MQXR2_USE_AGENT_BUFFER` ve `MQXR2_USE_EXIT_BUFFER` ayarlanmalıdır.

MQXR2_DEFAULT_CONTINUATION

Varsayılan devam.

Zincirdeki bir sonraki çıkışla devam etmek, çağrılan son çıkıştan gelen yanıtla bağlıdır:

- `MQXCC_SUPPRESS_FUNCTION` ya da `MQXC_CLOSE_CHANNEL` döndürülürse, zincirde başka çıkış çağrılmaz.
- Ters durumda, zincirdeki bir sonraki çıkış çağrılır.

MQXR2_CONTINUE_CHAIN

Sonraki çıkışla devam edin.

MQXR2_SUPPRESS_CHAIN

Zincirdeki kalan çıkışları atla.

Bu, çıkışa bir giriş/çıkış alanıdır.

Geribildirim (MQLONG)

Bu alan geribildirim kodunu belirtir.

Bu alan, çıkış yordamına giriş için `MQFB_NONE` olarak ayarlanır.

Bir kanal ileti çıkışı *ExitResponse* alanını `MQXCC_SUPPRESS_FUNCTION` olarak ayarlarsa, *Feedback* alanı iletinin neden teslim edilmeyen ileti kuyruğuna konduğunu tanımlayan geribildirim kodunu belirtir ve istendiyse bir kural dışı durum raporu göndermek için de kullanılır. Bu durumda, *Feedback* alanı `MQFB_NONE` ise, aşağıdaki geribildirim kodu kullanılır:

MQFB_STOPPED_BY_MSG_EXIT

İleti, kanal ileti çıkışı tarafından durduruldu.

Bu alanda kanal güvenliği, gönderme, alma ve ileti yeniden deneme çıkışları tarafından döndürülen değer `MCA` tarafından kullanılmaz.

ExitResponse `MQXCC_OK` ise, otomatik tanımlama çıkışları tarafından bu alanda döndürülen değer kullanılmaz; ters durumda, olay iletiindeki *AuxErrorDataInt1* parametresi için kullanılır.

Bu, çıkıştan bir giriş/çıkış alanıdır.

MaxSegmentUzunluğu (MQLONG)

Bu alan, tek bir iletimde gönderilebilecek bayt cinsinden uzunluk üst sınırını belirler.

Otomatik tanımlama çıkışı tarafından kullanılmaz. Bu çıkış, bir iletim kesiminin büyüklüğünü *MaxSegmentLength* değerinden büyük bir değere çıkarmamasını sağladığından, kanal gönderme çıkışıyla ilgilenir. Uzunluk, çıkışın değiştirmemesi gereken ilk 8 baytı içerir. Kanal başlatıldığında IBM MQ kanal işlevleri arasında değer karşılaştırılır. Kesim uzunluklarına ilişkin ek bilgi için [Kanalı-çıkış programlarının yazılması](#) başlıklı konuya bakın.

ExitReason `MQXR_INIT` ise, bu alandaki değer anlamlı değildir.

Bu, çıkışa bir giriş alanıdır.

ExitUserAlanı (MQBYTE16)

Bu alan, çıkış kullanıcı alanını belirtir-çıkışın kullanabileceği bir alan.

Çıkışın ilk çağrılmasından (`MQXR_INIT` olarak ayarlı bir *ExitReason* içeren) önce ikili sifra başlatılır ve çıkışta bu alanda yapılan değişiklikler, çıkışın çağrılmasıyla korunur.

Aşağıdaki değer tanımlandı:

MQXUA_NONE

Kullanıcı bilgisi yok.

Değer, alanın uzunluğu için ikili sıfırdır.

C programlama dili için, MQXUA_NONE_ARRAY değişmezi de tanımlanır; bu değişmezin değeri MQXUA_NONE ile aynı, ancak dizgi yerine bir karakter dizisi.

Bu alanın uzunluğu MQ_EXIT_USER_AREA_LENGTH tarafından verilir. Bu, çıkışa bir giriş/çıkış alanıdır.

ExitData (MQCHAR32)

Bu alan çıkış verilerini belirtir.

Bu alan, IBM MQ kanal işlevlerinin kanal tanımlamasından aldığı bilgilere ulaşmak için çıkış yordamının girişinde ayarlanır. Böyle bir bilgi yoksa, bu alan tamamen boşluktur.

Bu alanın uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir.

Bu, çıkışa bir giriş alanıdır.

Version, MQCXP_VERSION_2' den küçükse, bu yapıdaki aşağıdaki alanlar yoktur.

MsgRetrySayısı (MQLONG)

Bu alan, iletinin kaç kez yeniden denendiğini belirtir.

Belirli bir ileti için ilk kez çıkış çağrıldığında, bu alan sıfır değerine sahiptir (henüz yeniden deneme yapılmamıştır). Bu iletiye ilişkin çıkışın sonraki her çağrılışında, değer MCA tarafından bir artırılır.

Bu, çıkışa bir giriş alanıdır. *ExitReason* MQXR_INIT ise, bu alandaki değer anlamlı değildir. *Version*, MQCXP_VERSION_2' den küçükse alan yoktur.

MsgRetryAralığı (MQLONG)

Bu alan, koyma işleminin yeniden denenmesinden sonraki milisaniye cinsinden aralık alt sınırını belirtir.

Belirli bir ileti için çıkış ilk kez çağrıldığında, bu alan *MsgRetryInterval* kanal özneliğinin değerini içerir. Çıkış değeri değiştirmeden bırakabilir ya da milisaniye cinsinden farklı bir zaman aralığı belirtecek şekilde değiştirebilir. Çıkış *ExitResponse* içinde MQXCC_OK değerini döndürürse, MCA, MQOPEN ya da MQPUT işlemini yeniden denemeden önce en az bu zaman aralığını bekler. Belirtilen zaman aralığı sıfır ya da daha büyük olmalıdır.

Bu ileti için çıkışın çağrıldığı ikinci ve sonraki zamanlarda, bu alan çıkışın önceki çağrısının döndürdüğü değeri içerir.

MsgRetryInterval alanında döndürülen değer sıfırdan küçük ya da 999 999 999 'dan büyükse ve *ExitResponse* MQXCC_OK ise, MCA MQCXP 'deki *MsgRetryInterval* alanını yoksayar ve bunun yerine *MsgRetryInterval* kanal özneliği tarafından belirtilen aralığı bekler.

Bu, çıkışa bir giriş/çıkış alanıdır. *ExitReason* MQXR_INIT ise, bu alandaki değer anlamlı değildir. *Version*, MQCXP_VERSION_2' den küçükse alan yoktur.

MsgRetryNedeni (MQLONG)

Bu alan, iletiyi koymak için önceki girişimin neden kodunu belirtir.

Bu alan, iletiyi koymak için önceki girişimin neden kodudur; MQRC_* değerlerinden biridir.

Bu, çıkışa bir giriş alanıdır. *ExitReason* MQXR_INIT ise, bu alandaki değer anlamlı değildir. *Version*, MQCXP_VERSION_2' den küçükse alan yoktur.

Version, MQCXP_VERSION_3' den küçükse, bu yapıdaki aşağıdaki alanlar yoktur.

HeaderLength (MQLONG)

Bu alan, üstbilgi bilgilerinin uzunluğunu belirtir.

Bu alan yalnızca bir ileti çıkışı ve ileti yeniden deneme çıkışı için geçerlidir. Değer, ileti verilerinin başlangıcındaki yöneltme üstbilgisi yapılarının uzunluğudur; bunlar MQXQH yapısı, MQMDE (ileti tanımı uzantısı üstbilgisi) ve (dağıtım listesi iletilisi için) MQXQH yapısını izleyen MQOR ve MQPMR kayıtlarının MQDH yapısı ve dizileridir.

İleti çıkışı bu üstbilgi bilgilerini inceleyebilir ve gerekirse değiştirebilir, ancak çıkışın döndürdüğü verilerin doğru biçimde olması gerekir. Çıkış, örneğin, alıcı uçta ileti çıkışı, dengeleme değişiklikleri yapsa da, üstbilgi verilerini gönderme ucunda şifrelememeli ya da sıkıştırmamalıdır.

İleti çıkışı üstbilgi bilgilerini, uzunluğunu değiştirecek şekilde değiştirirse (örneğin, bir dağıtım listesi iletisine başka bir hedef ekleyerek), *HeaderLength* değerini döndürmeden önce buna uygun olarak değiştirmelidir.

Bu, çıkışa bir giriş/çıkış alanıdır. *ExitReason* MQXR_INIT ise, bu alandaki değer anlamlı değildir. *Version* , MQCXP_VERSION_3' den küçükse alan yoktur.

PartnerName (MQCHAR48)

Bu alan, iş ortağının adını belirtir.

İş ortağının adı:

- SVRCONN kanalları için, istemcide oturum açmış kullanıcı kimliğidir.
- Diğer tüm kanal tipleri için bu, iş ortağının kuyruk yöneticisi adıdır.

Çıkış kullanıma hazırlandığında, kuyruk yöneticisi ilk anlaşma gerçekleşinceye kadar iş ortağının adını bilmediği için bu alan boştur.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCXP_VERSION_3' den küçükse alan yoktur.

FAPLevel (MQLONG)

Anlaşmalı Biçimler ve Protokoller düzeyi.

Bu, çıkışa bir giriş alanıdır. Bu alanda yapılan değişiklikler yalnızca IBM hizmetinin yönlendirmesi altında yapılmalıdır. *Version* , MQCXP_VERSION_3' den küçükse alan yoktur.

CapabilityFlags (MQLONG)

Yetenek işaretini MQCF_NONE ya da MQCF_DIST_LIST olarak ayarlayabilirsiniz.

Aşağıdaki yetenek işaretlerinden birini ayarlayabilirsiniz:

MQCF_NONE

Bayrak yok.

MQCF_DIST_LIST

Dağıtım listeleri desteklenir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCXP_VERSION_3' den küçükse alan yoktur.

ExitNumber (MQLONG)

Bu alan, çıkışın sıra numarasını belirtir.

ExitId içinde tanımlanan tip içindeki çıkışın sıra numarası. Örneğin, çağrılmakta olan çıkış tanımlanan üçüncü ileti çıkışıysa, bu alan 3 değerini içerir. Çıkış tipi, çıkış listesinin tanımlanamadığı bir tipe (örneğin, bir güvenlik çıkışı), bu alan 1 değerine sahiptir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCXP_VERSION_3' den küçükse alan yoktur.

Version , MQCXP_VERSION_5' den küçükse, bu yapıdaki aşağıdaki alanlar yoktur.

ExitSpace (MQLONG)

Bu alan, çıkışın kullanması için ayrılan iletim arabelleğindeki bayt sayısını belirler.

Bu alan yalnızca bir gönderme çıkışı için geçerlidir. IBM MQ kanal işlevlerinin, çıkışın kullanması için iletim arabelleğinde ayırdığı bayt cinsinden alan miktarını belirler. Bu alan, çıkışın diğer uçta tamamlayıcı bir alma çıkışı tarafından kullanılmak üzere iletim arabelleğine az miktarda veri (genellikle birkaç yüz bayt geçmez) eklemesini sağlar. Gönderme çıkışı tarafından eklenen veriler alma çıkışı tarafından kaldırılmalıdır.

Değer z/OS üzerinde her zaman sıfırdır.

Not: Bu olanak, başarımı düşürebileceğinden ya da kanalın çalışmasını engelleyebileceğinden, büyük miktarlarda veri göndermek için kullanılmamalıdır.

ExitSpace değeri belirlenerek, çıkışın, iletim arabelleğinde her zaman, çıkışın kullanılabilmesi için en az bu sayıda bayt bulunacağı garanti edilir. Ancak, çıkış ayrılan miktardan daha az ya da iletim arabelleğinde

yer varsa ayrılan miktardan daha fazla kullanabilir. Arabellekteki çıkış alanı, var olan verileri izleyerek sağlanır.

ExitSpace , çıkış tarafından yalnızca *ExitReason* MQXR_INITdeğerine sahip olduğunda ayarlanabilir; diğer tüm durumlarda, çıkışın döndürdüğü değer yoksayıdır. Çıkışa girişte, MQXR_INIT çağrısı için *ExitSpace* sıfırdır ve diğer durumlarda MQXR_INIT çağrısıyla döndürülen değerdir.

MQXR_INIT çağrısıyla döndürülen değer negatifse ya da zincirdeki tüm gönderme çıkışları için istenen çıkış alanını ayırdıktan sonra ileti verileri için iletim arabelleğinde 1024 byte 'tan az kullanılabilir varsa, MCA bir hata iletisi verir ve kanalı kapatır. Benzer şekilde, veri aktarımı sırasında, gönderme çıkış zincirindeki çıkışlar, ileti verileri için iletim arabelleğinde 1024 bayttan daha az yer kalacak şekilde ayrıldığında, MCA bir hata iletisi verir ve kanalı kapatır. 1024 sınırı, kanalın denetim ve yönetim akışlarının, akışların bölümlenmesine gerek kalmadan, gönderme çıkışları zinciri tarafından işlenmesini sağlar.

Bu, *ExitReason* MQXR_INIT ise çıkışa bir giriş/çıkış alanıdır ve diğer tüm durumlarda bir giriş alanıdır. *Version* , MQCXP_VERSION_5' den küçükse alan yoktur.

SSLCertUserKimliği (MQCHAR12)

Bu alan, uzak sertifikayla ilişkilendirilmiş UserId değerini belirtir.

z/OS dışındaki tüm platformlarda boş

Bu, çıkışa bir giriş alanıdır. *Version* , MQCXP_VERSION_6' dan küçükse alan yoktur.

SSLRemCertIssNameUzunluğu (MQLONG)

Bu alan, SSLCertRemoteIssuerNamePtr tarafından gösterilen uzak sertifikayı veren kişinin tam Ayırt Edici Adının bayt cinsinden uzunluğunu belirtir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCXP_VERSION_6' dan küçükse alan yoktur. TLS kanalı değilse, değer sıfırdır.

SSLRemCertIssNamePtr (PMQVOID)

Bu alan, uzak sertifikayı veren kişinin tam ayırt edici adının adresini belirler.

Değeri, TLS kanalı değilse, boş değerli işaretçidir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCXP_VERSION_6' dan küçükse alan yoktur.

Not: Konu Ayırt Edici Adı ve Sertifika Veren Ayırt Edici Adı belirlenirken kanal güvenliği çıkışlarının davranışı IBM WebSphere MQ 7.1olarak değiştirilir. Daha fazla bilgi için bkz. [Kanal güvenliği çıkış programları](#).

SecurityParms (PMQCSP)

Bu alan, kimlik doğrulama kimlik bilgilerini belirtmek için kullanılan MQCSP yapısının adresini belirtir.

Bu alanın ilk değeri boş işaretçidir.

Bu, çıkışa bir giriş/çıkış alanıdır. *Version* , MQCXP_VERSION_6' dan küçükse alan yoktur.

Bu alandaki çıkışın döndürdüğü değer, MQXR_TERM komutuna kadar IBM MQ tarafından kullanılabilir olmalıdır.

CurHdrSıkıştırma (MQLONG)

Bu alan, üstbilgi verilerini sıkıştırmak için kullanılmakta olan tekniği belirtir.

Aşağıdakilerden birine ayarlanır:

MQCOMPRESS_NONE

Üstbilgi veri sıkıştırması gerçekleştirilmez.

MQCOMPRESS_SYSTEM

Üstbilgi veri sıkıştırması gerçekleştirilir.

Değer, bir gönderen kanalın ileti çıkışıyla, MQCD ' nin HdrCompList alanından erişilen karşılaştırılan desteklenen değerlerden birine değiştirilebilir. Bu, iletinin içeriğine dayalı olarak her ileti için seçilecek üstbilgi verilerini sıkıştırmak için kullanılan tekniği etkinleştirir. Değiştirilen değer yalnızca yürürlükteki

ileti için kullanılır. Öznitelik desteklenmeyen bir değere değiştirilirse kanal sona erer. Değer, gönderen bir kanalın ileti çıkışının dışında değiştirilirse yoksayılır.

Bu, çıkışa bir giriş/çıkış alanıdır. *Version* , MQCXP_VERSION_6' dan küçükse alan yoktur.

CurMsgSıkıştırma (MQLONG)

Bu alan, ileti verilerini sıkıştırmak için kullanılmakta olan tekniği belirtir.

Aşağıdakilerden birine ayarlanır:

MQCOMPRESS_NONE

Üstbilgi veri sıkıştırması gerçekleştirilmez.

MQCOMPRESS_RLE

İleti veri sıkıştırması, çalıştırma uzunluğu kodlaması kullanılarak gerçekleştirilir.

MQCOMPRESS_ZLIBFAST

İleti veri sıkıştırması, zlib sıkıştırma tekniği kullanılarak gerçekleştirilir. Hızlı sıkıştırma süresi tercih edilir.

MQCOMPRESS_ZLIBHIGH

İleti veri sıkıştırması, zlib sıkıştırma tekniği kullanılarak gerçekleştirilir. Yüksek düzeyde sıkıştırma tercih edilir.

Değer, bir gönderen kanalın ileti çıkışıyla, MQCD ' nin MsgCompList alanından erişilen kararlaştırılan desteklenen değerlerden birine değiştirilebilir. Bu, iletinin içeriğine dayalı olarak her ileti için karar verilecek ileti verilerini sıkıştırmak için kullanılan tekniği etkinleştirir. Değiştirilen değer yalnızca yürürlükteki ileti için kullanılır. Öznitelik desteklenmeyen bir değere değiştirilirse kanal sona erer. Değer, gönderen bir kanalın ileti çıkışının dışında değiştirilirse yoksayılır.

Bu, çıkışa bir giriş/çıkış alanıdır. *Version* , MQCXP_VERSION_6' dan küçükse alan yoktur.

Hconn (MQHCONN)

Bu alan, çıkış içinde MQI çağrılarını yapması gerekiyorsa çıkışın kullanacağı bağlantı tanıtıcısını belirtir.

Bu alan, istemci bağlantısı kanallarında çalışan çıkışlarla ilgili değildir; burada MQHC_UNUSABLE_HCONN (-1) değerini içerir.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCXP_VERSION_7' den küçükse alan yoktur.

SharingConversations (MQBOOL)

Bu alan, etkileşimin şu anda bu kanal örneğinde çalışabilecek tek etkileşim olup olmadığını ya da şu anda bu kanal örneğinde birden çok etkileşimin çalışıp çalışamayacağını belirtir.

Çıkış programının, aynı anda çalışan başka bir çıkış programı tarafından değiştirilme riskine tabi olup olmadığını da gösterir.

Bu alan yalnızca istemci bağlantısı ya da sunucu bağlantısı kanallarında çalışan çıkış programları için geçerlidir.

Aşağıdakilerden birine ayarlanır:

YANLIŞ

Çıkış eşgörünümü, şu anda bu kanal yönetim ortamında çalışabilecek tek çıkış eşgörünümüdür.

Bu, çıkışın MQCD alanlarını diğer kanal yönetim ortamlarında çalışan diğer çıkışlardan çekişmeden güvenli bir şekilde güncellemesini sağlar. MQCD alanlarında yapılan değişikliklerin kanal tarafından gerçekleştirilip gerçekleştirilmeyeceğini, “Kanal çıkışında MQCD alanlarının değiştirilmesi” sayfa 1490’indeki MQCD alanları çizelgesi belirler.

DOĞRU

Çıkış eşgörünümü, şu anda bu kanal yönetim ortamında çalışabilecek tek çıkış eşgörünümü değil. MQCD ' de yapılan değişiklikler, MQXR_INIT dışındaki Çıkış Nedenleri için “Kanal çıkışında MQCD alanlarının değiştirilmesi” sayfa 1490 içindeki MQCD alanları çizelgesinde listelenen değişiklikler dışında, kanal tarafından gerçekleştirilmez. Bu çıkış MQCD alanlarını güncellerse, bu kanal yönetim ortamında çalışan çıkışlar arasında diziselleştirme sağlayarak, aynı anda başka etkileşimler üzerinde çalışan diğer çıkışlardan çekişme olmamasına dikkat edin.

Bu, çıkışa bir giriş alanıdır. *Version* , MQCXP_VERSION_7' den küçükse alan yoktur.

MCAUserSource (MQLONG)

Bu alan, sağlanan MCA kullanıcı kimliğinin kaynağını belirtir.

Aşağıdaki değerlerden birini içerebilir:

MQUSRC_MAP

Kullanıcı kimliği, MCAUSER özniteliğinde belirtilir.

MQUSRC_CHANNEL

Kullanıcı kimliği, gelen iş ortağından ya da kanal nesnesinde tanımlanan MCAUSER alanında belirtilir.

Bu, çıkışa bir giriş alanıdır. Sürüm MQCXP_VERSION_8' den küçükse alan yoktur.

pEntryPuanlar (PMQIEP)

Bu alan, MQI ya da DCI çağrısına ilişkin arabirim giriş noktasının adresini belirtir.

Sürüm MQCXP_VERSION_8' den küçükse bu alan yoktur.

RemoteProduct (MQCHAR4)

Bu alan, uzak ürün adını belirtir.

Bu alan, DISPLAY CHSATU' un **RPRODUCT** alanında görüntülediği gibi, istemcinin uzak ürününü tanımlar; örneğin, C ya da Java.

Sürüm MQCXP_VERSION_9' dan küçükse bu alan yoktur.

RemoteVersion (MQCHAR8)

Bu alan, uzak sürümün adını belirtir.

Bu alan, DISPLAY CHSTATUS alanının **RVERSION** alanında görüntülediği şekilde istemci kitaplıklarının sürümünü tanımlar.

Sürüm MQCXP_VERSION_9' dan küçükse bu alan yoktur.

C bildirimi

Bu bildirim, MQCXP yapısına ilişkin C bildirimini gösterir.

```
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
    retried */
    MQLONG     MsgRetryInterval; /* Minimum interval in milliseconds after
    which the put operation should be
    retried */
    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 */
};

```

COBOL bildirim

Bu bildirim, MQCXP yapısına ilişkin COBOL bildirimini gösterir.

```

** 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).

```

RPG bildirim (ILE)

Bu bildirim, MQCXP yapısına ilişkin RPG bildirimini gösterir.

```

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

System/390 çevirici bildirimi

Bu bildirim, MQCXP yapısına ilişkin System/390 çevirici bildirimini gösterir.

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_EXITSPLACE	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-Çıkış bekleme tanımlayıcısı

MQXWD yapısı, MQXWAIT çağrısında bir giriş/çıkış deęiřtirgesidir.

Bu yapı yalnızca z/OSüzerinde desteklenir.

İlgili başvurular

[“Alanlar” sayfa 1507](#)

Bu konuda, MQXWD yapısındaki tüm alanlar listelenir ve her alan açıklanır.

[“C bildirimi” sayfa 1508](#)

Bu bildirim, MQXWD yapısına ilişkin C bildirimini gösterir.

[“System/390 çevirici bildirimi” sayfa 1508](#)

Bu bildirim, MQXWD yapısına ilişkin System/390 çevirici bildirimini gösterir.

Alanlar

Bu konuda, MQXWD yapısındaki tüm alanlar listelenir ve her alan açıklanır.

StrucId (MQCHAR4)

Bu alan, yapı tanıttıcısını belirtir.

Değer şöyle olmalıdır:

MQXWD_STRUC_ID

Çıkış bekleme tanımlayıcısı yapısına ilişkin tanıttıcı.

C programlama dili için, değişmez MQXWD_STRUC_ID_ARRAY da tanımlanır; bu değişmezin değeri MQXWD_STRUC_ID ile aynı, ancak bir dizgi yerine bir karakter dizisi.

Bu alanın ilk değeri MQXWD_STRUC_ID 'dir.

Sürüm (MQLONG)

Bu alan, yapı sürümü numarasını belirtir.

Değer şöyle olmalıdır:

MQXWD_VERSION_1

Çıkış bekleme tanımlayıcısı yapısına ilişkin sürüm numarası.

Bu alanın ilk değeri MQXWD_VERSION_1' dir.

Reserved1 (MQLONG)

Bu alan ayrılmış. Değeri sıfır olmalıdır.

Bu bir giriş alanıdır.

Reserved2 (MQLONG)

Bu alan ayrılmış. Değeri sıfır olmalıdır.

Bu bir giriş alanıdır.

Reserved3 (MQLONG)

Bu alan ayrılmış. Değeri sıfır olmalıdır.

Bu bir giriş alanıdır.

ECB (MQLONG)

Bu alan, beklenecek olay denetim bloğunu belirtir.

Bu alan, beklenecek olay denetim öbeğidir (ECB). MQXWAIT çağrısı yayınlanmadan önce sıfır olarak ayarlanmalıdır; başarıyla tamamlandığında posta kodunu içerir.

Bu alan bir giriş/çıkış alanıdır.

C bildirim

Bu bildirim, MQXWD yapısına ilişkin C bildirimini gösterir.

```
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 */
};
```

System/390 çevirici bildirim

Bu bildirim, MQXWD yapısına ilişkin System/390 çevirici bildirimini gösterir.

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)

Küme iş yükü çıkış çağrısı ve veri yapıları

Bu bölümde, küme iş yükü çıkışı ve veri yapılarına ilişkin başvuru bilgileri sağlanır. Bu, genel kullanım programlama arabirimi bilgisidir.


Küme iş yükü çıkışlarını aşağıdaki programlama dillerinde yazabilirsiniz:

- C
- System/390 çevirici (IBM MQ for z/OS)

Arama aşağıdaki kısımda açıklanmıştır:

- [“MQ_CLUSTER_WORKLOAD_EXIT -Arama açıklaması” sayfa 1510](#)

Çıkış tarafından kullanılan yapı veri tipleri aşağıda açıklanmıştır:

- [“MQXCLWLN -Küme iş yükü kayıtlarında gezinme” sayfa 1511](#)
- [“MQWXP -Küme iş yükü çıkış parametresi yapısı” sayfa 1515](#)
- [“MQWDR-Küme iş yükü hedef kayıt yapısı” sayfa 1523](#)
- [“MQWQR -Küme iş yükü kuyruğu kayıt yapısı” sayfa 1527](#)
- [“MQWCR -Küme iş yükü kümesi kayıt yapısı” sayfa 1532](#)
-  [z/OS üzerinde CLUSTER komutlarının zamanuyumsuz davranışı](#)

Bu bölüm boyunca, kuyruk yöneticisi öznitelikleri ve kuyruk öznitelikleri tam olarak gösterilir. MQSC komutlarında kullanılan eşdeğer adlar aşağıda gösterilmiştir. MQSC komutlarının ayrıntıları için bkz. [MQSC komutları](#).

Çizelge 824. Kuyruk yöneticisi öznitelikleri	
Tam ad	MQSC ' de kullanılan ad
<i>ClusterWorkloadData</i>	CLWLDATA
<i>ClusterWorkloadExit</i>	CLWLEXIT
<i>ClusterWorkloadLength</i>	CLWLEN

Çizelge 825. Kuyruk öznitelikleri	
Tam ad	MQSC ' de kullanılan ad
<i>DefBind</i>	DEFBIND
<i>DefPersistence</i>	DEFPSIST
<i>DefPriority</i>	AYRILMIŞ
<i>InhibitPut</i>	PUT
<i>QDesc</i>	TANIMLAMA

İlgili görevler

[Küme iş yükü çıkışlarının yazılması ve derlenmesi](#)

MQ_CLUSTER_WORKLOAD_EXIT -Arama açıklaması

Bir iletiyi kullanılabilir bir kuyruk yöneticisine yöneltmek için kuyruk yöneticisi tarafından küme iş yükü çıkışı çağrılır.

Not: Kuyruk yöneticisi tarafından MQ_CLUSTER_WORKLOAD_EXIT adlı bir giriş noktası sağlanmadı. Bunun yerine, küme iş yükü çıkışının adı ClusterWorkloadExit kuyruk yöneticisi özneteliğiyle tanımlanır.

MQ_CLUSTER_WORKLOAD_EXIT çıkışı tüm platformlarda desteklenir.

Sözdizimi

```
MQ_CLUSTER_WORKLOAD_EXIT (ExitParms)
```

İlgili başvurular

MQXCLWLN -Küme iş yükü kayıtlarında gezinme

MQXCLWLN çağrısı, küme önbelleğinde saklanan MQWDR, MQWQRve MQWCR kayıtları zincirlerinde dolaşmak için kullanılır.

MQWXP -Küme iş yükü çıkış parametresi yapısı

Aşağıdaki çizelge, MQWXP -Küme iş yükü çıkış parametresi yapısındaki alanları özetler.

MQWDR-Küme iş yükü hedef kayıt yapısı

Aşağıdaki çizelge, MQWDR -Küme iş yükü hedef kayıt yapısındaki alanları özetler.

MQWQR -Küme iş yükü kuyruğu kayıt yapısı

Aşağıdaki çizelge, MQWQR -Küme iş yükü kuyruğu kayıt yapısındaki alanları özetler.

MQWCR -Küme iş yükü kümesi kayıt yapısı

Aşağıdaki çizelge, MQWCR kümesi iş yükü kayıt yapısındaki alanları özetler.

MQ_CLUSTER_WORKLOAD_EXIT ile ilgili parametreler

MQ_CLUSTER_WORKLOAD_EXIT çağrısında parametrelerin açıklaması.

ExitParms (MQWXP) -giriş/çıkış

Parametre bloğundan çık.

- Çıkış, iş yükünün nasıl yönetileceğini belirtmek için MQWXP ' deki bilgileri ayarlar.

İlgili başvurular

Kullanım notları

Küme iş yükü çıkışı tarafından gerçekleştirilen işlev, çıkışın sağlayıcısı tarafından tanımlanır. Ancak, çıkışın MQWXPilişkili denetim öbeğinde tanımlanan kurallara uyması gerekir.

MQ_CLUSTER_WORKLOAD_EXIT için dil çağrıları

MQ_CLUSTER_WORKLOAD_EXIT , C ve High Level Assembler olmak üzere iki dili destekler.

Kullanım notları

Küme iş yükü çıkışı tarafından gerçekleştirilen işlev, çıkışın sağlayıcısı tarafından tanımlanır. Ancak, çıkışın MQWXPilişkili denetim öbeğinde tanımlanan kurallara uyması gerekir.

Kuyruk yöneticisi tarafından MQ_CLUSTER_WORKLOAD_EXIT adlı bir giriş noktası sağlanmadı. Ancak, C programlama dilinde MQ_CLUSTER_WORKLOAD_EXIT adı için bir typedef sağlanır. Değiştiregelerin doğru olduğundan emin olmak için, kullanıcı tarafından yazılan çıkışı bildirmek için typedef parametresini kullanın.

İlgili başvurular

MQ_CLUSTER_WORKLOAD_EXIT ile ilgili parametreler

MQ_CLUSTER_WORKLOAD_EXIT çağrısında parametrelerin açıklaması.

MQ_CLUSTER_WORKLOAD_EXIT için dil çağrıları

MQ_CLUSTER_WORKLOAD_EXIT , C ve High Level Assembler olmak üzere iki dili destekler.

MQ_CLUSTER_WORKLOAD_EXIT için dil çağruları

MQ_CLUSTER_WORKLOAD_EXIT , C ve High Level Assembler olmak üzere iki dili destekler.

C çağrısı

```
MQ_CLUSTER_WORKLOAD_EXIT (&ExitParms);
```

MQ_CLUSTER_WORKLOAD_EXIT yerine, küme iş yükü çıkış işlevinizin adını koyun.

MQ_CLUSTER_WORKLOAD_EXIT deęiřtirgelerini ařaęıdaki gibi bildirin:

```
MQWXP ExitParms; /* Exit parameter block */
```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL EXITNAME, (EXITPARMS)
```

Deęiřtirgeleri ařaęıdaki gibi bildirin:

```
EXITPARMS      CMQWXP      Exit parameter block
```

İlgili bařvurular

[MQ_CLUSTER_WORKLOAD_EXIT ile ilgili parametreler](#)

MQ_CLUSTER_WORKLOAD_EXIT çağrısında parametrelerin açıklaması.

[Kullanım notları](#)

Küme iş yükü çıkışı tarafından gerçekleştirilen işlev, çıkışın sağlayıcısı tarafından tanımlanır. Ancak, çıkışın MQWXPiliřkili denetim öbeęinde tanımlanan kurallara uyması gerekir.



MQXCLWLN -Küme iş yükü kayıtlarında gezinme

MQXCLWLN çağrısı, küme önbelleęinde saklanan MQWDR, MQWQR ve MQWCR kayıtları zincirlerinde dolařmak için kullanılır.

Küme önbelleęi, kümeyle ilgili bilgileri depolamak için kullanılan ana saklama alanıdır.

Küme önbelleęi statikse, sabit bir boyutu vardır. Dinamik olarak ayarlarsanız, küme önbelleęi gerektięi şekilde genişleyebilir.

Bir sistem deęiřtirgesini ya da makroyu kullanarak küme önbelleęi tipini STATIC ya da DYNAMIC olarak ayarlayın.

-  Çoklu platformlar üzerinde ClusterCacheType sistem parametresini kullanın.
-  z/OS üzerinde CSQ6SYSP makrosunda CLCACHE deęiřtirgesini kullanın.

Sözdizimi

```
MQXCLWLN (ExitParms, CurrentRecord, NextOffset, NextRecord, Compcode, Reason)
```

İlgili bařvurular

[MQ_CLUSTER_WORKLOAD_EXIT -Arama açıklaması](#)

Bir iletiyi kullanılabilir bir kuyruk yöneticisine yöneltmek için kuyruk yöneticisi tarafından küme iş yükü çıkışı çağrılır.

[MQWXP -Küme iş yükü çıkış parametresi yapısı](#)

Ařaęıdaki çizelge, MQWXP -Küme iş yükü çıkış parametresi yapısındaki alanları özetler.

MQWDR-Küme iş yükü hedef kayıt yapısı

Aşağıdaki çizelge, MQWDR -Küme iş yükü hedef kayıt yapısındaki alanları özetler.

MQWQR -Küme iş yükü kuyruğu kayıt yapısı

Aşağıdaki çizelge, MQWQR -Küme iş yükü kuyruğu kayıt yapısındaki alanları özetler.

MQWCR -Küme iş yükü kümesi kayıt yapısı

Aşağıdaki çizelge, MQWCR kümesi iş yükü kayıt yapısındaki alanları özetler.

MQXCLWLN için parametreler-Küme iş yükü kayıtlarında gezinme

MQXCLWLN çağrısında parametrelerin açıklaması.

ExitParms (MQWXP) -giriş/çıkış

Parametre bloğundan çık.

Bu yapı, çıkışın çağrılmasıyla ilgili bilgileri içerir. Çıkış, iş yükünün nasıl yönetileceğini belirtmek için bu yapıdaki bilgileri ayarlar.

CurrentRecord (MQPTR) -giriş

Geçerli kaydın adresi.

Bu yapı, çıkış tarafından incelenmekte olan kaydın adresiyle ilgili bilgileri içerir. Kayıt aşağıdaki tiplerden biri olmalıdır:

- Küme iş yükü hedef kaydı (MQWDR)
- Küme iş yükü kuyruğu kaydı (MQWQR)
- Küme iş yükü küme kaydı (MQWCR)

NextOffset (MQLONG) -giriş

Sonraki kaydın görelî konumu.

Bu yapı, sonraki kayıt ya da yapının görelî konumuyla ilgili bilgileri içerir. *NextOffset* , geçerli kayıttaki uygun görelî konum alanının değeridir ve aşağıdaki alanlardan biri olmalıdır:

- MQWDR içindeki ChannelDefGörelî Konum alanı
- ClusterRecOffset alanı- MQWDR
- ClusterRecOffset alanı- MQWQR
- ClusterRecOffset alanı- MQWCR

NextRecord (MQPTR) -çıkış

Sonraki kayıt ya da yapının adresi.

Bu yapı, sonraki kayıt ya da yapının adresiyle ilgili bilgileri içerir. *CurrentRecord* bir MQWDR' nin adresiyse ve *NextOffset* ChannelDefGörelî Konumu alanının değeriyse, *NextRecord* kanal tanımlaması yapısının (MQCD) adresidir.

Sonraki kayıt ya da yapı yoksa, kuyruk yöneticisi *NextRecord* ögesini boş değerli gösterge olarak ayarlar ve çağrı tamamlanma kodunu MQCC_WARNING ve neden kodunu MQRC_NO_RECORD_AVAILABLEdöndürür.

CompCode (MQLONG) -çıkış

Tamamlanma kodu.

Tamamlanma kodu aşağıdaki değerlerden birini içerir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_WARNING

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG) -çıkış

Neden kodu niteleyici CompCode

CompCode MQCC_OKise:

MQRC_NONE
(0, X'0000')

Rapor etmek için bir sebep yok.

CompCode MQCC_WARNINGise:

MQRC_NO_RECORD_AVAILABLE
(2359, X'0937')

Kullanılabilir kayıt yok. Zincirdeki sonraki kaydın adresini almak için küme iş yükü çıkışından bir MQXCLWLN çağrısı yayınlandı. Geçerli kayıt, zincirdeki son kayıttır. Düzeltici işlem: Yok.

CompCode MQCC_FAILEDise:

MQRC_CURRENT_RECORD_ERROR
(2357, X'0935')

CurrentRecord değiştirgesi geçersiz. Zincirdeki sonraki kaydın adresini almak için küme iş yükü çıkışından bir MQXCLWLN çağrısı yayınlandı. **CurrentRecord** değiştirgesiyle belirtilen adres, geçerli bir kaydın adresi değil.

CurrentRecord bir hedef kaydın, MQWDR, kuyruk kaydının (MQWQR) ya da küme kaydının (MQWCR) adresi olmalıdır küme önbelleğinde bulunur. Düzeltici işlem: Küme iş yükü çıkışının, küme önbelleğinde bulunan geçerli bir kaydın adresini geçtiğinden emin olun.

MQRC_ENVIRONMENT_ERROR
(2012, X'07DC')

Çağrı ortamda geçerli değil. Bir MQXCLWLN çağrısı yayınlandı, ancak bir küme iş yükü çıkışından değil.

MQRC_NEXT_OFFSET_ERROR
(2358, X'0936')

NextOffset değiştirgesi geçersiz. Zincirdeki sonraki kaydın adresini almak için küme iş yükü çıkışından bir MQXCLWLN çağrısı yayınlandı. **NextOffset** değiştirgesiyle belirtilen görelî konum geçerli değil. **NextOffset** aşağıdaki alanlardan birinin değeri olmalıdır:

- MQWDR içindeki ChannelDefGörelî Konum alanı
- ClusterRecOffset alanı- MQWDR
- ClusterRecOffset alanı- MQWQR
- ClusterRecOffset alanı- MQWCR

Düzeltme işlemi: **NextOffset** değiştirgesi için belirtilen değerin, daha önce listelenen alanlardan birinin değeri olduğundan emin olun.

MQRC_NEXT_RECORD_ERROR
(2361, X'0939')

NextRecord değiştirgesi geçersiz.

MQRC_WXP_ERROR
(2356, X'0934')

İş yükü çıkış parametresi yapısı geçersiz. Zincirdeki sonraki kaydın adresini almak için küme iş yükü çıkışından bir MQXCLWLN çağrısı yayınlandı. **ExitParms** iş yükü çıkış değiştirgesi yapısı geçersiz; bunun nedeni aşağıdakilerden biridir:

- Parametre göstergesi geçerli değil. Geçerli olmayan parametre işaretçileri her zaman algılanamaz; saptanmazsa, beklenmedik sonuçlar ortaya çıkar.
- StrucId alanı MQWXP_STRUC_IDdeğil.
- Sürüm alanı MQWXP_VERSION_2değil.
- Bağlam alanı, kuyruk yöneticisi tarafından çıkışa geçirilen değeri içermiyor.

Düzeltici işlem: **ExitParms** için belirtilen parametrenin, çıkış çağrıldığında çıkışa geçirilen MQWXP yapısı olduğundan emin olun.

İlgili başvurular

MQXCLWLN için kullanım notları-Küme iş yükü kayıtlarında gezinme
Önbellek statik olsa da küme kayıtlarında dolaşmak için MQXCLWLN kullanın.

MQXCLWLN dil çağruları

MQXCLWLN , C ve High Level Assembler olmak üzere iki dili destekler.

MQXCLWLN için kullanım notları-Küme iş yükü kayıtlarında gezinme

Önbellek statik olsa da küme kayıtlarında dolaşmak için MQXCLWLN kullanın.

Küme önbelleği dinamikse, kayıtlar arasında dolaşmak için MQXCLWLN çağrısı kullanılmalıdır. Kayıtlar arasında dolaşmak için basit işaretçi ve görel konum aritmetiği kullanılırsa, çıkış olağandışı sona erer.

Küme önbelleği durağansa, kayıtlar arasında dolaşmak için MQXCLWLN kullanılması gerekmez. Önbellek statik olduğunda bile genellikle MQXCLWLN kullanın. Daha sonra, iş yükü çıkışını değiştirmeniz gerekmeden küme önbelleğini dinamik olarak değiştirebilirsiniz.

İlgili başvurular

MQXCLWLN için parametreler-Küme iş yükü kayıtlarında gezinme

MQXCLWLN çağrısında parametrelerin açıklaması.

MQXCLWLN dil çağruları

MQXCLWLN , C ve High Level Assembler olmak üzere iki dili destekler.

MQXCLWLN dil çağruları

MQXCLWLN , C ve High Level Assembler olmak üzere iki dili destekler.

C çağrısı

```
MQXCLWLN (&ExitParms, CurrentRecord, NextOffset, &NextRecord, &CompCode, &Reason) ;
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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 */
}
```

High Level Assembler (Yüksek Düzeyli çevirici) çağrısı

```
CALL MQXCLWLN, (CLWLEXITPARMS, CURRENTRECORD, NEXTOFFSET, NEXTRECORD, COMPCODE, REASON)
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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
```

İlgili başvurular

MQXCLWLN için parametreler-Küme iş yükü kayıtlarında gezinme

MQXCLWLN çağrısında parametrelerin açıklaması.

MQXCLWLN için kullanım notları-Küme iş yükü kayıtlarında gezinme

Önbellek statik olsa da küme kayıtlarında dolaşmak için MQXCLWLN kullanın.

MQWXP -Küme iş yükü çıkış parametresi yapısı

Aşağıdaki çizelge, MQWXP -Küme iş yükü çıkış parametresi yapısındaki alanları özetler.

Çizelge 826. MQWXP içindeki alanlar		
Alan	Açıklama	Sayfa
<i>StrucId</i>	Yapı tanıtıcısı	StrucId
<i>Version</i>	Yapı sürümü numarası	Sürüm
<i>ExitId</i>	Çıkış tipi	ExitId
<i>ExitReason</i>	Çıkışın çağrılmasının nedeni	ExitReason
<i>ExitResponse</i>	Çıkıştan yanıt	ExitResponse
<i>ExitResponse2</i>	Çıkıştan gelen ikincil yanıt	ExitResponse2
<i>Feedback</i>	Geribildirim kodu	Geri bildirim
<i>Flags</i>	Değerleri işaretler. Bu bit işaretleri, konan iletiye ilişkin bilgileri belirtmek için kullanılır	İşaretler
<i>ExitUserArea</i>	Kullanıcı alanından çık	ExitUserAlanı
<i>ExitData</i>	Verilerden çık	ExitData
<i>MsgDescPtr</i>	İleti tanımlayıcısının adresi (MQMD)	MsgDescPtr
<i>MsgBufferPtr</i>	İleti verilerinin bir kısmını ya da tümünü içeren arabelleğin adresi	MsgBufferPtr
<i>MsgBufferLength</i>	İleti verilerini içeren arabelleğin uzunluğu	MsgBufferUzunluğu
<i>MsgLength</i>	Tam iletinin uzunluğu	MsgLength
<i>QName</i>	Kuyruğun adı	QName
<i>QMgrName</i>	Yerel kuyruk yöneticisinin adı	QMgrName
<i>DestinationCount</i>	Olası hedeflerin sayısı	DestinationCount
<i>DestinationChosen</i>	Seçilen hedef	DestinationChosen
<i>DestinationArrayPtr</i>	Hedef kayıtlara ilişkin işaretçi dizisinin adresi (MQWDR)	DestinationArrayPtr
<i>QArrayPtr</i>	Kuyruk kayıtlarına ilişkin işaretçi dizisinin adresi (MQWQR)	QArrayPtr
Not: Sürüm MQWXP_VERSION_2değerinden küçükse, geri kalan alanlar yoksayılr.		
<i>CacheContext</i>	Bağlam Bilgileri	CacheContext
<i>CacheType</i>	Küme önbelleğinin tipi	CacheType
Not: Sürüm MQWXP_VERSION_3değerinden küçükse, geri kalan alanlar yoksayılr.		
<i>CLWLMRUChannels</i>	İzin verilen etkin giden küme kanalı sayısı üst sınırı	CLWLMRUChannel
Not: Sürüm MQWXP_VERSION_4değerinden küçükse, geri kalan alanlar yoksayılr.		
<i>pEntryPoints</i>	MQI ve DCI çağrılarının yapılmasına izin veren MQIEP yapısının adresi	pEntryNoktalar

Küme iş yükü çıkış parametresi yapısı, küme iş yükü çıkışına geçirilen bilgileri açıklar.

Küme iş yükü çıkış parametresi yapısı tüm platformlarda desteklenir

Buna ek olarak, MQWXP1, MQWXP2 ve MQWXP3 yapıları geriye dönük uyumluluk için kullanılabilir.

İlgili başvurular

MQ_CLUSTER_WORKLOAD_EXIT -Arama açıklaması

Bir iletiyi kullanılabilir bir kuyruk yöneticisine yöneltmek için kuyruk yöneticisi tarafından küme iş yükü çıkışı çağrılır.

MQXCLWLN -Küme iş yükü kayıtlarında gezinme

MQXCLWLN çağırısı, küme önbelleğinde saklanan MQWDR, MQWQR ve MQWCR kayıtları zincirlerinde dolaşmak için kullanılır.

MQWDR-Küme iş yükü hedef kayıt yapısı

Aşağıdaki çizelge, MQWDR -Küme iş yükü hedef kayıt yapısındaki alanları özetler.

MQWQR -Küme iş yükü kuyruğu kayıt yapısı

Aşağıdaki çizelge, MQWQR -Küme iş yükü kuyruğu kayıt yapısındaki alanları özetler.

MQWCR -Küme iş yükü kümesi kayıt yapısı

Aşağıdaki çizelge, MQWCR kümesi iş yükü kayıt yapısındaki alanları özetler.

MQWXP 'deki Alanlar-Küme iş yükü çıkış parametresi yapısı

MQWXP -Küme iş yükü çıkış parametresi yapısındaki alanların açıklaması

StrucId (MQCHAR4)-giriş

Küme iş yükü çıkış parametresi yapısına ilişkin yapı tanıttıcısı.

- StrucId değeri MQWXP_STRUC_ID.
- C programlama dili için, MQWXP_STRUC_ID_ARRAY sabiti de tanımlanır. MQWXP_STRUC_ID ile aynı değere sahiptir. Bir dizgi yerine bir karakter dizisidir.

Sürüm (MQLONG)-giriş

Yapı sürümü numarasını gösterir. Sürüm aşağıdaki değerlerden birini alır:

MQWXP_VERSION_1

Version-1 küme iş yükü çıkış parametresi yapısı.

MQWXP_VERSION_1 tüm ortamlarda desteklenir.

MQWXP_VERSION_2

Version-2 küme iş yükü çıkış parametresi yapısı.

MQWXP_VERSION_2 aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

MQWXP_VERSION_3

Version-3 küme iş yükü çıkış parametresi yapısı.

MQWXP_VERSION_3 aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

MQWXP_VERSION_4

Version-4 küme iş yükü çıkış parametresi yapısı.

MQWXP_VERSION_4 aşağıdaki ortamlarda desteklenir:

-  AIX
-  IBM i
-  Linux
-  Windows

MQWXP_CURRENT_VERSION

Küme iş yükü çıkış parametresi yapısının yürürlükteki sürümü.

ExitId (MQLONG)-giriş

Çağrılmakta olan çıkışın tipini gösterir. Desteklenen tek çıkış küme iş yükü çıkışıdır.

- ExitId değeri MQXT_CLUSTER_WORKLOAD_EXIT olmalıdır

ExitReason (MQLONG)-giriş

Küme iş yükü çıkışının çağrılmasının nedenini gösterir. ExitReason aşağıdaki değerlerden birini alır:

MQXR_INIT

Çıkışın ilk kez çağrıldığını gösterir.

Ana saklama alanı gibi, çıkışın gereksinim duyabileceği kaynakları edinin ve kullanıma hazırlayın.

MQXR_TERM

Çıkışın sonlandırılacağını gösterir.

Çıkış başlatıldığından bu yana edinmiş olabileceği kaynakları (ana saklama alanı gibi) serbest bırakın.

MQXR_CLWL_OPEN

MQOPEN tarafından çağrılır.

MQXR_CLWL_PUT

MQPUT ya da MQPUT1 tarafından çağrılır.

MQXR_CLWL_MOVE

Kanal durumu değiştiğinde MCA tarafından çağrılır.

MQXR_CLWL_REPOS

Havuz yöneticisi PCF iletisi için MQPUT ya da MQPUT1 tarafından çağrılır.

MQXR_CLWL_REPOS_MOVE

Kanal durumu değiştiyse, havuz yöneticisi PCF iletisi için MCA tarafından çağrılır.

ExitResponse (MQLONG)-çıkış

İletinin işlenmesinin devam edip etmediğini belirtmek için ExitResponse değerini ayarlayın.

Aşağıdaki değerlerden biri olmalıdır:

MQXCC_OK

İletiyi olağan şekilde işlemeye devam edin.

- DestinationChosen , iletinin gönderileceği hedefi belirtir.

MQXCC_SUPPRESS_FUNCTION

İletiyi işlemeye devam etme.

- Kuyruk yöneticisi tarafından yapılan işlemler, çıkışın çağrılmasının nedenine bağlıdır:

Çizelge 827. Kuyruk yöneticisi tarafından gerçekleştirilen işlemler	
ExitReason	Uygulanan İşlem
<ul style="list-style-type: none"> - MQXR_CLWL_OPEN - MQXR_CLWL_REPOS - MQXR_CLWL_PUT 	MQOPEN, MQPUTya da MQPUT1 çağrı tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_STOPPED_BY_CLUSTER_EXITile başarısız olur.
<ul style="list-style-type: none"> - MQXR_CLWL_MOVE - MQXR_CLWL_REPOS_MOVE 	İleti, gönderilmeyen iletiler kuyruğuna yerleştirilir.

MQXCC_SUPPRESS_EXIT

Yürürlükteki iletiyi olağan şekilde işlemeye devam edin. Kuyruk yöneticisi kapanıncaya kadar çıkışı yeniden çağırmayın.

Kuyruk yöneticisi, ClusterWorkloadExit kuyruk yöneticisi özniteliği boşsa, sonraki iletileri işler. DestinationChosen , geçerli iletinin gönderildiği hedefi belirtir.

Diğer herhangi bir değer

İletiyi MQXCC_SUPPRESS_FUNCTION belirtilmiş gibi işleyin.

ExitResponse2 (MQLONG)-giriş/çıkış

Kuyruk yöneticisine daha fazla bilgi sağlamak için ExitResponse2 ayarını belirleyin.

- MQXR2_STATIC_CACHE varsayılan değerdir ve çıkışa girişte ayarlanır.
- ExitReason , MQXR_INITdeğerine sahip olduğunda, çıkış ExitResponse2içinde aşağıdaki değerlerden birini ayarlayabilir:

MQXR2_STATIC_CACHE

Çıkış için statik bir küme önbelleği gerekir.

- Küme önbelleği durağansa, çıkışın küme önbelleğindeki kayıt zincirlerinde dolaşmak için MQXCLWLN çağrısına gerek yoktur.
- Küme önbelleği dinamikse, çıkış önbellekteki kayıtlar arasında doğru bir şekilde dolaşamaz.

Not: Kuyruk yöneticisi, MQXR_INIT çağrısından gelen dönüşü, ExitResponse alanında MQXCC_SUPPRESS_EXIT çıkış döndürülmüş gibi işler.

MQXR2_DYNAMIC_CACHE

Çıkış, statik ya da dinamik bir önbellekle çalışabilir.

- Çıkış bu değeri döndürürse, çıkışın küme önbelleğindeki kayıt zincirlerinde dolaşmak için MQXCLWLN çağrısının kullanılması gerekir.

Geribildirim (MQLONG)-giriş

Ayrılmış bir alan. Değer sıfırdır.

İşaretler (MQLONG)-giriş

Konmakta olan iletiye ilişkin bilgileri gösterir.

- İşaretler değeri MQWXP_PUT_BY_CLUSTER_CHLdeğeridir. İleti, yerel olarak ya da küme olmayan bir kanaldan değil, bir küme kanalından geliyor. Başka bir deyişle, ileti başka bir küme kuyruk yöneticisinden geldi.

Ayrılmış (MQLONG)-giriş

Ayrılmış bir alan. Değer sıfırdır.

ExitUserArea (MQBYTE16)-giriş/çıkış

Çıkışa yapılan çağrılar arasında iletişim kurmak için ExitUserArea (ExitUser Alanı) seçeneğini belirleyin.

- ExitUserArea , çıkışın ilk çağrılmasından önce ikili sıfır olarak başlatılır. Çıkış tarafından bu alanda yapılan değişiklikler, MQCONN çağrı ile eşleşen MQDISC çağrı arasında oluşan çıkışın çağrılmaları boyunca korunur. MQDISC çağrısı gerçekleştiğinde alan ikili sıfıra sıfırlanır.

- Çıkışın ilk çağırısı, MQXR_INIT değerine sahip ExitReason alanıyla gösterilir.
- Aşağıdaki sabitler tanımlanır:

MQXUA_NONE -dizgi

MQXUA_NONE_ARRAY -karakter dizisi

Kullanıcı bilgisi yok. Her iki değişmez de alanın uzunluğu için ikili sıfırdır.

MQ_EXIT_USER_AREA_LENGTH

ExitUserArea uzunluğu.

ExitData (MQCHAR32)-giriş

ClusterWorkloadData kuyruk yöneticisi özniteliğinin değeri. Bu öznitelik için herhangi bir değer tanımlanmadıysa, bu alan tamamen boşluktur.

- ExitData uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir.

MsgDescPtr (PMQMD)-giriş

İşlenmekte olan iletiye ilişkin ileti tanımlayıcısının (MQMD) bir kopyasının adresi.

- Çıkış tarafından ileti tanımlayıcısında yapılan değişiklikler kuyruk yöneticisi tarafından yoksayılır.
- ExitReason aşağıdaki değerlerden birine sahipse MsgDescPtr boş değerli göstergeye ayarlanır ve çıkışa ileti tanımlayıcısı geçirilmez:
 - MQXR_INIT
 - MQXR_TERM
 - MQXR_CLWL_OPEN

MsgBufferPtr (PMQVOID)-giriş

İleti verilerinin ilk MsgBufferLength byte 'ının bir kopyasını içeren arabelleğin adresi.

- Çıkış tarafından ileti verilerinde yapılan değişiklikler kuyruk yöneticisi tarafından yoksayılır.
- Aşağıdaki durumda çıkışa ileti verisi iletilmez:
 - MsgDescPtr boş değerli göstergedir.
 - İletide veri yok.
 - ClusterWorkloadLength kuyruk yöneticisi özniteliği sıfırdır.

Bu durumlarda, MsgBufferPtr boş değerli işaretçidir.

MsgBufferLength (MQLONG)-giriş

Çıkışa geçirilen ileti verilerini içeren arabelleğin uzunluğu.

- Uzunluk, ClusterWorkloadLength kuyruk yöneticisi özniteliği tarafından denetlenir.
- Uzunluk, iletinin tamamının uzunluğundan az olabilir; bkz. MsgLength.

MsgLength (MQLONG)-giriş

Çıkışa geçirilen tam iletinin uzunluğu.

- MsgBufferLength tam iletinin uzunluğundan az olabilir.
- MsgLength , ExitReason MQXR_INIT, MQXR_TERM ya da MQXR_CLWL_OPEN ise sıfırdır.

QName (MQCHAR48)-giriş

Hedef kuyruğun adı. Kuyruk bir küme kuyruğudur.

- QName uzunluğu MQ_Q_NAME_LENGTH' dir.

QMgrName (MQCHAR48)-giriş

Küme iş yükü çıkışını çağırın yerel kuyruk yöneticisinin adı.

- QMgrName uzunluğu MQ_Q_MGR_NAME_LENGTH' dir.

DestinationCount (MQLONG)-giriş

Olası varış noktalarının sayısı. Hedefler, hedef kuyruğun eşgörünümleridir ve hedef kayıtlar tarafından açıklanır.

- Hedef kayıt bir MQWDR yapısıdır. Kuyruğun her eşgörünümü için olası her rota için bir yapı vardır.
- MQWDR yapıları bir işaretçi dizisi tarafından adreslenir, bkz. DestinationArrayPtr.

DestinationChosen (MQLONG)-giriş/çıkış

Seçilen hedef.

- İletinin gönderileceği rotayı ve kuyruk eşgörünümünü tanıtan MQWDR yapısının numarası.
- Değer, 1- DestinationCount aralığındadır.
- Çıkışa giriş yapan DestinationChosen , kuyruk yöneticisinin seçtiği rota ve kuyruk eşgörünümünü gösterir. Çıkış bu seçeneği kabul edebilir ya da farklı bir rota ve kuyruk örneği seçebilir.
- Çıkış tarafından ayarlanan değer 1- DestinationCount aralığında olmalıdır. Başka bir değer döndürülürse, kuyruk yöneticisi çıkışa girişte DestinationChosen değerini kullanır.

DestinationArrayPtr (PPMQWDR)-giriş

Hedef kayıtlara ilişkin işaretçi dizisinin adresi (MQWDR).

- DestinationCount hedef kaydı vardır.

QArrayPtr (PPMQWR)-giriş

Kuyruk kayıtlarına ilişkin işaretçi dizisinin adresi (MQWR).

- Kuyruk kayıtları varsa, bunlardan DestinationCount vardır.
- Kullanılabilir kuyruk kaydı yoksa, boş değerli gösterge QArrayPtr olur.

Not: QArrayPtr , DestinationCount sıfırdan büyük olduğunda bile boş değerli gösterge olabilir.

CacheContext (MQPTR): Sürüm 2-giriş

CacheContext alanı, kuyruk yöneticisi tarafından kullanılmak üzere ayrılmıştır. Çıkış, bu alanın değerini değiştirmemelidir.

CacheType (MQLONG): Sürüm 2-giriş

Küme önbelleği aşağıdaki tiplerden birine sahiptir:

MQCLCT_STATIC

Önbellek statik.

- Önbelleğin boyutu sabittir ve kuyruk yöneticisi çalıştıkça büyüyemez.
- Bu önbellek tipindeki kayıtlarda dolaşmak için MQXCLWLN aramasını kullanmanıza gerek yoktur.

MQCLCT_DYNAMIC

Önbellek dinamik.

- Değişen küme bilgilerini barındırmak için önbelleğin boyutu artabilir.
- Bu önbellek tipindeki kayıtlarda dolaşmak için MQXCLWLN çağrısının kullanılması gerekir.

CLWLMRUChannel (MQLONG): Sürüm 3-giriş

Küme iş yükü seçimi algoritması tarafından kullanılmak üzere değerlendirilecek etkin giden küme kanalı sayısı üst sınırını belirtir.

- CLWLMRUChannel , 1-999 999 999 arasında bir değerdir.

pEntryPoints (PMQIEP): Sürüm 4

MQI ve DCI çağrılarının yapılabileceği bir MQIEP yapısının adresi.

İlgili başvurular

MQWXP için ilk değerler ve dil bildirimleri

MQWXP -Küme iş yükü çıkış parametresi yapısı için ilk değerler ve C ve High Level Assembler Dil bildirimleri.

MQWXP için ilk değerler ve dil bildirimleri

MQWXP -Küme iş yükü çıkış parametresi yapısı için ilk değerler ve C ve High Level Assembler Dil bildirimleri.

Çizelge 828. MQWXP içindeki alanlar

Alan adı	Değişmezin adı	Değişmezin değeri
<i>StrucId</i>	MQWXP_STRUC_ID	'WXP↵'
<i>Version</i>	MQWXP_VERSION_2	2
<i>ExitId</i>	Yok	0
<i>ExitReason</i>	MQXCC_OK	0
<i>ExitResponse</i>	Yok	0
<i>ExitResponse2</i>	Yok	0
<i>Flags</i>	Yok	0
<i>ExitUserArea</i>	{MQXUA_NONE_ARRAY}	0
<i>ExitData</i>	Yok	""
<i>MsgDescPtr</i>	Yok	NULL
<i>MsgBufferPtr</i>	Yok	NULL
<i>MsgBufferLength</i>	Yok	0
<i>MsgBufferPtr</i>	Yok	0
<i>QName</i>	Yok	""
<i>QMgrName</i>	Yok	""
<i>DestinationCount</i>	Yok	0
<i>DestinationChosen</i>	Yok	0
<i>DestinationArrayPtr</i>	Yok	NULL
<i>QArrayPtr</i>	Yok	NULL
<i>CacheContext</i>	Yok	NULL
<i>CacheType</i>	MQCLCT_DYNAMIC	1
<i>CLWLMRUChannels</i>	Yok	0
<i>pEntryPoints</i>	Yok	NULL

Notlar:

1. ↵ simgesi, tek bir boş karakteri temsil eder.
2. C programlama dilinde, MQWXP_DEFAULT makro değişkeni varsayılan değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQWDR MyWXP = {MQWXP_DEFAULT};
```

C bildirimi

```
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    CLWLMRUChannels;     /* 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
*                                     data
MQWXP_MSGBUFFERLENGTH                DS    F      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_CLWLMRUCHANNELS                 DS    F      Number of most recently used
*                                     channels for workload balancing

MQWXP_LENGTH                          EQU    *-MQWXP Length of structure
MQWXP_AREA                             ORG    MQWXP
                                         DS    CL(MQWXP_LENGTH)

```

İlgili başvurular

[MQWXP ' deki Alanlar-Küme iş yükü çıkış parametresi yapısı](#)

[MQWXP -Küme iş yükü çıkış parametresi yapısındaki alanların açıklaması](#)

MQWDR-Küme iş yükü hedef kayıt yapısı

Aşağıdaki çizelge, MQWDR -Küme iş yükü hedef kayıt yapısındaki alanları özetler.

Çizelge 829. MQWDR içindeki alanlar		
Alan	Açıklama	Sayfa
<i>StrucId</i>	Yapı tanıtıcısı	StrucId
<i>Version</i>	Yapı sürümü numarası	Sürüm
<i>StrucLength</i>	MQWDR yapısının uzunluğu	StrucLength
<i>QMgrFlags</i>	Kuyruk yöneticisi işaretleri	QMgrFlags
<i>QMgrIdentifier</i>	Kuyruk yöneticisi tanıtıcısı	QMgrIdentifier
<i>QMgrName</i>	Kuyruk yöneticisi adı	QMgrName
<i>ClusterRecOffset</i>	İlk küme kaydının mantıksal görel konumu (MQWCR)	ClusterRecGörel Konumu
<i>ChannelState</i>	Kanal durumu	ChannelState
<i>ChannelDefOffset</i>	Kanal tanımlaması yapısının mantıksal görel konumu (MQCD)	ChannelDefKayması
Not: Sürüm MQWDR_VERSION_2değerinden küçükse, geri kalan alanlar yoksayılr.		
<i>DestSeqNumber</i>	Kanal hedef sıra numarası	DestSeqNumarası
<i>DestSeqFactor</i>	Ağırlıklandırma için kanal hedef sıra katsayısı	DestSeqKatsayısı

Küme iş yükü hedef kayıt yapısı, iletiye ilişkin olası hedeflerden biriyle ilgili bilgileri içerir. Hedef kuyruğun her eşgörünümü için bir küme iş yükü hedef kayıt yapısı vardır.

Küme iş yükü hedef kayıt yapısı tüm ortamlarda desteklenir.

Buna ek olarak, geriye dönük uyumluluk için MQWDR1 ve MQWDR2 yapıları da kullanılabilir.

İlgili başvurular

[MQ_CLUSTER_WORKLOAD_EXIT](#) -Arama açıklaması

Bir iletiyi kullanılabilir bir kuyruk yöneticisine yöneltmek için kuyruk yöneticisi tarafından küme iş yükü çıkışı çağrılır.

[MQXCLWLN](#) -Küme iş yükü kayıtlarında gezinme

[MQXCLWLN](#) çağırısı, küme önbelleğinde saklanan MQWDR, MQWQRve MQWCR kayıtları zincirlerinde dolaşmak için kullanılır.

[MQWXP](#) -Küme iş yükü çıkış parametresi yapısı

Aşağıdaki çizelge, [MQWXP](#) -Küme iş yükü çıkış parametresi yapısındaki alanları özetler.

[MQWQR](#) -Küme iş yükü kuyruğu kayıt yapısı

Aşağıdaki çizelge, [MQWQR](#) -Küme iş yükü kuyruğu kayıt yapısındaki alanları özetler.

[MQWCR](#) -Küme iş yükü kümesi kayıt yapısı

Aşağıdaki çizelge, [MQWCR](#) kümesi iş yükü kayıt yapısındaki alanları özetler.

MQWDR-Küme iş yükü hedef kayıt yapısındaki alanlar

MQWDR -Küme iş yükü hedef kayıt yapısındaki parametrelerin açıklaması.

StrucId (MQCHAR4) -giriş

Küme iş yükü hedef kayıt yapısına ilişkin yapı tanıtıcısı.

- StrucId değeri MQWDR_STRUC_ID.

- C programlama dili için, MQWDR_STRUC_ID_ARRAY sabiti de tanımlanır. MQWDR_STRUC_ID ile aynı değere sahiptir. Bir dizgi yerine bir karakter dizisidir.

Sürüm (MQLONG) -giriş

Yapı sürümü numarası. Sürüm aşağıdaki değerlerden birini alır:

MQWDR_VERSION_1

Version-1 kümesi iş yükü hedef kaydı.

MQWDR_VERSION_2

Version-2 küme iş yükü hedef kaydı.

MQWDR_CURRENT_VERSION

Küme iş yükü hedef kaydının geçerli sürümü.

StrucLength (MQLONG) -giriş

MQWDR yapısının uzunluğu. StrucLength aşağıdaki değerlerden birini alır:

MQWDR_LENGTH_1

version-1 küme iş yükü hedef kaydının uzunluğu.

MQWDR_LENGTH_2

version-2 küme iş yükü hedef kaydının uzunluğu.

MQWDR_CURRENT_LENGTH

Küme iş yükü hedef kaydının yürürlükteki sürümünün uzunluğu.

QMgrFlags (MQLONG) -giriş

MQWDR yapısıyla açıklanan hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinin özelliklerini gösteren kuyruk yöneticisi işaretleri. Şu işaretler tanımlandı:

MQQMF_REPOSITORY_Q_MGR

Hedef, tam bir havuz kuyruğu yöneticisidir.

MQQMF_CLUSSDR_USER_DEFINED

Küme-gönderen kanalı el ile tanımlandı.

MQQMF_CLUSSDR_AUTO_DEFINED

Küme-gönderen kanalı otomatik olarak tanımlandı.

MQQMF_AVAILABLE

Hedef kuyruk yöneticisi ileti almak için kullanılabilir.

Diğer değerler

Alandaki diğer işaretler, iç amaçlar için kuyruk yöneticisi tarafından ayarlanabilir.

QMgrIdentifier (MQCHAR48) -giriş

Kuyruk yöneticisi tanıtıcısı, MQWDR yapısıyla açıklanan hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisine ilişkin benzersiz bir tanıtıcıdır.

- Tanıtıcı, kuyruk yöneticisi tarafından oluşturulur.
- QMgrIdentifier uzunluğu MQ_Q_MGR_IDENTIFIER_LENGTH' dir.

QMgrName (MQCHAR48) -giriş

MQWDR yapısının tanımladığı hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinin adı.

- QMgrName , yerel kuyruk yöneticisinin adı ve kümedeki başka bir kuyruk yöneticisi olabilir.
- QMgrName uzunluğu MQ_Q_MGR_NAME_LENGTH' dir.

ClusterRecOffset (MQLONG) -giriş

MQWDR yapısına ait ilk MQWCR yapısının mantıksal görelî konumu.

- Statik önbellekler için ClusterRecOffset , MQWDR yapısına ait ilk MQWCR yapısının görelî konumudur.
- Görelî konum, MQWDR yapısının başlangıcından itibaren bayt cinsinden ölçülür.
- Dinamik önbelleklerle işaretçi aritmetiği için mantıksal görelî konumu kullanmayın. Sonraki kaydın adresini almak için MQXCLWLN araması kullanılmalıdır.

ChannelState (MQLONG) -giriş

Yerel kuyruk yöneticisini MQWDR yapısıyla tanıtilan kuyruk yöneticisine bağlayan kanalın durumu. Aşağıdaki değerler kullanılabilir:

MQCHS_BINDING

Kanal ortakla pazarlık yapıyor.

MQCHS_INACTIVE

Kanal etkin değil.

MQCHS_INITIALIZING

Kanal başlatılıyor.

MQCHS_PAUSED

Kanal durakladı.

MQCHS_REQUESTING

İstekte bulunan kanal bağlantı istiyor.

MQCHS_RETRYING

Kanal yeniden bağlantı kurmaya çalışıyor.

MQCHS_RUNNING

Kanal ileti aktarıyor ya da bekliyor.

MQCHS_STARTING

Kanal aktif olmayı bekliyor.

MQCHS_STOPPING

Kanal duruyor.

MQCHS_STOPPED

Kanal durdu.

ChannelDefGörelİ Konumu (MQLONG) -giriş

Kanal tanımlamasının mantıksal görelİ konumu (MQCD) yerel kuyruk yöneticisini MQWDR yapısıyla tanıtilan kuyruk yöneticisine bağlayan kanal için.

- ChannelDefGörelİ Konumu , ClusterRecGörelİ Konumu
- Mantıksal görelİ konum, gösterge aritmetiğinde kullanılamaz. Sonraki kaydın adresini almak için MQXCLWLN çağırısı kullanılmalıdır.

DestSeqKatsayısı (MQLONG) -giriş

Ağırlığa dayalı olarak kanal seçimine izin veren hedef sıra katsayısı.

- DestSeqFaktörİ , kuyruk yöneticisi tarafından değiştirilmeden önce kullanılır.
- İş yükü yöneticisi, DestSeqFactor ürününü, iletilerin ağırlığına göre dağıtılmasını güvenceye almak üzere artırır.

DestSeqNumarası (MQLONG) -giriş

Kuyruk yöneticisi değiştirmeden önce küme kanalı hedef değeri.

- İş yükü yöneticisi, her ileti gönderildiğinde DestSeqNumarası ' nı artırır.
- İş yükü çıkışları, iletiyi hangi kanaldan göndereceğinize karar vermek için DestSeqNumarası ' nı kullanabilir.

İlgili başvurular

MQWDR için ilk değerler ve dil bildirimleri

MQWDR -Cluster iş yükü hedef kaydı için ilk değerler ve C ve High Level Assembler Dil bildirimleri.

MQWDR için ilk değerler ve dil bildirimleri

MQWDR -Cluster iş yükü hedef kaydı için ilk değerler ve C ve High Level Assembler Dil bildirimleri.

Çizelge 830. MQWDR içindeki alanlar

Alan adı	Değişmezin adı	Değişmezin değeri
<i>StrucId</i>	MQWDR_STRUC_ID	'WDR↵'
<i>Version</i>	MQWDR_VERSION_1	1
<i>StrucLength</i>	MQWDR_CURRENT_LENGTH ³	136
<i>QMgrFlags</i>	MQWDR_NONE	0
<i>QMgrIdentifier</i>	Yok	" "
<i>QMgrName</i>	Yok	" "
<i>ClusterRecOffset</i>	Yok	0
<i>ChannelState</i>	Yok	0
<i>ChannelDefOffset</i>	Yok	0
<i>DestSeqNumber</i>	Yok	0
<i>DestSeqFactor</i>	Yok	0

Notlar:

1. ↵ simgesi, tek bir boş karakteri temsil eder.
2. C programlama dilinde, MQWDR_DEFAULT makro değişkeni varsayılan değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQWDR MyWDR = {MQWDR_DEFAULT};
```

3. İlk değerler, yapının uzunluğunu bilerek yapının sürüm 1 'e değil, geçerli sürümün uzunluğuna ayarlar.

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      ORG  MQWDR
                DS   CL(MQWDR_LENGTH)
```

C bildirim

```
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 */
};

```

İlgili başvurular

[MQWDR-Küme iş yükü hedef kayıt yapısındaki alanlar](#)

[MQWDR -Küme iş yükü hedef kayıt yapısındaki parametrelerin açıklaması.](#)

MQWQR -Küme iş yükü kuyruğu kayıt yapısı

Aşağıdaki çizelge, MQWQR -Küme iş yükü kuyruğu kayıt yapısındaki alanları özetler.

Çizelge 831. MQWQR içindeki alanlar		
Alan	Açıklama	Sayfa
<i>StrucId</i>	Yapı tanıtıcısı	StrucId
<i>Version</i>	Yapı sürümü numarası	Sürüm
<i>StrucLength</i>	MQWQR yapısının uzunluğu	StrucLength
<i>QFlags</i>	Kuyruk işaretleri	QFlags
<i>QName</i>	Kuyruk adı	QName
<i>QMgrIdentifier</i>	Kuyruk yöneticisi tanıtıcısı	QMgrIdentifier
<i>ClusterRecOffset</i>	İlk küme kaydının görel konumu (MQWCR)	ClusterRecGörel Konumu
<i>QType</i>	Kuyruk tipi	QType
<i>QDesc</i>	Kuyruk tanımlaması	QDesc
<i>DefBind</i>	Varsayılan bağ tanımı	DefBind
<i>DefPersistence</i>	Varsayılan ileti kalıcılığı	DefPersistence
<i>DefPriority</i>	Varsayılan ileti önceliği	DefPriority
<i>InhibitPut</i>	Kuyruğa koyma işlemlerine izin verilip verilmediğini belirler	InhibitPut
Not: Sürüm MQWQR_VERSION_2değerinden küçükse, geri kalan alanlar yoksayılr.		
<i>CWLQueuePriority</i>	Kuyruğun önceliğini gösteren 0-9 arası bir değer	CLWLQueuePriority
<i>CLWLQueueRank</i>	Kuyruğun sırasını gösteren 0-9 arası bir değer	CLWLQueueRank
Not: Sürüm MQWQR_VERSION_3değerinden küçükse, geri kalan alanlar yoksayılr.		
<i>DefPutResponse</i>	Varsayılan koyma yanıtı	DefPutYanıtı

Küme iş yükü kuyruğu kayıt yapısı, iletiye ilişkin olası hedeflerden biriyle ilgili bilgileri içerir. Hedef kuyruğun her eşgörünümü için bir küme iş yükü kuyruğu kayıt yapısı vardır.

Küme iş yükü kuyruğu kayıt yapısı tüm ortamlarda desteklenir.

Ayrıca, geriye dönük uyumluluk için MQWQR1 ve MQWQR2 yapıları da kullanılabilir.

İlgili başvurular

[MQ_CLUSTER_WORKLOAD_EXIT -Arama açıklaması](#)

Bir iletiyi kullanılabilir bir kuyruk yöneticisine yönlendirmek için kuyruk yöneticisi tarafından küme iş yükü çıkışı çağrılır.

[MQXCLWLN -Küme iş yükü kayıtlarında gezinme](#)

MQXCLWLN çağırısı, küme önbelleğinde saklanan MQWDR, MQWQR ve MQWCR kayıtları zincirlerinde dolaşmak için kullanılır.

MQWXP -Küme iş yükü çıkış parametresi yapısı

Aşağıdaki çizelge, MQWXP -Küme iş yükü çıkış parametresi yapısındaki alanları özetler.

MQWDR-Küme iş yükü hedef kayıt yapısı

Aşağıdaki çizelge, MQWDR -Küme iş yükü hedef kayıt yapısındaki alanları özetler.

MQWCR -Küme iş yükü kümesi kayıt yapısı

Aşağıdaki çizelge, MQWCR kümesi iş yükü kayıt yapısındaki alanları özetler.

MQWQR ' deki alanlar-Küme iş yükü kuyruğu kayıt yapısı

MQWQR -Küme iş yükü kuyruğu kayıt yapısındaki alanların açıklaması.

StrucId (MQCHAR4) -giriş

Küme iş yükü kuyruğu kayıt yapısına ilişkin yapı tanıtıcısı.

- StrucId değeri MQWQR_STRUC_ID.
- C programlama dili için, MQWQR_STRUC_ID_ARRAY sabiti de tanımlanır. MQWQR_STRUC_ID ile aynı değere sahiptir. Bir dizgi yerine bir karakter dizisidir.

Sürüm (MQLONG) -giriş

Yapı sürümü numarası. Sürüm aşağıdaki değerlerden birini alır:

MQWQR_VERSION_1

Version-1 küme iş yükü kuyruğu kaydı.

MQWQR_VERSION_2

Version-2 küme iş yükü kuyruğu kaydı.

MQWQR_VERSION_3

Version-3 kümesi iş yükü kuyruğu kaydı.

MQWQR_CURRENT_VERSION

Küme iş yükü kuyruğu kaydının yürürlükteki sürümü.

StrucLength (MQLONG) -giriş

MQWQR yapısının uzunluğu. StrucLength aşağıdaki değerlerden birini alır:

MQWQR_LENGTH_1

version-1 küme iş yükü kuyruğu kaydının uzunluğu.

MQWQR_LENGTH_2

version-2 küme iş yükü kuyruğu kaydının uzunluğu.

MQWQR_LENGTH_3

version-3 küme iş yükü kuyruğu kaydının uzunluğu.

MQWQR_CURRENT_LENGTH

Küme iş yükü kuyruğu kaydının yürürlükteki sürümünün uzunluğu.

QFlags (MQLONG) -giriş

Kuyruk işaretleri, kuyruğun özelliklerini gösterir. Şu işaretler tanımlandı:

MQQF_LOCAL_Q

Hedef yerel bir kuyruk.

MQQF_CLWL_USEQ_ANY

Koymada yerel ve uzak kuyrukların kullanılmasına izin verir.

MQQF_CLWL_USEQ_LOCAL

Yalnızca yerel kuyruk koymalarına izin ver.

Diğer değerler

Alandaki diğer işaretler, iç amaçlar için kuyruk yöneticisi tarafından ayarlanabilir.

QName (MQCHAR48) -giriş

İletinin olası hedeflerinden biri olan kuyruğun adı.

- QName uzunluđu MQ_Q_NAME_LENGTH' dir.

QMgrIdentifier (MQCHAR48) -giriş

Kuyruk yöneticisi tanıtıcısı, MQWQR yapısının tanımladığı kuyruk eşgörünümünü barındıran kuyruk yöneticisine ilişkin benzersiz bir tanıtıcıdır.

- Tanıtıcı, kuyruk yöneticisi tarafından oluşturulur.
- QMgrIdentifier uzunluđu MQ_Q_MGR_IDENTIFIER_LENGTH' dir.

ClusterRecOffset (MQLONG) -giriş

MQWQR yapısına ait ilk MQWCR yapısının mantıksal görelî konumu.

- Statik önbellekler için ClusterRecOffset , MQWQR yapısına ait ilk MQWCR yapısının görelî konumudur.
- Görelî konum, MQWQR yapısının başlangıcından itibaren bayt cinsinden ölçülür.
- Dinamik önbelleklerle işaretçi aritmetiđi için mantıksal görelî konumu kullanmayın. Sonraki kaydın adresini almak için MQXCLWLN araması kullanılmalıdır.

QType (MQLONG) -giriş

Hedef kuyruđun kuyruk tipi. Aşğıdaki deđerler kullanılabilir:

MQCQT_LOCAL_Q

Yerel kuyruk.

MQCQT_ALIAS_Q

Diđer ad kuyruđu.

MQCQT_REMOTE_Q

Uzak kuyruk.

MQCQT_Q_MGR_ALIAS

Kuyruk yöneticisi diđer adı.

QDesc (MQCHAR64) -giriş

MQWQR yapısının tanımladığı hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinde tanımlanan kuyruk tanımlaması kuyruđu özniteliđi.

- QDesc uzunluđu MQ_Q_DESC_LENGTH' dir.

DefBind (MQLONG) -giriş

MQWQR yapısıyla açıklanan hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinde tanımlanan varsayılan bađ tanımlama kuyruđu özniteliđi. Kümeler içeren gruplar kullanılırken MQBND_BIND_ON_OPEN ya da MQBND_BIND_ON_GROUP belirtilmelidir. Aşğıdaki deđerler kullanılabilir:

MQBND_BIND_ON_OPEN

MQOPEN çağırısı tarafından düzeltilen bađlama.

MQBND_BIND_NOT_FIXED

Bađlama düzeltilmedi.

MQBND_BIND_ON_GROUP

Bir uygulamanın, bir grup iletinin aynı hedef örneđe ayrılmasını istemesini sađlar.

DefPersistence (MQLONG) -giriş

MQWQR yapısıyla açıklanan hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinde tanımlanan varsayılan ileti kalıcılık kuyruđu özniteliđi. Aşğıdaki deđerler kullanılabilir:

MQPER_PERSISTENT

İleti kalıcı.

MQPER_NOT_PERSISTENT

İleti kalıcı deđil.

DefPriority (MQLONG) -giriş

MQWQR yapısının tanımladığı hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinde tanımlanan varsayılan ileti önceliđi kuyruđu özniteliđi. Öncelik aralıđı: 0- MaxPriority.

- 0 en düşük önceliktir.
- `MaxPriority` , hedef kuyruğun bu eşgörünümünü barındıran kuyruk yöneticisinin kuyruk yöneticisi özniteliğidir.

InhibitPut (MQLONG) -giriş

MQWQR yapısıyla açıklanan hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinde tanımlanan koyma engeli kuyruk özniteliği. Aşağıdaki değerler kullanılabilir:

MQQA_PUT_INHIBITED

Koyma işlemleri engellenmiş.

MQQA_PUT_ALLOWED

Koyma işlemlerine izin verilir.

CLWLQueuePriority (MQLONG) -giriş

MQWQR yapısının tanımladığı hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinde tanımlanan küme iş yükü kuyruğu önceliği özniteliği.

CLWLQueueRank (MQLONG) -giriş

MQWQR yapısının tanımladığı hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinde tanımlanan küme iş yükü kuyruğu sırası.

DefPutResponse (MQLONG) -giriş

MQWQR yapısının tanımladığı hedef kuyruk eşgörünümünü barındıran kuyruk yöneticisinde tanımlanan varsayılan koyma yanıt kuyruğu özniteliği. Aşağıdaki değerler kullanılabilir:

MQPRT_SYNC_RESPONSE

MQPUT ya da MQPUT1 çağrılarına zamanuyumlu yanıt.

MQPRT_ASYNC_RESPONSE

MQPUT ya da MQPUT1 çağrılarına zamanuyumsuz yanıt.

İlgili başvurular

MQWQR için ilk değerler ve dil bildirimleri

MQWQR -Küme iş yükü kuyruğu kaydı için ilk değerler ve C ve High Level Assembler Dil bildirimleri.

MQWQR için ilk değerler ve dil bildirimleri

MQWQR -Küme iş yükü kuyruğu kaydı için ilk değerler ve C ve High Level Assembler Dil bildirimleri.

<i>Çizelge 832. MQWQR içindeki alanlar</i>		
Alan adı	Değişmezin adı	Değişmezin değeri
<i>StrucId</i>	MQWQR_STRUC_ID_ARRAY	'WQR→'
<i>Version</i>	MQWQR_VERSION_1	1
<i>StrucLength</i>	MQWQR_CURRENT_LENGTH ³	212
<i>QFlags</i>	Yok	0
<i>QName</i>	Yok	" "
<i>QMgrIdentifier</i>	Yok	" "
<i>ClusterRecOffset</i>	Yok	0
<i>QType</i>	Yok	0
<i>QDesc</i>	Yok	" "
<i>DefBind</i>	Yok	0
<i>DefPersistence</i>	Yok	0
<i>DefPriority</i>	Yok	0
<i>InhibitPut</i>	Yok	0

Çizelge 832. MQWQR içindeki alanlar (devamı var)

Alan adı	Değişmezin adı	Değişmezin değeri
CLWLQueuePriority	Yok	0
CLWLQueueRank	Yok	0
DefPutResponse	Yok	1

Notlar:

1. – simgesi, tek bir boş karakteri temsil eder.
2. C programlama dilinde, MQWQR_DEFAULT makro değişkeni varsayılan değerleri içerir. Yapıdaki alanlar için başlangıç değerleri sağlamak üzere bunu aşağıdaki şekilde kullanın:

```
MQWQR MyWQR = {MQWQR_DEFAULT};
```

3. İlk değerler, yapının uzunluğunu bilerek yapının sürüm 1 'e değil, geçerli sürümün uzunluğuna ayarlar.

C bildirimi

```
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)
```

İlgili başvurular

[MQWQR ' deki alanlar-Küme iş yükü kuyruğu kayıt yapısı](#)

[MQWQR -Küme iş yükü kuyruğu kayıt yapısındaki alanların açıklaması.](#)

MQWCR -Küme iş yükü kümesi kayıt yapısı

Aşağıdaki çizelge, MQWCR kümesi iş yükü kayıt yapısındaki alanları özetler.

Çizelge 833. MQWCR içindeki alanlar		
Alan	Açıklama	Sayfa
<i>ClusterName</i>	Kümenin adı	ClusterName
<i>ClusterRecOffset</i>	Sonraki küme kaydının görel konumu (MQWCR)	ClusterRecGörel Konumu
<i>ClusterFlags</i>	Küme işaretleri	ClusterFlags

Küme iş yükü kümesi kayıt yapısı, bir kümeye ilişkin bilgiler içerir. Hedef kuyruğun ait olduğu her küme için bir küme iş yükü kümesi kayıt yapısı vardır.

Küme iş yükü kümesi kayıt yapısı tüm ortamlarda desteklenir.

İlgili başvurular

[MQ_CLUSTER_WORKLOAD_EXIT -Arama açıklaması](#)

Bir iletiyi kullanılabilir bir kuyruk yöneticisine yöneltmek için kuyruk yöneticisi tarafından küme iş yükü çıkışı çağrılır.

[MQXCLWLN -Küme iş yükü kayıtlarında gezinme](#)

[MQXCLWLN çağırısı](#), küme önbelleğinde saklanan MQWDR, MQWQRve MQWCR kayıtları zincirlerinde dolaşmak için kullanılır.

[MQWXP -Küme iş yükü çıkış parametresi yapısı](#)

Aşağıdaki çizelge, MQWXP -Küme iş yükü çıkış parametresi yapısındaki alanları özetler.

[MQWDR-Küme iş yükü hedef kayıt yapısı](#)

Aşağıdaki çizelge, MQWDR -Küme iş yükü hedef kayıt yapısındaki alanları özetler.

[MQWQR -Küme iş yükü kuyruğu kayıt yapısı](#)

Aşağıdaki çizelge, MQWQR -Küme iş yükü kuyruğu kayıt yapısındaki alanları özetler.

MQWCR -Küme iş yükü kümesi kayıt yapısındaki alanlar.

MQWCR -Küme iş yükü kümesi kayıt yapısındaki alanların açıklaması.

ClusterName (MQCHAR48) -giriş

MQWCR yapısına sahip hedef kuyruk eşgörünümünün ait olduğu kümenin adı. Hedef kuyruk eşgörünümü bir MQWDR yapısı tarafından tanımlanır.

- ClusterName uzunluğu MQ_CLUSTER_NAME_LENGTH' dir.

ClusterRecOffset (MQLONG) -giriş

Sonraki MQWCR yapısının mantıksal görel konumu.

- Başka MQWCR yapısı yoksa, ClusterRecOffset sıfır olur.
- Görel konum, MQWCR yapısının başlangıcından itibaren bayt cinsinden ölçülür.

ClusterFlags (MQLONG) -giriş

Küme işaretleri, MQWCR yapısıyla tanımlan kuyruk yöneticisinin özelliklerini gösterir. Şu işaretler tanımlandı:

MQQMF_REPOSITORY_Q_MGR

Hedef, tam bir havuz kuyruğu yöneticisidir.

MQQMF_CLUSSDR_USER_DEFINED

Küme-gönderen kanalı el ile tanımlandı.

MQQMF_CLUSSDR_AUTO_DEFINED

Küme-gönderen kanalı otomatik olarak tanımlandı.

MQQMF_AVAILABLE

Hedef kuyruk yöneticisi ileti almak için kullanılabilir.

Diğer değerler

Alandaki diğer işaretler, iç amaçlar için kuyruk yöneticisi tarafından ayarlanabilir.

İlgili başvurular

[MQWCR için ilk değerler ve dil bildirimleri](#)

[MQWCR -Küme iş yükü kümesi kayıt yapısı için başlangıç değerleri ve C ve High Level Assembler Dil bildirimleri.](#)

MQWCR için ilk değerler ve dil bildirimleri

[MQWCR -Küme iş yükü kümesi kayıt yapısı için başlangıç değerleri ve C ve High Level Assembler Dil bildirimleri.](#)

Çizelge 834. MQWCR içindeki alanlar		
Alan adı	Değişmezin adı	Değişmezin değeri
<i>ClusterName</i>	Yok	" "
<i>ClusterRecOffset</i>	Yok	0
<i>ClusterFlags</i>	Yok	0

C bildirimi

```
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
MQWCR_ORG ORG MQWCR
MQWCR_AREA DS CL(MQWCR_LENGTH)
```

İlgili başvurular

[MQWCR -Küme iş yükü kümesi kayıt yapısındaki alanlar.](#)

[MQWCR -Küme iş yükü kümesi kayıt yapısındaki alanların açıklaması.](#)

API çıkış başvurusu

Bu bölümde, API çıkışlarını yazan bir programcı için esas olarak ilgilenilen başvuru bilgileri sağlanır.

Genel kullanım notları

notlar:

1. Tüm çıkış işlevleri MQXEP çağrısını verebilir; bu çağrı özellikle API çıkış işlevlerinden kullanılmak üzere tasarlanmıştır.
2. MQ_INIT_EXIT işlevi, MQXEP dışında herhangi bir MQ çağrısı yayınlayamaz.

3. Yürürlükteki bağlantı için MQDISC çağrısı yayınlayamazsınız.
4. Bir çıkış işlevi MQCONN çağrısını ya da MQCONNX çağrısını MQCNO_HANDLE_SHARE_NONE seçeneğiyle verirse, çağrı MQRC_ALREADY_CONNECTED neden koduyla tamamlanır ve döndürülen tanıtıcı, çıkışa değiştirge olarak geçirilen tanıtıcı ile aynıdır.
5. Genel olarak, bir API çıkış işlevi bir MQI çağrısı yayınladığında, API çıkışları özyineli olarak çağrılmaz. Ancak, bir çıkış işlevi MQCONNX çağrısını MQCNO_HANDLE_SHARE_BLOCK ya da MQCNO_HANDLE_SHARE_NO_BLOCK seçenekleriyle yaparsa, çağrı yeni bir paylaşılan tanıtıcı döndürür. Bu, çıkış takımına kendi bağlantı tanıtıcısı ve dolayısıyla uygulamanın iş biriminden bağımsız bir iş birimi sağlar. Çıkış takımı, bu tanıtıcıyı, iletileri kendi iş birimi içine koymak ve almak ve bu iş birimini kesinleştirmek ya da geri almak için kullanabilir; tüm bunlar, uygulamanın iş birimini herhangi bir şekilde etkilemeden yapılabilir.

Çıkış işlevi, uygulama tarafından kullanılan tanıtıcıdan farklı bir bağlantı tanıtıcısı kullandığından, çıkış işlevi tarafından verilen MQ çağrıları, ilgili API çıkış işlevlerinin çağrılmasına neden olur. Bu nedenle, çıkış işlevleri özyineli olarak çağrılabilir. MQAXP ve çıkış zinciri alanındaki *ExitUserArea* alanının bağlantı tanıtıcısı kapsamına sahip olduğunu unutmayın. Sonuç olarak, bir çıkış işlevi, özyineli olarak çağrılan başka bir örneğe zaten etkin olduğunu işaret etmek için bu alanları kullanamaz.

6. Çıkış işlevleri, uygulamanın iş birimine ileti yerleştirip alabilir. Uygulama iş birimini kesinleştirdiğinde ya da geri çevirdiğinde, iş birimi içindeki tüm iletiler, iş birimine kimin yerleştirdiğine (uygulama ya da çıkış işlevi) bakılmaksızın, kesinleştirilir ya da yedeklenir. Ancak çıkış, uygulamanın sistem sınırlarını daha önce aşmasına neden olabilir (örneğin, bir iş birimindeki kesinleştirilmemiş ileti sayısı üst sınırını aşarak).

Bir çıkış işlevi uygulamanın çalışma birimini bu şekilde kullandığında, çıkış işlevi genellikle MQCMIT çağrısını yayınlamaktan kaçınmalıdır; bu, uygulamanın çalışma birimini kesinleştirir ve uygulamanın doğru çalışmasını bozabilir. Ancak, çıkış işlevi kesinleştirilmekte olan iş birimini önleyen ciddi bir hatayla karşılaşır (örneğin, uygulamanın iş biriminin bir parçası olarak bir ileti konan bir hata), çıkış işlevinin bazen MQBACK çağrısına yanıt vermesi gerekebilir. MQBACK çağrıldığında, uygulama iş birimi sınırlarının değiştirilmediğinden emin olun. Bu durumda, uygulamanın iş biriminin geriletildiğini saptayabilmesi için, çıkış işlevinin, tamamlama kodu MQCC_WARNING ve neden kodu MQRC_BACKED_OUT ile uygulamaya döndürülmesini güvenceye almak için uygun değerleri ayarlaması gerekir.

Bir çıkış işlevi MQ çağrılarını yayınlamak için uygulamanın bağlantı tanıtıcısını kullanırsa, bu çağrılar API çıkış işlevlerinin daha fazla çağrılmasıyla sonuçlanmaz.

7. Bir MQXR_BEFORE çıkış işlevi olağandışı sona ererse, kuyruk yöneticisi hatadan kurtulabilir. Yapabiliyorsa, kuyruk yöneticisi, çıkış işlevi MQXCC_FAILED döndürmüş gibi işlemeye devam eder. Kuyruk yöneticisi kurtaramazsa, uygulama sonlandırılır.
8. Bir MQXR_AFTER çıkış işlevi olağandışı sona ererse, kuyruk yöneticisi hatadan kurtulabilir. Yapabiliyorsa, kuyruk yöneticisi, çıkış işlevi MQXCC_FAILED döndürmüş gibi işlemeye devam eder. Kuyruk yöneticisi kurtaramazsa, uygulama sonlandırılır. İkinci durumda, bir iş birimi dışında alınan iletilerin kaybolduğunu unutmayın (bu, kuyruktan bir ileti kaldırıldıktan hemen sonra başarısız olan uygulamayla aynı durumdur).
9. MCA işlemi iki aşamalı bir kesinleştirme gerçekleştirir.

Bir API çıkışı, hazırlanmış bir MCA sürecinden bir MQCMIT ' yi algılar ve iş birimi içinde bir işlem gerçekleştirme girişiminde bulunursa, işlem MQRC_UOW_NOT_KULLANILABILIR neden koduyla başarısız olur.

10. Birden çok IBM MQ kurulumu varsa, sonraki sürüme eklenen yeni işlevsellik önceki sürümlerle çalışmayabileceğinden, daha önceki bir IBM MQ sürümü için yazılan çıkışları kullanın. Yayınlar arasındaki değişikliklerle ilgili daha fazla bilgi için bkz. [IBM MQ 8.0 içinde değişiklikler](#).

IBM MQ API çıkış değiştirgesi yapısı (MQAXP)

Dış denetim öbeği olan MQAXP yapısı, API çıkışına giriş ya da çıkış değiştirgesi olarak kullanılır. Bu konuda, kuyruk yöneticilerinin çıkış işlevlerini nasıl işlediğine ilişkin bilgiler de verilir.

MQAXP ' de şu C bildirimi var:

```

typedef struct tagMQAXP {
MQCHAR4   StructId;           /* 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 */
};

```

Bir API çıkışındaki işlemler çağrıldığında aşağıdaki parametre listesi geçirilir:

StrucId (MQCHAR4)-giriş

Değeri şu olan çıkış değıştirgesi yapısı tanıtıcısı:

```
MQAXP_STRUC_ID.
```

Çıkış işleyicisi, girişte bu alanı her çıkış işlevine ayarlar.

Sürüm (MQLONG)-giriş

Şu değere sahip yapı sürümü numarası:

MQAXP_VERSION_1

Sürüm 1 API çıkış parametresi yapısı.

MQAXP_VERSION_2

Sürüm 2 API çıkış parametresi yapısı.

MQAXP_CURRENT_VERSION

API çıkış parametresi yapısı için geçerli sürüm numarası.

Çıkış işleyicisi, girişte bu alanı her çıkış işlevine ayarlar.

ExitId (MQLONG)-giriş

Girişte çıkış yordamına ayarlanan çıkış tanıtıcısı, çıkış tipini gösterir:

MQXT_API_EXIT

API çıkışı.

ExitReason (MQLONG)-giriş

Çıkışın çağrılmasının nedeni, her bir çıkış işlevine ilişkin girişte ayarlanır:

MQXR_CONNECTION

Çıkış, bir MQCONN ya da MQCONNX çağrısından önce kendisini kullanıma hazırlamak ya da bir MQDISC çağrısından sonra kendisini sona erdirmek için çağrılıyor.

MQXR_BEFORE

Bir API çağrısı yürütülmeden ya da MQGET üzerinde veri dönüştürülmeden önce çıkış çağrılıyor.

MQXR_AFTER

Bir API çağrısı yürütüldükten sonra çıkış çağrılıyor.

ExitResponse (MQLONG)-çıkış

Çıkıştan gelen yanıt, her çıkış işlevine girişte kullanıma hazırlandı:

MQXCC_OK

Normal şekilde devam edin.

Çıkış işlevinin yürütülmesi sonucunda kuyruk yöneticisiyle iletişim kurmak için bu alan çıkış işleviyle ayarlanmalıdır. Değer aşağıdakilerden biri olmalıdır:

MQXCC_OK

Çıkış işlevi başarıyla tamamlandı. Normal şekilde devam edin.

Bu değer, tüm MQXR_* çıkış işlevleriyle ayarlanabilir. ExitResponse2 , zincirin daha sonra çıkış işlevlerinin çağrılıp çağrılmayacağına karar vermek için kullanılır.

MQXCC_FAILED

Bir hata nedeniyle çıkış işlevi başarısız oldu.

Bu değer, tüm MQXR_* çıkış işlevleriyle ayarlanabilir. Kuyruk yöneticisi CompCode ögesini MQCC_FAILED olarak ayarlıyor ve Nedeni:

- İşlev MQ_INIT_EXIT ise MQRC_API_EXIT_INIT_ERROR
- İşlev MQ_TERM_EXIT ise, MQRC_API_EXIT_TERM_ERROR
- Diğer tüm çıkış işlevleri için MQRC_API_EXIT_ERROR

Ayarlanan değerler, zincirin sonraki bir çıkış işleviyle değiştirilebilir.

ExitResponse2 yoksayılır; kuyruk yöneticisi, MQXR2_SUPPRESS_CHAIN döndürülmüş gibi işlemeye devam eder.

MQXCC_SUPPRESS_FUNCTION

IBM MQ API işlevini engelle.

Bu değer yalnızca bir MQXR_BEFORE çıkış işleviyle ayarlanabilir. API çağrısını atlar. MQ_DATA_CONV_ON_GET_EXIT tarafından döndürülürse, veri dönüştürme atlanır. Kuyruk yöneticisi CompCode değerini MQCC_FAILED olarak ve Reason to MQRC_SUPPRESSED_BY_EXIT olarak ayarlar, ancak ayarlanan değerler zincirin sonraki bir çıkış işleviyle değiştirilebilir. Çağrıya ilişkin diğer parametreler, çıkışın bıraktığı gibi kalır. ExitResponse2 , zincirin daha sonra çıkış işlevlerinin çağrılıp çağrılmayacağına karar vermek için kullanılır.

Bu değer bir MQXR_AFTER ya da MQXR_CONNECTION çıkış işleviyle ayarlanırsa, kuyruk yöneticisi MQXCC_FAILED döndürülmüş gibi işlemeye devam eder.

MQXCC_SKIP_FUNCTION

IBM MQ API işlevini atlayın.

Bu değer yalnızca bir MQXR_BEFORE çıkış işleviyle ayarlanabilir. API çağrısını atlar. MQ_DATA_CONV_ON_GET_EXIT tarafından döndürülürse, veri dönüştürme atlanır. Çıkış işlevi CompCode ve Reason değerlerini uygulamaya döndürülecek değerlere ayarlamalıdır; ancak, ayarlanan değerler zincirin sonraki bir çıkış işleviyle değiştirilebilir. Çağrıya ilişkin diğer parametreler, çıkışın bıraktığı gibi kalır. ExitResponse2 , zincirin daha sonra çıkış işlevlerinin çağrılıp çağrılmayacağına karar vermek için kullanılır.

Bu değer bir MQXR_AFTER ya da MQXR_CONNECTION çıkış işleviyle ayarlanırsa, kuyruk yöneticisi MQXCC_FAILED döndürülmüş gibi işlemeye devam eder.

MQXCC_SUPPRESS_EXIT

Çıkış kümesine ait tüm çıkış işlevlerini engelleyin.

Bu değer yalnızca MQXR_BEFORE ve MQXR_AFTER çıkış işlevleriyle ayarlanabilir. Bu mantıksal bağlantı için bu çıkış kümesine ait çıkış işlevlerinin sonraki tüm çağrılarını atlar. MQXR_CONNECTION ExitReason ile MQ_TERM_EXIT işlevi çağrıldığında, mantıksal bağlantı kesme isteği oluşuncaya kadar bu atlama devam eder.

Çıkış işlevi CompCode ve Reason değerlerini uygulamaya döndürülecek değerlere ayarlamalıdır; ancak, ayarlanan değerler zincirin sonraki bir çıkış işleviyle değiştirilebilir. Çağrıya ilişkin diğer parametreler, çıkışın bıraktığı gibi kalır. ExitResponse2 yoksayıldı.

Bu değer bir MQXR_CONNECTION çıkış işleviyle ayarlanırsa, kuyruk yöneticisi MQXCC_FAILED döndürülmüş gibi işlemeye devam eder.

ExitResponse ile ExitResponse2 arasındaki etkileşim ve bunun çıkış işleme üzerindeki etkisi hakkında bilgi için bkz. [“Kuyruk yöneticileri çıkış işlevlerini nasıl işler?” sayfa 1539.](#)

ExitResponse2 (MQLONG)-çıkış

Bu, MQXR_BEFORE çıkış işlevleri için birincil çıkış yanıt kodunu niteleyen ikincil bir çıkış yanıt kodudur. Şu şekilde kullanıma hazırlandı:

```
MQXR2_DEFAULT_CONTINUATION
```

bir IBM MQ API çağrı çıkışı işlevine girişte. Daha sonra aşağıdaki değerlerden birine ayarlanabilir:

MQXR2_DEFAULT_CONTINUATION

ExitResponse(Çıkış Yanıtı) değerine bağlı olarak, zincirdeki sonraki çıkışa devam edilip edilmeyeceğini belirler.

ExitResponse MQXCC_SUPPRESS_FUNCTION ya da MQXCC_SKIP_FUNCTION ise, daha sonra MQXR_BEFORE zincirinde çıkış işlevlerini ve MQXR_AFTER zincirinde eşleşen çıkış işlevlerini atlayın. MQXR_BEFORE zincirinin önceki çıkış işlevleriyle eşleşen MQXR_AFTER zincirindeki çıkış işlevlerini çağırın.

Tersi durumda, zincirdeki sonraki çıkışı çağırın.

MQXR2_SUPPRESS_CHAIN

Zinciri bastırın.

Daha sonra MQXR_BEFORE zincirinde çıkış işlevlerini ve bu API çağrısını çağırarak için MQXR_AFTER zincirinde eşleşen çıkış işlevlerini atlayın. MQXR_BEFORE zincirinin önceki çıkış işlevleriyle eşleşen MQXR_AFTER zincirindeki çıkış işlevlerini çağırın.

MQXR2_CONTINUE_CHAIN

Zincirdeki bir sonraki çıkışla devam edin.

ExitResponse ile ExitResponse2 arasındaki etkileşim ve bunun çıkış işleme üzerindeki etkisi hakkında bilgi için bkz. [“Kuyruk yöneticileri çıkış işlevlerini nasıl işler?” sayfa 1539.](#)

Geribildirim (MQLONG)-giriş/çıkış

Çıkış işlevi çağrıları arasında geribildirim kodlarını iletiyor. Bu, aşağıdakiler için kullanıma hazırlanır:

```
MQFB_NONE (0)
```

zincirdeki ilk çıkışın ilk işlevini çağırılmadan önce.

Çıkışlar bu alanı, geçerli bir MQFB_* ya da MQRC_* değeri de içinde olmak üzere herhangi bir değere ayarlayabilir. Çıkışlar bu alanı MQFB_APPL_FIRST-MQFB_APPL_LAST aralığında kullanıcı tanımlı bir geribildirim değerine de ayarlayabilir.

APICallerType (MQLONG)-giriş

IBM MQ API çağrısının kuyruk yöneticisi için dış mı, yoksa iç mi olduğunu belirten API çağrı tipi: MQXACT_EXTERNAL ya da MQXACT_INTERNAL.

ExitUserAlanı (MQBYTE16)-giriş/çıkış

Belirli bir ExitInfoNesnesi ile ilişkili tüm çıkışlar için kullanılabilir bir kullanıcı alanı. hconn için ilk çıkış işlevini (MQ_INIT_EXIT) çağırılmadan önce MQXUA_NONE (ExitUserAlanı uzunluğu için ikili sınırlar) olarak başlatılır. Bundan sonra, bir çıkış işlevi tarafından bu alanda yapılan değişiklikler, aynı çıkışın işlevlerinin çağrılmasıyla korunur.

Bu alan, 4 MQLONG ' un bir katına hizalanır.

Çıkışlar, bu bölgeden ayırdıkları herhangi bir depolamayı da tutturabilir.

Her bir hconn için, bir çıkış zincirindeki her çıkışın farklı bir ExitUserAlanı vardır. ExitUserAlanı bir zincirdeki çıkışlar tarafından paylaşılamaz ve bir çıkışa ilişkin ExitUserAlanının içeriği bir zincirdeki başka bir çıkışın kullanımına sunulmaz.

C programları için, MQXUA_NONE_ARRAY değişmezi MQXUA_NONE ile aynı değerle, ancak dizgi yerine bir karakter dizisi olarak tanımlanır.

Bu alanın uzunluğu MQ_EXIT_USER_AREA_LENGTH tarafından verilir.

ExitData (MQCHAR32)-giriş

Çıkış verileri, girişte her çıkış işlemine ilişkin olarak, çıkışta sağlanan çıkışa özgü verilerin 32 karakterine ayarlanır. Çıkışta değer tanımlamazsanız, bu alan tamamen boşluktur.

Bu alanın uzunluğu MQ_EXIT_DATA_LENGTH tarafından verilir.

ExitInfoAd (MQCHAR48)-giriş

Girişte her çıkış işlemine ilişkin olarak, stanzas 'ta çıkış tanımlamalarında belirtilen ApiExit_name değerine ayarlanan çıkış bilgisi adı.

ExitPDArea (MQBYTE48)-giriş/çıkış

Bir çıkış işlevinin her çağrılışında MQXPDA_NONE (alanın uzunluğu için ikili sıfırlar) olarak kullanıma hazırlanan bir sorun saptama alanı.

C programları için, MQXPDA_NONE_ARRAY değişmezi MQXPDA_NONE ile aynı değerle, ancak dizgi yerine bir karakter dizisi olarak tanımlanır.

Çıkış işleyicisi, işlem başarılı olsa bile, bu alanı her zaman çıkışın sonundaki IBM MQ izlemesine yazar.

Bu alanın uzunluğu MQ_EXIT_PD_AREA_LENGTH tarafından verilir.

QMGrName (MQCHAR48)-giriş

Bir IBM MQ API çağrısının işlenmesi sonucunda bir çıkış çağırana, uygulamanın bağlı olduğu kuyruk yöneticisinin adı.

MQCONN ya da MQCONNX çağrılarında belirtilen bir kuyruk yöneticisinin adı boşsa, bu alan, uygulamanın sunucu ya da istemci olması fark etmez, uygulamanın bağlı olduğu kuyruk yöneticisinin adına ayarlanır.

Çıkış işleyicisi, girişte bu alanı her çıkış işlemine ayarlar.

Bu alanın uzunluğu MQ_Q_MGR_NAME_LENGTH tarafından verilir.

ExitChainAreaPtr (PMQACH)-giriş/çıkış

Bu, bir zincirdeki farklı çıkışların çağrılması boyunca verileri iletmek için kullanılır. Bir çıkış zincirindeki ilk çıkışın ilk işlevi (ExitReason MQXR_CONNECTION ile MQ_INIT_EXIT) çağrılmadan önce boş değerli (NULL) göstergeye ayarlanır. Bir çağrıdaki çıkışın döndürdüğü değer, sonraki çağrıya geçirilir.

Çıkış zinciri alanının nasıl kullanılacağına ilişkin daha fazla ayrıntı için bkz. "[Çıkış zinciri alanı ve çıkış zinciri alanı üstbilgisi \(MQACH\)](#)" sayfa 1542 .

Hconfig (MQHCONFIG)-giriş

Kullanıma hazırlanmakta olan işlem kümesini gösteren yapılandırma tanıtıcısı. Bu değer, MQ_INIT_EXIT işlevinde kuyruk yöneticisi tarafından oluşturulur ve daha sonra API çıkış işlemine geçirilir. Her çıkış işlemine giriş için ayarlanır.

MQI ve DCI çağrıları yapmak için MQIEP yapısına ilişkin gösterge olarak Hconfig 'i kullanabilirsiniz. HConfig değıştirgesini MQIEP yapısına gösterge olarak kullanmadan önce, HConfig parametresinin ilk 4 byte 'ının MQIEP yapısının StrucId ile eşleştiğini denetlemeniz gerekir.

İşlev (MQLONG)-giriş

"[Dış sabitler](#)" sayfa 1543 içinde açıklanan MQXF_ * değışmezleri olan geçerli değerler olan işlem tanıtıcısı.

Çıkış işleyicisi, çıkışın çağrılmasına neden olan IBM MQ API çağrısına bağlı olarak, her çıkış işlemine giriş için bu alanı doğru değere ayarlar.

ExitMsgTanıtıcısı (MQHMSG)-giriş/çıkış

İşlev MQXF_GET ve ExitReason MQXR_AFTER ise, bu alanda, API çıkışını kaydederken MQXEPO yapısında belirtilen ExitProperties dizgisiyle eşleşen diğer özelliklere ve ileti tanımlayıcı alanlarına API çıkışına izin veren geçerli bir ileti tanıtıcısı döndürülür.

ExitMsgtanıtıcısında döndürülen ileti dışı tanımlayıcı özellikleri, belirtildiyse, MQGMO yapısındaki MsgHandle içinde ya da ileti verilerinde bulunmaz.

İşlev MQXF_GET ve ExitReason MQXR_BEFORE ise, çıkış programı bu alanı MQHM_NONE olarak ayarlarsa, ExitMsgHandle özelliklerinin doldurulmasını engeller.

Sürüm MQAXP_VERSION_2' den küçükse bu alan ayarlanmaz.

Kuyruk yöneticileri çıkış işlevlerini nasıl işler?

Bir çıkış işlevinden dönüş sırasında kuyruk yöneticisi tarafından gerçekleştirilen işleme hem ExitResponse , hem de ExitResponse2' ye bağlıdır.

Çizelge 835 sayfa 1539 içinde, bir MQXR_BEFORE çıkış işlevine ilişkin olası birleşimler ve bunların etkileri özetlenerek şunları gösterir:

- API çağrısının CompCode ve Neden parametrelerini ayarlayan kişi
- MQXR_BEFORE zincirinde kalan çıkış işlevlerinin ve MQXR_AFTER zincirinde eşleşen çıkış işlevlerinin çağrılıp çağrılmayacağını belirler
- API çağrılıp çağrılmayacağını belirler

MQXR_AFTER çıkış işlevi için:

- CompCode ve Reason MQXR_BEFORE ile aynı şekilde ayarlanır
- ExitResponse2 yoksayılr (MQXR_AFTER zincirindeki geri kalan çıkış işlevleri her zaman çağrılır)
- MQXCC_SUPPRESS_FUNCTION ve MQXCC_SKIP_FUNCTION geçerli değil

MQXR_CONNECTION çıkış işlevi için:

- CompCode ve Reason MQXR_BEFORE ile aynı şekilde ayarlanır
- ExitResponse2 yoksayıldı
- MQXCC_SUPPRESS_FUNCTION, MQXCC_SKIP_FUNCTION, MQXCC_SUPPRESS_EXIT geçerli değil

Her durumda, bir çıkışın ya da kuyruk yöneticisinin CompCode ve Reason ayarını belirlediği durumlarda, ayarlanan değerler daha sonra çağrılan bir çıkışla ya da API çağrısıyla değiştirilebilir (API çağrısı daha sonra çağrılırsa).

Çizelge 835. MQXR_BEFORE çıkış işleme			
ExitResponse değeri	CompCode ve Reason set by	ExitResponse2 (varsayılan devam) Zinciri değeri	ExitResponse2 API değeri (varsayılan devam)
MQXCC_OK	exit	Y	Y
MQXCC_SUPPRESS_EXIT	exit	Y	Y
MQXCC_SUPPRESS_FUNCTION	Kuyruk Yöneticisi	N	N
MQXCC_SKIP İşLEVI	exit	N	N
MQXCC_FAILED	Kuyruk Yöneticisi	N	N

İstemciler çıkış işlevlerini nasıl işler?

Genel olarak, istemciler çıkış işlevlerini sunucu uygulamalarının işlediği gibi işlerler ve bu yapıdaki *QMgrName* özniteliği, işlevin bir sunucuda ya da istemcide olması fark etmez.

Ancak, istemcinin *mqs.ini* dosyası kavramı olmadığından *ApiExitCommon* ve *APIExitTemplate* stanzas geçerli değildir. Yalnızca *ApiExitLocal* kısmı geçerlidir ve bu parça *mqclient.ini* dosyasında yapılandırılır.

IBM MQ API çıkış bağlamı yapısı (MQAXC)

Dış denetim öbeği olan MQAXC yapısı, bir API çıkışına giriş değiştirgesi olarak kullanılır.

MQAXC aşağıdaki C bildirimine sahip:

```

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 */
};

```

MQAXC ' ye ilişkin değıştirgeler şunlardır:

StrucId (MQCHAR4)-giriş

MQAXC_STRUC_IDdeğerine sahip çıkış baęlamı yapısı tanıtıcısı. C programları için, MQAXC_STRUC_ID_ARRAY değışmezi de MQAXC_STRUC_ID ile aynı değerle, ancak dizgi yerine bir karakter dizisi olarak tanımlanır.

Çıkış işleyicisi, girişte bu alanı her çıkış işlevine ayarlar.

Sürüm (MQLONG)-giriş

Şu değere sahip yapı sürümü numarası:

MQAXC_VERSION_2

Çıkış baęlamı yapısına ilişkin sürüm numarası.

MQAXC_CURRENT_VERSION

Çıkış baęlamı yapısına ilişkin yürürlükteki sürüm numarası.

Çıkış işleyicisi, girişte bu alanı her çıkış işlevine ayarlar.

Ortam (MQLONG)-giriş

Çıkış işlevinin yönlendirilmesiyle sonuçlanan bir IBM MQ API çağrısını yayınlayan ortam. Bu alan için geçerli değerler şunlardır:

MQXE_DIĞER

Bu değer, bir sunucu uygulamasından çıkışın çağrılması durumunda bir API çıkışının gördüğü çağrılarla tutarlıdır. Bu, bir API çıkışının istemcide değışmeden çalıştığı ve farklı bir şey görmediği anlamına gelir.

Çıkışın istemcide çalışıp çalışmadığını gerçekten belirlemesi gerekiyorsa, çıkış bunu *ChannelName* ve *ChannelDefinition* alanlarına bakarak yapabilir.

MQXE_MCA

İleti kanalı aracısı

MQXE_MCA_SVRCONN

Bir istemci adına hareket eden bir ileti kanalı aracısı

MQXE_COMMAND_SERVER

Komut sunucusu

MQXE_MQSC

runmqsc komutu yorumlayıcısı

Çıkış işleyicisi, girişte bu alanı her çıkış işlevine ayarlar.

UserId (MQCHAR12)-giriş

Uygulamayla ilişkili kullanıcı kimliği. Özellikle, istemci baęlantılarında bu alan, kanal kodunun çalıştırıldığı kullanıcı kimliğinin tersine, benimsenen kullanıcının kullanıcı kimliğini içerir. İstemciden

boş bir kullanıcı kimliği akarsa, kullanılmakta olan kullanıcı kimliği için herhangi bir değişiklik yapılmaz. Yani, yeni bir kullanıcı kimliği benimsenmez.

Çıkış işleyicisi, girişte bu alanı her çıkış işlevine ayarlar. Bu alanın uzunluğu MQ_USER_ID_LENGTH tarafından verilir.

Bir istemci söz konusu olduğunda, bu, istemciden sunucuya gönderilen kullanıcı kimliğidir. Kullanıcı kimliğini değiştiren bir MCAUser ya da CHLAUTH yapılandırması olabileceği için, kuyruk yöneticisinde istemcinin çalıştırdığı etkin kullanıcı kimliği bu olmayabilir.

SecurityId (MQBYTE40)-giriş

Uygulamayı çalıştıran kullanıcı kimliğinin uzantısı. Uzunluğu MQ_SECURITY_ID_LENGTH tarafından verilir.

Bir istemci söz konusu olduğunda, bu, istemciden sunucuya gönderilen kullanıcı kimliğidir. Kullanıcı kimliğini değiştiren bir MCAUser ya da CHLAUTH yapılandırması olabileceği için, kuyruk yöneticisinde istemcinin çalıştırdığı etkin kullanıcı kimliği bu olmayabilir.

ConnectionName (MQCHAR264)-giriş

Bağlantı adı alanı, istemcinin adresine ayarlanır. Örneğin, TCP/IP için bu, istemci IP adresi olur.

Bu alanın uzunluğu MQ_CONN_NAME_LENGTH tarafından verilir.

Bir istemci söz konusu olduğunda, bu, kuyruk yöneticisinin ortak adresidir.

LongMCAUserIdLength (MQLONG)-giriş

Uzun MCA kullanıcı kimliğinin uzunluğu.

MCA kuyruk yöneticisine bağlandığında, bu alan uzun MCA kullanıcı kimliğinin uzunluğuna ayarlanır (ya da böyle bir tanıtıcı yoksa sıfır olur).

Bir istemci söz konusu olduğunda, bu, istemci uzun kullanıcı kimliğidir.

LongRemoteUserIdLength (MQLONG)-giriş

Uzun uzak kullanıcı kimliğinin uzunluğu.

MCA kuyruk yöneticisine bağlandığında, bu alan uzun uzak kullanıcı kimliğinin uzunluğuna ayarlanır. Aksi takdirde bu alan sıfır olarak ayarlanır.

Bir istemci söz konusu olduğunda, bu alanı sıfır olarak ayarlayın.

LongMCAUserIdPtr (MQPTR)-giriş

Uzun MCA kullanıcı kimliğinin adresi.

MCA kuyruk yöneticisine bağlandığında, bu alan uzun MCA kullanıcı kimliğinin adresine (ya da böyle bir tanıtıcı yoksa boş değerli bir göstergeye) ayarlanır.

Bir istemci söz konusu olduğunda, bu, istemci uzun kullanıcı kimliğidir.

LongRemoteUserIdPtr (MQPTR)-giriş

Uzun uzak kullanıcı kimliğinin adresi.

MCA kuyruk yöneticisine bağlandığında, bu alan uzun uzak kullanıcı kimliğinin adresine (ya da böyle bir tanıtıcı yoksa boş değerli bir göstergeye) ayarlanır.

Bir istemci söz konusu olduğunda, bu alanı sıfır olarak ayarlayın.

ApplName (MQCHAR28)-giriş

IBM MQ API çağrısını yayınlayan uygulamanın ya da bileşenin adı.

ApplName oluşturulmasına ilişkin kurallar, bir MQPUT için varsayılan ad oluşturulmasına ilişkin kurallarla aynıdır.

Bu alanın değeri, işletim sisteminin program adı sorgulanarak bulunur. Uzunluğu MQ_APPL_NAME_LENGTH tarafından verilir.

ApplType (MQLONG)-giriş

IBM MQ API çağrısını yayınlayan uygulamanın ya da bileşenin tipi.

Değer, uygulamanın derlendiği platform için MQAT_DEFAULT ya da tanımlı MQAT_* değerlerinden birine denk geliyor.

Çıkış işleyicisi, girişte bu alanı her çıkış işlevine ayarlar.

ProcessId (MQPID)-giriş

İşletim sistemi işlem tanıtıcısı.

Varsa, çıkış işleyicisi bu alanı her çıkış işlevine giriş için ayarlar.

ThreadId (MQTID)-giriş

MQ iş parçacığı tanıtıcısı. Bu, MQ izleme ve FFST dökümlerinde kullanılan tanıtıcıyla aynıdır, ancak işletim sistemi iş parçacığı tanıtıcısından farklı olabilir.

Varsa, çıkış işleyicisi bu alanı her çıkış işlevine giriş için ayarlar.

ChannelName (MQCHAR)-giriş

Varsa ve biliniyorsa, boşluklarla doldurarak kanalın adı.

Uygulanamıyorsa, bu alan NULL (boş değerli) karakterlere ayarlanır.

Reserved1 (MQBYTE4)-giriş

Bu alan ayrılmış.

ChanneDefinition (PMQCD)-giriş

Varsa ve biliniyorsa, kullanılmakta olan kanal tanımına ilişkin gösterge.

Uygulanamıyorsa, bu alan NULL (boş değerli) karakterlere ayarlanır.

İşaretçinin yalnızca bağlantı bir IBM MQ kanalı adına işleniyorsa ve kanal tanımı okunduysa tamamlandığını unutmayın.

Özellikle, kanal için ilk MQCONN çağrısı yapıldığında kanal tanımlaması sunucuda verilmez. Ayrıca, işaretçi doldurulursa, işaretçi tarafından gösterilen yapı (ve herhangi bir alt yapı) salt okunur olarak değerlendirilmelidir; yapının herhangi bir güncellenmesi öngörülemez sonuçlara yol açar ve desteklenmez.

Bir istemci söz konusu olduğunda, istemci için değer belirtilmemiş alanlar dışındaki alanlar, bir istemci uygulaması için uygun değerleri içerir.

Çıkış zinciri alanı ve çıkış zinciri alanı üstbilgisi (MQACH)

Gerekliyse, çıkış işlevi bir çıkış zinciri alanı için depolama alabilir ve MQAXP içindeki ExitChainAreaPtr değerini bu depolamayı gösterecek şekilde ayarlayabilir.

Çıkışlar (aynı ya da farklı çıkış işlevleri) birden çok çıkış zinciri alanı edinebilir ve bunları birbirine bağlayabilir. Çıkış zinciri alanları, çıkış işleyicisinden çağrıldığında yalnızca bu listeden eklenmeli ya da kaldırılmalıdır. Bu, farklı iş parçacıklarının aynı anda listeye alan eklemesinden ya da listeden alan kaldırmasından kaynaklanan bir diziselleştirme sorunu olmamasını sağlar.

Bir çıkış zinciri alanı, C bildirim olan bir MQACH üstbilgi yapısıyla başlamalıdır:

```
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 */
};
```

Çıkış zinciri alanı üstbilgisindeki alanlar şunlardır:

StrucId (MQCHAR4)-giriş

MQACH_STRUC_ID 'nin MQACH_DEFAULT tarafından tanımlanan ilk değeri olan çıkış zinciri alanı yapı tanıtıcısı.

C programları için, MQACH_STRUC_ID_ARRAY değişmezi de tanımlandı; bu değer MQACH_STRUC_ID ile aynı, ancak dizgi yerine bir karakter dizisi olarak.

Sürüm (MQLONG)-giriş

Yapı sürümü numarası:

MQACH_VERSION_1

Çıkış değiştirgeci yapısının sürüm numarası.

MQACH_CURRENT_VERSION

Çıkış bağlamı yapısının yürürlükteki sürüm numarası.

MQACH_DEFAULT tarafından tanımlanan bu alanın ilk değeri şudur: MQACH_CURRENT_VERSION.

Not: Bu yapının yeni bir sürümünü tanıtırarsanız, var olan parçanın düzeni değişmez. Çıkış işlevleri, sürüm numarasının, çıkış işlevinin kullanması gereken alanları içeren en düşük sürüme eşit ya da bu sürümden büyük olup olmadığını denetlemelidir.

StrucLength (MQLONG)-giriş

MQACH yapısının uzunluğu. Çıkışlar, çıkış verilerinin başlangıcını saptamak ve çıkış tarafından yaratılan yapının uzunluğuna ayarlamak için bu alanı kullanabilir.

MQACH_DEFAULT tarafından tanımlanan bu alanın ilk değeri MQACH_CURRENT_LENGTH.

ChainAreaUzunluğu (MQLONG)-giriş

Çıkış zinciri alanı uzunluğu, MQACH üstbilgisi de içinde olmak üzere, yürürlükteki çıkış zinciri alanının genel uzunluğuna ayarlanır.

MQACH_DEFAULT tarafından tanımlanan bu alanın ilk değeri sıfırdır.

ExitInfoAd (MQCHAR48)-giriş

Çıkış bilgisi adı.

Bir çıkış bir MQACH yapısı yarattığında, bu alanı kendi ExitInfoadıyla kullanıma hazırlamalıdır; böylece, daha sonra bu MQACH yapısı bu çıkışın başka bir eşgörünümü tarafından ya da birlikte çalışan bir çıkış tarafından bulunabilir.

MQACH_DEFAULT tarafından tanımlanan bu alanın ilk değeri sıfır uzunluklu bir dizedir ({}).

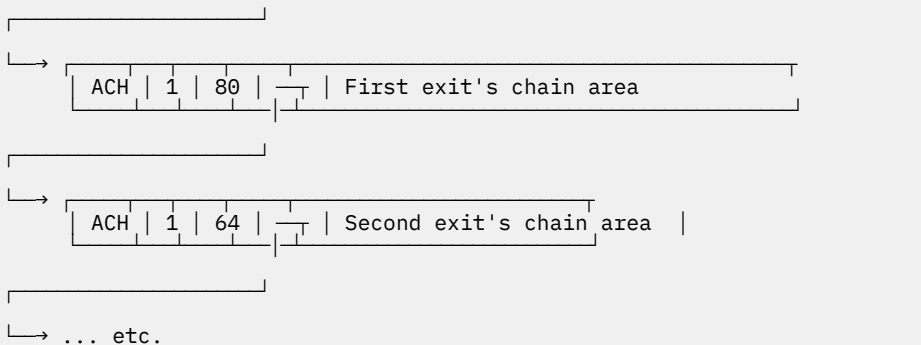
NextChainAreaPtr (PMQACH)-giriş

Boş değerli gösterge (NULL) için, MQACH_DEFAULT tarafından tanımlanan ilk değeri olan bir sonraki çıkış zinciri alanına ilişkin gösterge.

Çıkış işlevleri, elde ettikleri herhangi bir çıkış zinciri alanı için depolamayı serbest bırakmalı ve zincir işaretçilerini, çıkış zinciri alanlarını listeden kaldırmak için kullanmalıdır.

Bir çıkış zinciri alanı aşağıdaki gibi oluşturulabilir:

MQAXP.ExitChainAreaPtr —┐



Dış sabitler

API için kullanılacak dış değişmezlere ilişkin başvuru bilgileri var olduğundan bu konuyu kullanın.

API çıkışları için aşağıdaki dış sabitler kullanılabilir:

MQXF_ * (çıkış işlevi tanıtıcıları)

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_ * (çıkış nedenleri)

MQXR_BEFORE	1	X'00000001'
MQXR_AFTER	2	X'00000002'
MQXR_CONNECTION	3	X'00000003'

MQXE_ * (ortamlar)

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* _ * (ek sabitler)

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*_* (boş değer değişmezleri)

MQXPDA_NONE X'00...00' (48 nulls)
MQXPDA_NONE_ARRAY '\0','\0',...,'\0','\0'

MQXCC_* (tamamlanma kodları)

MQXCC_FAILED -8

MQRC_* (neden kodları)

MQRC_API_EXIT_ERROR 2374 X'00000946'

Bir çıkış işlevi çağrısı geçersiz bir yanıt kodu döndürdü ya da bir şekilde başarısız oldu ve kuyruk yöneticisi yapılacak sonraki işlemi saptayamıyor.

Hatalı yanıt kodunu saptamak için MQAXP 'nin ExitResponse ve ExitResponse2 alanlarını inceleyin ve çıkışı geçerli bir yanıt kodu döndürecek şekilde değiştirin.

MQRC_API_EXIT_INIT_ERROR 2375 X'00000947'

Kuyruk yöneticisi, bir API çıkış işlevi için yürütme ortamını kullanıma hazırlarken bir hatayla karşılaştı.

MQRC_API_EXIT_TERM_ERROR 2376 X'00000948'

Kuyruk yöneticisi, bir API çıkış işlemine ilişkin yürütme ortamını kapatırken bir hata saptadı.

MQRC_EXIT_REASON_ERROR 2377 X'00000949'

Bir çıkış giriş noktası kayıt çağrısında (MQXEP) belirtilen ExitReason alanının değeri hatalı.

Hatalı çıkış nedeni değerini saptamak ve düzeltmek için ExitReason alanının değerini inceleyin.

MQRC_RESERVED_VALUE_ERROR 2378 X'0000094A'

Ayrılmış alanın değeri hatalı.

Ayrılmış değeri saptamak ve düzeltmek için Ayrılmış alanının değerini inceleyin.

C dili tip tipleri

Bu konu, C dilinde bulunan API çıkışlarıyla ilişkili tip tipleri hakkında bilgi sağlar.

API çıkışlarıyla ilişkili C dili tip tipleri şunlardır:

```
typedef PMLONG MQPOINTER PPMQLONG;  
typedef PMQBYTE MQPOINTER PPMQBYTE;  
typedef PMQHOBX MQPOINTER PPMQHOBX;  
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;
```

Çıkış giriş noktası kayıt çağrısı (MQXEP)

MQXEP, MQXEP C dil başlatma ve MQXEP C işlev prototipi hakkında bilgi edinmek için bu bilgileri kullanın.

MQXEP çağrısı kullanarak şunları yapın:

1. Çıkış işlevlerinin çağrılacağı IBM MQ API çıkış çağırma noktalarından önce ve sonra kaydetme
2. Çıkış işlevi giriş noktalarını belirtin
3. Çıkış işlevi giriş noktalarının kaydını kaldır

Genellikle MQ_INIT_EXIT çıkış işlevindeki MQXEP çağrılarını kodlar, ancak bunları sonraki herhangi bir çıkış işlevinde belirtebilirsiniz.

Zaten kayıtlı bir çıkış işlevini kaydettirmek için bir MQXEP çağırısı kullanırsanız, ikinci MQXEP çağırısı, kayıtlı çıkış işlevini değiştirerek başarıyla tamamlanır.

NULL çıkış işlevini kaydettirmek için bir MQXEP çağırısı kullanırsanız, MQXEP çağırısı başarıyla tamamlanır ve çıkış işlevinin kaydı kaldırılır.

Bir bağlantı isteğinin ömrü boyunca belirli bir çıkış işlevini kaydettirmek, kayıttan silmek ve yeniden kaydetmek için MQXEP çağrıları kullanılırsa, önceden kayıtlı çıkış işlevi yeniden etkinleştirilir. Ayrılmış ve bu çıkış işlevi eşgörünümüyle ilişkilendirilmiş saklama alanı, çıkışın işlevleri tarafından kullanılabilir. (Bu saklama alanı genellikle sonlandırma çıkış işlevinin çağrılması sırasında serbest bırakılır.)

MQXEP arabirimi:

```
MQXEP (Hconfig, ExitReason, Function, EntryPoint, &ExitOpts, &CompCode, &Reason)
```

Burada:

Hconfig (MQHCONFIG)-giriş

Başlatılmakta olan işlev kümesini içeren API çıkışını gösteren yapılandırma tanıtıcısı. Bu değer, MQ_INIT_EXIT işlevini çağırmadan hemen önce kuyruk yöneticisi tarafından oluşturulur ve MQAXP ' de her API çıkış işlevine geçirilir.

ExitReason (MQLONG)-giriş

Giriş noktasının kaydedilmesinin nedeni, aşağıdaki nedenlerden kaynaklanıyor:

- Bağlantı düzeyi kullanıma hazırlama ya da sonlandırma (MQXR_CONNECTION)
- Bir IBM MQ API çağırısından önce (MQXR_BEFORE)
- IBM MQ API çağırısından sonra (MQXR_AFTER)

İşlev (MQLONG)-giriş

MQXF_ * değişmezleri olan geçerli değerler olan işlev tanıtıcısı (bkz. [“Dış sabitler” sayfa 1543](#)).

EntryPoint (PMQFUNC)-giriş

Kaydedilecek çıkış işlevine ilişkin giriş noktasının adresi. NULL değeri, çıkış işlevinin sağlanmadığını ya da çıkış işlevinin önceki bir kaydının kayıttan kaldırıldığını gösterir.

ExitOpts(MQXEPO)

API çıkışları, API çıkışlarının nasıl kaydedildiğini denetleyen seçenekleri belirtebilir. Bu alan için boş değerli bir gösterge belirtilirse, MQXEPO yapısının varsayılan değerleri varsayılır.

CompCode (MQLONG)-çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

Tamamlanma kodu MQCC_FAILED ise:

MQRC_HCONFIG_HATA

(2280, X'8E8') Belirtilen yapılandırma tanıtıcısı geçerli değil. MQAXP ' deki yapılandırma tanıtıcısını kullanın.

MQRC_EXIT_REASON_HATA

(2377, X' 949 ') Belirtilen çıkış işlevi çağırma nedeni geçerli değil ya da belirtilen çıkış işlevi tanıtıcısı için geçerli değil.

Geçerli çıkış işlevi çağırma nedenlerinden (MQXR_* değeri) birini kullanın ya da geçerli bir işlem tanıtıcısı ve çıkış nedeni birleşimi kullanın. (Bkz. [Çizelge 836 sayfa 1547.](#))

MQRC_FUNCTION_HATA

(2281, X'8E9') Belirtilen işlem tanıtıcısı API çıkış nedeni için geçerli değil. Aşağıdaki çizelge, geçerli işlem tanıtıcıları ve ExitReasons(ExitReasons) birleşimlerini göstermektedir.

<i>Çizelge 836. Geçerli işlem tanıtıcısı birleşimleri ve ExitReasons</i>	
İşlev	ExitReason
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_GERI 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 ') Bir çıkış işlevini kaydetme ya da kayıttan çıkarma girişimi bir kaynak sorunu nedeniyle başarısız oldu.

MQRC_UNEXPECTED_HATASI

(2195, X'893 ') Bir çıkış işlevini kaydetme ya da kayıttan kaldırma girişimi beklenmeyen bir şekilde başarısız oldu.

MQRC_NAMETY_NAME_ERROR

(2442, X'098A') ExitProperties adı geçersiz.

MQRC_XEPO_HATA

(2507, X'09CB') Çıkış seçenekleri yapısı geçerli değil.

MQXEP C dil çağırısı

```
MQXEP (Hconfig, ExitReason, Function, EntryPoint, &ExitOpts, &CompCode, &Reason);
```

Değiştirge listesine ilişkin bildirim:

```
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 */
```

MQXEP C işlevi prototipi

```
void MQXEP (
MQHCONFIG      Hconfig,          /* Configuration handle */
MQLONG         ExitReason,      /* Exit reason */
MQLONG         Function,        /* Function identifier */
PMQFUNC        EntryPoint,      /* Function entry point */
PMQXEPO        pExitOpts;       /* Options that control the action of MQXEP */
MQLONG         pCompCode,       /* Address of completion code */
MQLONG         pReason);       /* Address of reason code qualifying completion
                               code */
```

İşlevleri çıkar

Bu bölümde, işlev çağrılarını kullanırken size yardımcı olacak bazı genel bilgiler ve tek tek çıkış işlevlerinin nasıl çağrılacağını açıklar.

API çıkış yordamlarına ilişkin genel kuralları anlamak ve çıkış yürütme ortamını ayarlamak ve temizlemek için bu bilgileri kullanın.

API çıkış yordamlarına ilişkin genel kurallar

API çıkış yordamları çağrılırken aşağıdaki genel kurallar geçerlidir:

- Her durumda, API çağırısı değiştirgelerinin geçerliliği denetlenmeden önce ve güvenlik denetimlerinden önce (MQCONN, MQCONNX ya da MQOPEN durumunda) API çıkış işlevleri yönlendirilir.
- Bir çıkış yordamına girilen ve çıkış yordamından alınan alanların değerleri şunlardır:
 - Bir *bundan önce* IBM MQ API çıkış işlevine girişte, bir alanın değeri uygulama programı tarafından ya da önceki bir çıkış işlevi çağırısı tarafından ayarlanabilir.
 - Bir *before* IBM MQ API çıkış işlevinden çıktıda, bir alanın değeri değiştirilmeden bırakılabilir ya da çıkış işleviyle başka bir değere ayarlanabilir.
 - Bir *bundan sonra* IBM MQ API çıkış işlevine girişte, bir alanın değeri IBM MQ API çağırısını işledikten sonra kuyruk yöneticisi tarafından ayarlanan değer olabilir ya da çıkış işlevleri zincirinde önceki bir çıkış işlevi çağırısıyla bir değere ayarlanabilir.
 - Bir *bundan sonra* IBM MQ API çağırısı çıkış işlevinin çıkışında, bir alanın değeri değiştirilmeden bırakılabilir ya da çıkış işleviyle başka bir değere ayarlanabilir.
- Çıkış işlevleri, ExitResponse ve ExitResponse2 alanlarını kullanarak kuyruk yöneticisiyle iletişim kurmalıdır.
- CompCode ve Neden kodu alanları uygulamayla iletişim kurar. Kuyruk yöneticisi ve çıkış işlevleri, CompCode ve Neden kodu alanlarını ayarlayabilir.
- MQXEP çağırısı, MQXEP ' yi çağırarak çıkış işlevlerine yeni neden kodları döndürür. Ancak, çıkış işlevleri bu yeni neden kodlarını var olan ve yeni uygulamaların anlayabileceği herhangi bir neden koduna çevirebilir.

- Her çıkış işlevi prototipinin, CompCode ve Reason dışında ek bir yönlendirme düzeyi olan API işlevine benzer parametreleri vardır.
- API çıkışları MQI çağrıları verebilir (MQDISC dışında), ancak bu MQI çağrıları API çıkışlarını kendileri çağırmaz.

Uygulamanın bir sunucuda ya da istemcide olması fark etmez, API çıkış çağrılarının sıralanmasını tahmin edemezsiniz. Bir API çıkışı BEFORE çağrısını hemen AFTER çağrısı izlemeyebilir.

BEFORE aramasını başka bir BEFORE araması izleyebilir. Örneğin:

```
MQCTL ' DEN ÖNCE
CALLBACK İşlemeden önce
MQPUT ÖNCESİ
MQPUT SONRASINDAN
GERİ ÇAĞIRMA SONRASI
MQCTL ' DEN SONRA
```

veya

```
XAOPEN 'DEN ÖNCE
MQCONNX ÖNCESİ
MQCONNX SONRASINDAN
XAOPEN 'DEN SONRA
```

İstemcide, PreConnect çıkışı adı verilen MQCONN ya da MQCONNX çağrısının davranışını değiştirebilen bir çıkış vardır. PreConnect çıkışı, kuyruk yöneticisi adını da içeren MQCONN ya da MQCONNX çağrısındaki değiştirgelerden herhangi birini değiştirebilir. İstemci önce bu çıkışı çağırır, daha sonra MQCONN ya da MQCONNX çağrılarını çağırır. Yalnızca ilk MQCONN ya da MQCONNX çağrısını API çıkışını çağırır; sonraki yeniden bağlanma çağrıları etkili olmaz.

Yürütme ortamı

Genel olarak, çıkış işlevlerindeki tüm hatalar, MQAXP ' deki ExitResponse ve ExitResponse2 alanları kullanılarak çıkış işleyicisine geri iletilir.

Bu hatalar da MQCC_ * ve MQRC_ * değerlerine dönüştürülür ve CompCode ve Reason alanlarında uygulamaya geri iletilir. Ancak, çıkış işleyicisi mantığında karşılaşılan hatalar uygulamaya CompCode ve Reason alanlarında MQCC_ * ve MQRC_ * değerleri olarak geri iletilir.

MQ_TERM_EXIT işlevi bir hata döndürürse:

- MQDISC çağrısı zaten gerçekleşmiş
- *bundan sonra* MQ_TERM_EXIT işlevini kullanmak için başka bir fırsat yoktur (bu nedenle, çıkış yürütme ortamı temizliği gerçekleştirilir)
- Çıkış yürütme ortamı temizliği gerçekleştirilmedi

Hala kullanımda olabileceği için çıkış boşaltılmıyor. Ayrıca, *before* çıkışının başarılı olduğu çıkış zincirinde daha aşağıda yer alan diğer kayıtlı çıkışlar da ters sırayla yönlendirilir.

Çıkış yürütme ortamını ayarlama

Belirtik bir MQCONN ya da MQCONNX çağrısı işlenirken, çıkış işleme mantığı çıkış kullanıma hazırlama işlevini (MQ_INIT_EXIT) çağırılmadan önce çıkış yürütme ortamını ayarlar. Çıkış yürütme ortamı kurulumu, çıkışın yüklenmesini, depolama alanının edinilmesini ve çıkış değiştirgesi yapılarının kullanıma hazırlanmasını içerir. Çıkış yapılandırma tanıtıcısı da ayrılır.

Bu aşamada hatalar oluşursa, MQCONN ya da MQCONNX çağrısı CompCode MQCC_FAILED ile başarısız olur ve aşağıdaki neden kodlarından biri:

MQRC_API_EXIT_LOAD_ERROR (UYGULAMA_HATASI)

Bir API çıkış modülünü yükleme girişimi başarısız oldu.

MQRC_API_EXIT_NOT_BULUNAMADI

API çıkış modülünde bir API çıkış işlevi bulunamadı.

MQRC_STORAGE_KULLANILAMIYOR

Kullanılabilir depolama alanı yetersiz olduğundan, API çıkış işlevi için yürütme ortamını başlatma girişimi başarısız oldu.

MQRC_API_EXIT_INIT_ERROR (UYGULAMA_HATASI)

API çıkış işlevi için yürütme ortamı kullanıma hazırlanırken bir hatayla karşılaşıldı.

Çıkış yürütme ortamı temizleniyor

Belirtik bir MQDISC çağrısı işlenirken ya da bir uygulamanın sona ermesinin sonucu olarak örtük bir bağlantı kesme isteği işlenirken, çıkış işleme mantığının, çıkış sonlandırma işlevini (MQ_TERM_EXIT) çağırıldıktan sonra çıkış yürütme ortamını temizlemesi gerekebilir.

Çıkış yürütme ortamının temizlenmesi, çıkış değiştirgesi yapıları için depolamanın serbest bırakılmasını ve daha önce belleğe yüklenen modüllerin silinmesini içerir.

Bu aşamada hatalar oluşursa, belirtik bir MQDISC çağrısı CompCode MQCC_FAILED ile başarısız olur ve neden kodu (örtük bağlantı kesme isteklerinde hatalar vurgulanmaz):

MQRC_API_EXIT_TERM_HATA

API çıkış işlevi için yürütme ortamı kapatılırken bir hatayla karşılaşıldı. Çıkış, MQ_TERM* API çıkış işlevi çağrılarında önce ya da sonra MQDISC ' den herhangi bir hata döndürmemelidir.

İstemcilerde API çıkışları

Bir istemci, MQCONN ve MQCONNX çağrılarının davranışını değiştirmek için PreConnect çıkışını kullanır ve API çıkış özelliklerini desteklemez.

PreConnect çıkışı

Bir istemcide, LDAP sunucusu gibi merkezi bir havuzdan kanal tanımlamasını aramak için PreConnect çıkışı kullanılabilir.

PreConnect çıkışı, bir MQCONN ya da MQCONNX çağrısındaki (örneğin, kuyruk yöneticisi adı) herhangi bir değiştirgeyi ya da tüm değiştirgeyi de değiştirebilir.

İstemci uygulamaları söz konusu olduğunda, PreConnect çıkışı API çıkışından önce çağrılmalıdır; MQCONN ya da MQCONNX API çıkışı yalnızca kuyruk yöneticisinin adı bilindikten sonra çağrılır ve bu ad PreConnect çıkışıyla değiştirilebilir.

Çıkışı yalnızca ilk MQCONN ya da MQCONNX çağrısından çağırıldığını unutmayın.

API çıkış özellikleri

Bir sunucuda API çıkışları, kullanıma hazırlama sırasında bir MQXEPO yapısını kaydedebilir. MQXEPO yapısı, çıkışın ilgilendiği özellikler grubunun ayrıntılarını içeren ExitProperties alanını içerir. Bu, çıkışın herhangi bir uygulama iletilisi özellik tanıtıcısından ayrı olarak işleyebileceği ayrı bir ileti özelliği tanıtıcısı oluşturma etkisine sahiptir.

Bir istemcide API çıkış özellikleri desteklenmez. Bir istemcide özellik grubu adını kaydetme girişiminde bulunulursa, işlev MQRC_EXIT_PROPS_NOT_SUPPORTED neden koduyla başarısız olur.

Geriletme-MQ_BACK_EXIT

MQ_BACK_EXIT, *bundan önce* ve *bundan sonra* geriletme işlemlerini gerçekleştirmek için bir geriletme çıkış işlevi sağlar. *bundan önce* ve *bundan sonra* arka çağrı çıkış işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_BACK işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```
MQ_BACK_EXIT (&ExitParms, &ExitContext, &Hconn, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış
Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış
Bağlam yapısından çık.

Hconn (MQHCONN)-giriş
Bağlantı tanıtıcısı.

CompCode (MQLONG)-giriş/çıkış
Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK
Başarıyla tamamlandı.

MQCC_UYARISI
Kısmi tamamlama.

MQCC_FAILED
Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış
Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE
(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_BACK_EXIT (&ExitParms, &ExitContext, &Hconn, &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```

Başlat-MQ_BEGIN_EXIT

MQ_BEGIN_EXIT, *bundan önce ve bundan sonra* MQBEGIN çağrı işlemlerini gerçekleştirmek için bir başlangıç çıkış işlevi sağlar. *bundan önce ve bundan sonra* MQBEGIN çağrı çıkış işlevlerini kaydetmek için MQXF_BEGIN işlev tanıtıcısını MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle kullanın.

Bu işleve ilişkin arabirim:

```
MQ_BEGIN_EXIT (&ExitParms, &ExitContext, &Hconn, &pBeginOptions, &CompCode,
               &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pBeginSeçenekleri (PMQBO)-giriş/çıkış

Seçenekleri başlatmak için işaretçi.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_BEGIN_EXIT (&ExitParms, &ExitContext, &Hconn, &pBeginOptions, &CompCode,
               &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```


Geri çağırma-MQ_CALLBACK_EXIT

MQ_CALLBACK_EXIT, *bundan önce ve bundan sonra* geri çağırma işlemlerini gerçekleştirmek için bir çıkış işlevi sağlar. MQXF_CALLBACK işlev tanıtıcısını MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle kullanarak *bundan önce ve bundan sonra* geri çağırma çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
MQ_CALLBACK_EXIT (&ExitParms, &ExitContext, &Hconn, &pMsgDesc, &pGetMsgOpts,  
                 &pBuffer, &MQCBCContext)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık

Hconn (MQHCONN)-giriş/çıkış

Bağlantı tanıtıcı

pMsgAçıklaması

İleti tanımlayıcı

pGetMsgOpts

MQGET işlemini denetleyen seçenekler

pBuffer

İleti verilerini içerecek alan

pmQCBCContext

Geri çağırma ile ilgili bağlam verileri

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
MQAXP      ExitParms;      /* Exit parameter structure */  
MQAXC      ExitContext;   /* Exit context structure */  
MQHCONN    Hconn;        /* Connection handle */  
PMQMD      pMsgDesc;     /* Message descriptor */  
PMQGM      pGetMsgOpts;  /* Options that define the operation of the consumer */  
PMQVOID    pBuffer;     /* Area to contain the message data */  
PMQCBC     pContext;     /* Context data for the callback */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_SUBRQ_EXIT (&ExitParms, &ExitContext, &Hconn, &pMsgDesc, &pGetMsgOpts, &pBuffer,  
              &pContext);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
void MQENTRY MQ_CALLBACK_EXIT (  
PMQAXP      pExitParms;   /* Exit parameter structure */  
PMQAXC      pExitContext; /* Exit context structure */  
PMQHCONN    pHconn;      /* Connection handle */  
PPMQMD      ppMsgDesc;   /* Message descriptor */  
PPMQGM      ppGetMsgOpts; /* Options that define the operation of the consumer */  
PPMQVOID    ppBuffer;    /* Area to contain the message data */  
PPMQCBC     ppContext;   /* Context data for the callback */
```

Kullanım notları

1. Geri çağırma çıkışı, tüketici çağrılmadan önce ve tüketicinin tüketici işlevi tamamlandıktan sonra çağrılır. MQMD ve MQGMO yapıları değiştirilebilirse de, önceki çıkıştaki değerlerin değiştirilmesi, ileti

tüketici işlevine teslim edilecek kuyruktan zaten kaldırıldığından, iletinin kuyruktan alınmasını yeniden teşvik etmez.

Geri çağırma işlevlerini yönet-MQ_CB_EXIT

MQ_CB_EXIT, *bundan önce* ve *bundan sonra* MQCB çağrılarını gerçekleştirmek için bir çıkış işlevi sağlar. *bundan önce* ve *bundan sonra* MQCB çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_CB işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```
MQ_CB_EXIT (&ExitParms, &ExitContext, &Hconn, &Operation, &pCallbackDesc,  
            &Hobj, &pMsgDesc, &pGetMsgOpts, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık

Hconn (MQHCONN)-giriş/çıkış

Bağlantı tanıtıcı

İşlem (MQLONG)-giriş/çıkış

İşlem değeri

pCallbackAçıklaması (PMQCBD)-giriş/çıkış

Geri çağırma tanımlayıcısı

Hobj (MQHOBJ)-giriş/çıkış

Nesne tanıtıcısı

pMsgDesc (PMQMD)-giriş/çıkış

İleti tanımlayıcı

pGetMsgOpts (PMQGMO)-giriş/çıkış

MQCB 'nin işlemini denetleyen seçenekler

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu

Neden (MQLONG)-giriş/çıkış

Neden kodu niteleyici CompCode

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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. */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_CB_EXIT (&ExitParms, &ExitContext, &Hconn, &Operation, &Hobj, &pMsgDesc,  
            &pGetMsgOpts, &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```

void MQENTRY MQ_CB_EXIT (
PMQAXP    pExitParms;    /* Exit parameter structure */
PMQAXC    pExitContext;  /* Exit context structure */
PMQHCONN  pHconn;       /* Connection handle */
PMLONG    pOperation;   /* Callback operation */
PMQHOBJS  pHobj;        /* Object handle */
PPMQMD    ppMsgDesc;    /* Message descriptor */
PPMQGMO   ppGetMsgOpts; /* Options that control the action of MQCB */
PMLONG    pCompCode;    /* Completion code */
PMLONG    pReason;      /* Reason code qualifying CompCode */

```

Kapat-MQ_CLOSE_EXIT

MQ_CLOSE_EXIT, *bundan önce ve bundan sonra* MQCLOSE çağrı işlemlerini gerçekleştirmek için bir kapatma çıkış işlevi sağlar. *bundan önce ve bundan sonra* MQCLOSE çağrı çıkış işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_CLOSE işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```

MQ_CLOSE_EXIT (&ExitParms, &ExitContext, &Hconn, &pHobj,
               &Options, &CompCode, &Reason)

```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pHobj (PMQHOBJS)-giriş

Nesne tanıtıcısı işaretçisi.

Seçenekler (PMLONG)-giriş/çıkış

Seçenekleri kapatın.

CompCode (PMLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu

Neden (PMLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ise, çıkış işlevi neden kodu alanını herhangi bir geçerli MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```

MQAXP    ExitParms;    /* Exit parameter structure */
MQAXC    ExitContext;  /* Exit context structure */
MQHCONN  Hconn;       /* Connection handle */
PMQHOBJS pHobj;       /* Ptr to object handle */
PMLONG   Options;     /* Close options */

```

```
MQLONG      CompCode;      /* Completion code */
MQLONG      Reason;        /* Reason code */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_CLOSE_EXIT (&ExitParms, &ExitContext,&Hconn, &pHobj, &Options,
               &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```

Kesinleştir-MQ_CMITS_EXIT

MQ_CMITS_EXIT, kesinleştirme işlemesinden *önce* ve *sonra* gerçekleştirmek için bir kesinleştirme çıkışı işlevi sağlar. *bundan önce* ve *bundan sonra* çağrı çıkışı işlevlerini kesinleştirmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_CMITS işlev tanıtıcısını kullanın.

Bir kesinleştirme işlemi başarısız olursa ve hareket geri çekilirse, MQCMITS çağrısı MQCC_WARNING ve MQRC_BACKED_OUT ile başarısız olur. Bu dönüş ve neden kodları, çıkış işlevlerine iş biriminin geriletildiğini gösteren *bundan sonra* MQCMITS çıkış işlevlerine aktarılır.

Bu işleve ilişkin arabirim:

```
MQ_CMITS_EXIT (&ExitParms, &ExitContext, &Hconn, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_CMITY_EXIT (&ExitParms, &ExitContext, &Hconn, &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```

Kullanım notları

1. Burada açıklanan MQ_GET_EXIT işlev arabirimi hem MQXF_GET çıkış işlevi hem de [“MQXF_DATA_CONV_ON_GET” sayfa 1563 çıkış işlevi için kullanılır.](#)

Bu iki çıkış işlevi için ayrı giriş noktaları tanımlanır; bu nedenle, *her ikisini* engellemek için MQXEP çağrısının iki kez kullanılması gerekir; bu çağrı kullanım işlevi tanıtıcısı MQXF_GET.

MQ_GET_EXIT arabirimi MQXF_GET ve MQXF_DATA_CONV_ON_GET için aynı olduğundan, her ikisi için de tek bir çıkış işlevi kullanılabilir; MQAXP yapısındaki *Function* alanı hangi çıkış işlevinin çağrıldığını gösterir. Diğer bir seçenek olarak, MQXEP çağrısı iki durum için farklı çıkış işlevlerini kaydettirmek için kullanılabilir.

Bağlan ve bağlan uzantısı-MQ_CONNX_EXIT

MQ_CONNX_EXIT, *bundan önce* ve *bundan sonra* MQCONN işlemlerini gerçekleştirmek için bağlantı çıkışı işlevi, *bundan önce* ve *bundan sonra* MQCONNX işlemlerini gerçekleştirmek için bağlantı uzantısı çıkış işlevi sağlar.

Burada açıklandığı gibi, hem MQCONN hem de MQCONNX çağrı çıkışı işlevleri için aynı arabirim çağrılır.

İleti kanalı aracısı (MCA) gelen istemci bağlantısına yanıt verdiğinde, MCA istemci durumu tam olarak bilinmeden önce IBM MQ API çağrılarında bağlanabilir ve bu çağrıları yapabilir. Bu API çağrıları, MCA programının kendisine dayalı olarak (örneğin, MQAXC 'nin UserId ve ConnectionName alanlarında) MQAXC ile API çıkış işlevlerini çağırır.

MCA sonraki gelen istemci API çağrılarında yanıt verdiğinde, MQAXC yapısı gelen istemciyi temel alır ve UserId ve ConnectionName alanlarını uygun şekilde ayarlar.

Bir MQCONN ya da MQCONNX çağrısında uygulama tarafından ayarlanan kuyruk yöneticisi adı, temeldeki bağlanma çağrısına geçerilir. Kuyruk yöneticisinin adını değiştirmek için *bundan önce* MQ_CONNX_EXIT girişimleri sonuç vermez.

bundan önce ve *bundan sonra* MQCONN ve MQCONNX çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_CONN ve MQXF_CONNX işlev tanıtıcılarını kullanın.

An MQ_CONNX_EXIT exit called for reason MQXR_BEFORE *olmamalıdır* issue any IBM MQ API calls, as the correct environment has not been set up at this time.

Bir MQ_CONNX_EXIT, çağrıldığı bağlantıya ilişkin bir API çıkış çağrısından MQDISC ' yi çağırılmaz. Bu kısıtlama hem istemci hem de sunucu API çıkışları için geçerlidir.

MQCONN ve MQCONNX arabirimi aynı:

```
MQ_CONNX_EXIT (&ExitParms, &ExitContext, &pQMgrName, &pConnectOpts,  
&pHconn, &CompCode, &Reason);
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

pQMGrAd (PMQCHAR)-giriş

MQCONNX çağrısında belirtilen kuyruk yöneticisi adını gösteren gösterge. MQCONN ya da MQCONNX çağrısında çıkış bu adı değiştirmemelidir.

pConnectSeçmeler (PMQCNO)-giriş/çıkış

MQCONNX çağrısının işlemini denetleyen seçeneklere ilişkin gösterge.

Ayrıntılar için bkz. [“MQCNO-Bağlantı seçenekleri” sayfa 316.](#)

MQXF_CONN çıkış işlevi için, pConnectSeçenekleri varsayılan bağlanma seçenekleri yapısını gösteriyor (MQCNO_DEFAULT).

pHconn (PMQHCONN)-giriş

Bağlantı tanıtıcısını gösteren gösterge.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama)

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_CONNX_EXIT (&ExitParms, &ExitContext, &pQMgrName, &pConnectOps,
                &pHconn, &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
PPMQCNO     ppConnectOpts,  /* Address of ptr to connection options */
PPMQHCONN   pHconn,         /* Address of ptr to connection handle */
PMQLONG     pCompCode,      /* Address of completion code */
PMQLONG     pReason);      /* Address of reason code qualifying
                             completion code */
```

Kullanım notları

1. Burada açıklanan MQ_CONNX_EXIT işlevi arabirimi hem MQCONN çağrısı hem de MQCONNX çağrısı için kullanılır. Ancak, bu iki çağrı için ayrı giriş noktaları tanımlanır. Her iki çağrıyı da engellemek için, MQXEP çağrısı MQXF_CONN işlev tanıtıcısıyla en az iki kez ve MQXF_CONN ile yeniden kullanılmalıdır.

MQ_CONNX_EXIT arabirimi MQCONN ve MQCONNX için aynı olduğundan, her iki çağrı için de tek bir çıkış işlevi kullanılabilir; MQAXP yapısındaki *Function* alanı hangi çağrının devam ettiğini gösterir. Diğer bir seçenek olarak, MQXEP çağrısı, iki çağrı için farklı çıkış işlevlerini kaydettirmek üzere kullanılabilir.

2. Bir ileti kanalı aracısı (MCA) gelen istemci bağlantısına yanıt verdiğinde, MCA istemci durumu tam olarak bilinmeden önce bir dizi MQ çağrısı verebilir. Bu MQ çağrıları, API çıkış işlevlerinin istemciyle (örneğin, kullanıcı kimliği ve bağlantı adı) değil, MCA ile ilgili verileri içeren MQAXC yapısıyla çağrılmasına neden olur. Ancak, istemci durumu tam olarak bilindikten sonra sonraki MQ çağrıları, API çıkış işlevlerinin MQAXC yapısındaki uygun istemci verileriyle çağrılmasına neden olur.
3. Kuyruk yöneticisi herhangi bir değiştirge geçerlilik denetimi gerçekleştirmeden önce tüm MQXR_BEFORE çıkış işlevleri çağrılır. Bu nedenle parametreler geçersiz olabilir (parametrelerin adresleri için geçersiz işaretçiler dahil).

Kuyruk yöneticisi herhangi bir yetki denetimi gerçekleştirmeden önce MQ_CONNX_EXIT işlevi çağrılır.

4. Çıkış işlevi, MQCONN ya da MQCONNX çağrısında belirtilen kuyruk yöneticisinin adını değiştirmemelidir. Ad çıkış işleviyle değiştirilirse, sonuçlar tanımsız olur.
5. MQ_CONNX_EXIT için bir MQXR_BEFORE çıkış işlevi, MQXEP dışında MQ çağrıları yayınlamayamaz.

Denetim geri çağrısı-MQ_CTL_EXIT

MQ_CTL_EXIT, *bundan önce* ve *bundan sonra* denetim geri çağırma işlemini gerçekleştirmek için bir abonelik isteği çıkış işlevi sağlar. MQXF_CTL işlev tanıtıcısını MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle kullanarak *bundan önce* ve *bundan sonra* denetim geri çağırma çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
MQ_CTL_EXIT (&Hconn, &Operation, &ControlOpts, &CompCode, &Reason)
```

burada parametreler şunlardır:

Hconn (MQHCONN)-giriş/çıkış

Bağlantı tanıtıcısı.

İşlem (MQLONG) giriş/çıkış

Belirtilen nesne tanıtıcısı için tanımlanan geri çağırma sırasında işlenmekte olan işlem

ControlOpts (MQCTLO) giriş/çıkış

MQCTL işlemini denetleyen seçenekler

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_CTL_EXIT (&Hconn, &Operation, &ControlOpts, &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```

Bağlantıyı Kes-MQ_DISC_EXIT

MQ_DISC_EXIT, *bundan önce* ve *bundan sonra* MQDISC çıkış işlemlerini gerçekleştirmek için bir bağlantı kesme çıkışı işlevi sağlar. *bundan önce* ve *bundan sonra* MQDISC çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_DISC işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```
MQ_DISC_EXIT (&ExitParms, &ExitContext, &pHconn,
&CompCode, &Reason);
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

pHconn (PMQHCONN)-giriş

Bağlantı tanıtıcısını gösteren gösterge.

MQDISC öncesi çağrı için, bu alanın değeri aşağıdakilerden biridir:

- MQCONN ya da MQCONNX çağrısında bağlantı tanıtıcısı döndürüldü
- Ortama özgü bağdaştırıcının kuyruk yöneticisine bağlı olduğu ortamlar için sıfır
- Önceki bir çıkış işlevi çağrısıyla ayarlanan değer

MQDISC çağrısından sonra için, bu alanın değeri sıfır ya da önceki bir çıkış işlevi çağrısı tarafından ayarlanan bir değerdir.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
PMQHCONN   pHconn;        /* Ptr to Connection handle */
MQLONG     CompCode;      /* Completion code */
MQLONG     Reason;        /* Reason code */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_DISC_EXIT (&ExitParms, &ExitContext, &pHconn,
              &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
void MQENTRY MQ_DISC_EXIT (
PMQAXP      pExitParms,    /* Address of exit parameter structure */
PMQAXC      pExitContext,  /* Address of exit context structure */
PPMHCONN    ppHconn,      /* Address of ptr to connection handle */
PMQLONG     pCompCode,    /* Address of completion code */
PMQLONG     pReason);     /* Address of reason code qualifying
                             completion code */
```

Get-MQ_GET_EXIT

MQ_GET_EXIT, *bundan önce* ve *bundan sonra* MQGET çağrılarını işlemek için bir get exit işlevi sağlar.

İki işlev tanıtıcısı vardır:

1. *bundan önce* ve *bundan sonra* MQGET çağrı çıkışı işlevlerini kaydetmek için MQXF_GET çıkış nedenleriyle MQXR_BEFORE ve MQXR_AFTER kullanın.

2. MQXF_DATA_CONV_ON_GET işlev tanıtıcısının kullanılmasıyla ilgili bilgi için bkz. [“MQXF_DATA_CONV_ON_GET” sayfa 1563](#) .

Bu işleve ilişkin arabirim:

```
MQ_GET_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &pMsgDesc,  
             &pGetMsgOpts, &BufferLength, &pBuffer, &pDataLength,  
             &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

Hobj (MQHOBJ)-giriş/çıkış

Nesne tanıtıcısı.

pMsgDesc (PMQMD)-giriş/çıkış

İleti tanımlayıcısına ilişkin gösterge.

pGetMsgOpts (PMQGMO)-giriş/çıkış

İleti seçeneklerini almak için işaretçi.

BufferLength (MQLONG)-giriş/çıkış

İleti arabelleği uzunluğu.

pBuffer (PMQBYTE)-giriş/çıkış

İleti arabelleğine işaret ediyor.

pDataUzunluğu (PMQLONG)-giriş/çıkış

Veri uzunluğu alanını gösterir.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */

```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```

MQ_GET_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &pMsgDesc,
             &pGetMsgOpts, &BufferLength, &pBuffer, &pDataLength,
             &CompCode, &Reason)

```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```

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 */
MQLONG      pBufferLength, /* Address of message buffer length */
PPMQBYTE    ppBuffer,     /* Address of ptr to message buffer */
PPMQLONG    ppDataLength, /* Address of ptr to data length field */
MQLONG      pCompCode,    /* Address of completion code */
MQLONG      pReason);     /* Address of reason code qualifying
                             completion code */

```

Kullanım notları

1. Burada açıklanan MQ_GET_EXIT işlev arabirimi hem MQXF_GET çıkış işlevi hem de [“MQXF_DATA_CONV_ON_GET”](#) sayfa 1563 çıkış işlevi için kullanılır.

Bu iki çıkış işlevi için ayrı giriş noktaları tanımlanır; bu nedenle, *her ikisini* engellemek için MQXEP çağrısının iki kez kullanılması gerekir; bu çağrı kullanım işlevi tanıtıcısı MQXF_GET.

MQ_GET_EXIT arabirimi MQXF_GET ve MQXF_DATA_CONV_ON_GET için aynı olduğundan, her ikisi için de tek bir çıkış işlevi kullanılabilir; MQAXP yapısındaki *Function* alanı hangi çıkış işlevinin çağrıldığını gösterir. Diğer bir seçenek olarak, MQXEP çağrısı iki durum için farklı çıkış işlevlerini kaydettirmek için kullanılabilir.

MQXF_DATA_CONV_ON_GET

MQXF_DATA_CONV_ON_GET işlev tanıtıcısı MQ_GET_EXIT ile kullanılır.

Bu çağrıya ilişkin arabirim ve örnek bir C dili bildirimi hakkında bilgi için bkz. [MQ_GET_EXIT](#) .

Kullanım notları

Kayıtlıysa, iletiler uygulamaya geldiğinde ancak herhangi bir veri dönüştürme gerçekleşmeden önce bu giriş noktası çağrılır. Bu, ileti veri dönüştürmeye geçirilmeden önce API çıkışının şifre çözme ya da açma gibi işlemleri gerçekleştirmesi gerekiyorsa yararlı olabilir. Çıkış, gerekiyorsa, MQXCC_SUPPRESS_FUNCTION döndürülerek veri dönüştürmenin atlanmasına neden olabilir; daha fazla bilgi için [MQAXP](#) yapısına bakın.

Bu giriş noktasının istemcide kaydedilmesi, veri dönüştürmenin istemci makinesinde yerel olarak gerçekleştirilmesine neden olur. Doğru işlem için, uygulama dönüştürme çıkışlarını istemcide kurmak gerekebilir. MQXF_DATA_CONV_ON_GET ögesinin zamanuyumsuz tüketmek için de kullanıldığını unutmayın.

[MQ_GET_EXIT](#) çağrısı kullanırken, MQXF_DATA_CONV_ON_GET komutunu MQXR_BEFORE çıkış nedeniyle kullanarak bir *bundan önce* MQGET veri dönüştürme çıkış işlevini kaydettirin.

MQXF_DATA_CONV_ON_GET için MQXR_AFTER çıkış işlevi yoktur; MQXF_GET için MQXR_AFTER çıkış işlevi, veri dönüştürmeden sonra çıkış için gereken yeteneği sağlar.

`MQ_GET_EXIT` çağrısı için ayrı giriş noktaları tanımlanır; bu nedenle *her iki* çıkış işlevini kesmek için `MQXEP` çağrısı iki kez kullanılmalıdır; bu çağrı kullanım işlevi tanıtıcısı için `MQXF_DATA_CONV_ON_GET`.

`MQ_GET_EXIT` arabirimi `MQXF_GET` ve `MQXF_DATA_CONV_ON_GET` için aynı olduğundan, her ikisi için de tek bir çıkış işlevi kullanılabilir; `MQAXP` yapısındaki *Function* alanı hangi çıkış işlevinin çağrıldığını gösterir. Diğer bir seçenek olarak, `MQXEP` çağrısı iki durum için farklı çıkış işlevlerini kaydettirmek için kullanılabilir.

Kullanıma hazırlama-MQ_INIT_EXIT

`MQ_INIT_EXIT`, `MQAXP` ' de `ExitReason` değerinin `MQXR_CONNECTION` olarak ayarlanmasıyla gösterilen bağlantı düzeyi kullanıma hazırlama sağlar.

Başlatma sırasında aşağıdakilere dikkat edin:

- `MQ_INIT_EXIT` işlevi, IBM MQ API komutlarını ve ilgili olduğu `ENTRY` ve `EXIT` noktalarını kaydettirmek için `MQXEP` ' yi çağırır.
- Çıkışların tüm IBM MQ API fiillerini kesmesi gerekmez. Çıkış işlevleri yalnızca bir ilgi alanı kaydedildiyse çağrılır.
- Çıkış tarafından kullanılacak depolama, başlatılırken alınabilir.
- Bu işleve çağrı başarısız olursa, çağrılan `MQCONN` ya da `MQCONNX` çağrısı da `CompCode` ve `MQAXP` ' deki `ExitResponse` alanının değerine bağlı olarak başarısız olur.
- Şu anda doğru ortam ayarlanmadığından, bir `MQ_INIT_EXIT` çıkışı IBM MQ API çağrılarını yayınlamamalıdır.
- Bir `MQ_INIT_EXIT` `MQXCC_FAILED` ile başarısız olursa, kuyruk yöneticisi `MQCC_FAILED` ve `MQRC_API_EXIT_ERROR` ile çağrılan `MQCONN` ya da `MQCONNX` çağrısından geri döner.
- Kuyruk yöneticisi, ilk `MQ_INIT_EXIT` çağrılmadan önce API çıkış işlevi yürütme ortamını kullanıma hazırlarken bir hata saptarsa, `MQCC_FAILED` ve `MQRC_API_EXIT_INIT_ERROR` ile `MQ_INIT_EXIT` çağrısını çağırarak `MQCONN` ya da `MQCONNX` çağrısından döner.

`MQ_INIT_EXIT` arabirimi:

```
MQ_INIT_EXIT (&ExitParms, &ExitContext, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma koduna ilişkin geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden koduna işaret eden gösterge.

Tamamlanma kodu `MQCC_OK` ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu `MQCC_FAILED` ya da `MQCC_WARNING` ise, çıkış işlevi neden kodu alanını geçerli bir `MQRC_*` değerine ayarlayabilir.

Uygulamaya döndürülen CompCode ve Nedeni, MQAXP ' deki ExitResponse alanının değerine bağlıdır.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQLONG     CompCode;       /* Completion code */
MQLONG     Reason;        /* Reason code */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_INIT_EXIT (&ExitParms, &ExitContext, &CompCode, &Reason)
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```

Kullanım notları

1. MQ_INIT_EXIT işlevi, kesilecek belirli MQ çağrılarına ilişkin çıkış işlevlerinin adreslerini kaydettirmek için MQXEP çağrısını verebilir. Tüm MQ çağrılarını kesmek ya da hem MQXR_BEFORE, hem de MQXR_AFTER çağrılarını kesmek gerekmez. Örneğin, bir çıkış takımı yalnızca MQPUT MQXR_BEFORE çağrısının önünü kesmeyi seçebilir.
2. Çıkış takımındaki çıkış işlevleri tarafından kullanılacak depolama alanı, MQ_INIT_EXIT işlevi tarafından edinilebilir. Diğer bir seçenek olarak, çıkış işlevleri gerektiğinde ve çağrıldığında saklama alanı edinebilir. Ancak, çıkış takımı sonlandırılmadan önce tüm saklama alanı serbest bırakılmalıdır; MQ_TERM_EXIT işlevi depolamayı serbest bırakabilir ya da daha önce çağrılan bir çıkış işlevi olabilir.
3. MQ_INIT_EXIT, MQAXP 'nin ExitResponse alanında MQXCC_FAILED değerini döndürürse ya da başka bir şekilde başarısız olursa, MQ_INIT_EXIT' in çağrılmasına neden olan MQCONN ya da MQCONNX çağrısı da başarısız olur; **CompCode** ve **Reason** parametreleri uygun değerlere ayarlanır.
4. MQ_INIT_EXIT işlevi, MQXEP dışında MQ çağrılarını yayınlamaz.

Sorgu-MQ_INQ_EXIT

MQ_INQ_EXIT, *bundan önce ve bundan sonra* MQINQ çağrı işlemlerini gerçekleştirmek için bir sorgu çıkış işlevi sağlar. *bundan önce ve bundan sonra* MQINQ çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_INQ işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```
MQ_INQ_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &SelectorCount,
             &pSelectors, &IntAttrCount, &pIntAttrs, &CharAttrLength,
             &pCharAttrs, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

Hobj (MQHOBJ)-giriş

Nesne tanıtıcısı.

SelectorCount (MQLONG)-giriş

Seçicilerin sayısı

pSelectors (PMQLONG)-giriş/çıkış

Seçici değerleri dizisine işaretçi.

IntAttrSayı (MQLONG)-giriş

Tamsayı özniteliklerinin sayısı.

pIntÖzntlk (PMQLONG)-giriş/çıkış

Tamsayı öznitelik değerleri dizisini gösteren gösterge.

CharAttrUzunluk (MQLONG)-giriş/çıkış

Karakter öznitelikleri dizisi uzunluğu.

pCharÖzntlk (PMQCHAR)-giriş/çıkış

Karakter öznitelikleri dizisini gösteren gösterge.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_INQ_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &SelectorCount,
             &pSelectors, &IntAttrCount, &pIntAttrs, &CharAttrLength,
             &pCharAttrs, &CompCode, &Reason)
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```

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 */
PMQHOBJS  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 */

```

Aç-MQ_OPEN_EXIT

MQ_OPEN_EXIT, *bundan önce* ve *bundan sonra* MQOPEN çağrı işlemlerini gerçekleştirmek için bir açık çıkış işlevi sağlar. *bundan önce* ve *bundan sonra* MQOPEN çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_OPEN işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```

MQ_OPEN_EXIT (&ExitParms, &ExitContext, &Hconn, &pObjDesc, &Options,
&pHobj, &CompCode, &Reason)

```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pObjTanım (PMQOD)-giriş/çıkış

Nesne tanımlayıcısına ilişkin gösterge.

Seçenekler (MQLONG)-giriş/çıkış

Seçenekleri açın.

pHobj (PMQHOBJS)-giriş

Nesne tanıtıcısı işaretçisi.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;   /* Exit context structure */
MQHCONN    Hconn;        /* Connection handle */
PMQOD      pObjDesc;     /* Ptr to object descriptor */
MQLONG     Options;      /* Open options */
PMQHOBJS   pHobj;       /* Ptr to object handle */
MQLONG     CompCode;     /* Completion code */
MQLONG     Reason;      /* Reason code */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_OPEN_EXIT (&ExitParms, &ExitContext, &Hconn, &pObjDesc, &Options,
              &pHobj, &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
PPMQHOBJS   ppHobj,     /* Address of ptr to object handle */
PMQLONG     pCompCode,   /* Address of completion code */
PMQLONG     pReason);    /* Address of reason code qualifying
                           completion code */
```

Koyma-MQ_PUT_EXIT

MQ_PUT_EXIT, *bundan önce ve bundan sonra* MQPUT çağrılarını işlemek için bir koyma çıkışı işlevi sağlar. *bundan önce ve bundan sonra* MQPUT çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_PUT işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```
MQ_PUT_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &pMsgDesc,
             &pPutMsgOpts, &BufferLength, &pBuffer, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

Hobj (MQHOBJS)-giriş/çıkış

Nesne tanıtıcısı.

pMsgDesc (PMQMD)-giriş/çıkış

İleti tanımlayıcısına ilişkin gösterge.

pPutMsgOpts (PMQPMO)-giriş/çıkış

İleti seçeneklerini koymak için işaretçi.

BufferLength (MQLONG)-giriş/çıkış

İleti arabelleği uzunluğu.

pBuffer (PMQBYTE)-giriş/çıkış

İleti arabelleğine işaret ediyor.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_PUT_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &pMsgDesc,
             &pPutMsgOpts, &BufferLength, &pBuffer, &CompCode, &Reason)
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
void MQENTRY MQ_PUT_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 */
    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 */
```

Kullanım notları

- Kuyruk yöneticisi tarafından oluşturulan rapor iletileri, olağan çağrı işlemeyi atlar. Sonuç olarak, bu tür iletiler MQ_PUT_EXIT işlevi ya da MQPUT1 işlevi tarafından engellenemez. Ancak, ileti kanalı aracısı tarafından oluşturulan rapor iletileri olağan şekilde işlenir ve bu nedenle, MQ_PUT_EXIT işlevi ya da MQ_PUT1_EXIT işlevi tarafından yakalanabilir. MCA tarafından oluşturulan tüm rapor iletilerinin alındığından emin olmak için hem MQ_PUT_EXIT hem de MQ_PUT1_EXIT kullanılmalıdır.

Put1 - MQ_PUT1_EXIT

MQ_PUT1_EXIT , *bundan önce* ve *bundan sonra* MQPUT1 çağrı işlemlerini gerçekleştirmek için bir *yalnızca bir ileti* koy çıkış işlevi sağlar. MQXF_PUT1 işlev tanıtıcısını, *bundan önce* ve *bundan sonra* MQPUT1 çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle kullanın.

Bu işleve ilişkin arabirim:

```
MQ_PUT1_EXIT (&ExitParms, &ExitContext, &Hconn, &pObjDesc, &pMsgDesc,
&pPutMsgOpts, &BufferLength, &pBuffer, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pObjTanım (PMQOD)-giriş/çıkış

Nesne tanımlayıcısına ilişkin gösterge.

pMsgDesc (PMQMD)-giriş/çıkış

İleti tanımlayıcısına ilişkin gösterge.

pPutMsgOpts (PMQPMO)-giriş/çıkış

İleti seçeneklerini koymak için işaretçi.

BufferLength (MQLONG)-giriş/çıkış

İleti arabelleği uzunluğu.

pBuffer (PMQBYTE)-giriş/çıkış

İleti arabelleğine işaret ediyor.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */

```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```

MQ_PUT1_EXIT (&ExitParms, &ExitContext, &Hconn, &pObjDesc, &pMsgDesc,
              &pPutMsgOpts, &BufferLength, &pBuffer, &CompCode, &Reason)

```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```

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 */
PMQBYTE     pBuffer, /* Address of ptr to message buffer */
PMQLONG     pCompCode, /* Address of completion code */
PMQLONG     pReason); /* Address of reason code qualifying
                        completion code */

```

Ayarla-MQ_SET_EXIT

MQ_SET_EXIT, *bundan önce ve bundan sonra* MQSET çağrı işlemlerini gerçekleştirmek için bir çıkış ayarlama işlevi sağlar. *bundan önce ve bundan sonra* MQSET çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_SET işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```

MQ_SET_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &SelectorCount,
             &pSelectors, &IntAttrCount, &pIntAttrs, &CharAttrLength,
             &pCharAttr, &CompCode, &Reason)

```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

Hobj (MQHOBJ)-giriş

Nesne tanıtıcısı.

SelectorCount (MQLONG)-giriş

Seçicilerin sayısı

pSelectors (PMQLONG)-giriş/çıkış

Seçici değerleri dizisine işaretçi.

IntAttrSayı (MQLONG)-giriş

Tamsayı özniteliklerinin sayısı.

pIntÖzntlk (PMQLONG)-giriş/çıkış

Tamsayı öznitelik değerleri dizisini gösteren gösterge.

CharAttrUzunluk (MQLONG)-giriş/çıkış

Karakter öznitelikleri dizisi uzunluğu.

pCharÖzntlk (PMQCHAR)-giriş/çıkış

Karakter öznitelik değeri gösteren gösterge.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_SET_EXIT (&ExitParms, &ExitContext, &Hconn, &Hobj, &SelectorCount,
             &pSelectors, &IntAttrCount, &pIntAttrs, &CharAttrLength,
             &pCharAttrs, &CompCode, &Reason)
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```

Durum-MQ_STAT_EXIT

MQ_STAT_EXIT, *bundan önce* ve *bundan sonra* MQSTAT çağrılarını işlemek için bir durum çıkışı işlevi sağlar. *bundan önce* ve *bundan sonra* MQSTAT çağrı çıkışı işlevlerini kaydetmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_STAT işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```
MQ_STAT_EXIT (&ExitParms, &ExitContext, &Hconn, &Type, &pStatus
              &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

Tip (MQLONG)-giriş

Alınacak durum bilgilerinin tipi.

pStatus (PMQSTS)-çıkış

Durum arabelleğine ilişkin gösterge.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```

Sonlandırma-MQ_TERM_EXIT

MQ_TERM_EXIT, MQXF_TERM ve ExitReason MQXR_CONNECTION işlev tanıtıcısıyla kayıtlı bağlantı düzeyi sonlandırması sağlar. Kayıtlıysa, her bağlantı kesme isteği için MQ_TERM_EXIT bir kez çağrılır.

Sonlandırmanın bir parçası olarak, çıkış için depolama alanı artık gerekli değildir ve temizleme işlemi gerçekleştirilebilir.

MQXCC_FAILED ile bir MQ_TERM_EXIT başarısız olursa, kuyruk yöneticisi MQCC_FAILED ve MQRC_API_EXIT_ERROR ile çağrılan MQDISC ' den döner.

Kuyruk yöneticisi, son MQ_TERM_EXIT çağrıldıktan sonra API çıkış işlevi yürütme ortamını sonlandırırken bir hata saptarsa, kuyruk yöneticisi MQCC_FAILED ve MQRC_API_EXIT_TERM_ERROR ile MQ_TERM_EXIT çağrısını çağırır MQDISC çağrısından döner.

Bu işleve ilişkin arabirim:

```
MQ_TERM_EXIT (&ExitParms, &ExitContext, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ise, çıkış işlevi neden kodu alanını herhangi bir geçerli MQRC_* değerine ayarlayabilir.

Uygulamaya döndürülen CompCode ve Nedeni, MQAXP ' deki ExitResponse alanının değerine bağlıdır.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
MQAXP      ExitParms;      /* Exit parameter structure */
MQAXC      ExitContext;    /* Exit context structure */
MQLONG     CompCode;      /* Completion code */
MQLONG     Reason;        /* Reason code */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_TERM_EXIT (&ExitParms, &ExitContext, &CompCode, &Reason)
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
void MQENTRY MQ_TERM_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 */
```

Kullanım notları

1. MQ_TERM_EXIT işlevi isteğe bağlıdır. Sonlandırılacak bir işlem yoksa, bir çıkış takımının sonlandırma çıkışını kaydettirmesi gerekmez.

Çıkış takımına ait işlevler bağlantı sırasında kaynak alıyorsa, MQ_TERM_EXIT işlevi, bu kaynakların serbest bırakılması (örneğin, devingen olarak elde edilen saklama alanının serbest bırakılması) için uygun bir noktadır.

2. MQDISC çağrısı yayınlandığında bir MQ_TERM_EXIT işlevi kaydedilirse, tüm MQDISC çıkış işlevleri çağrıldıktan sonra çıkış işlevi çağrılır.
3. MQ_TERM_EXIT, MQAXP 'nin ExitResponse alanında MQXCC_FAILED değerini döndürürse ya da başka bir şekilde başarısız olursa, MQ_TERM_EXIT 'in çağrılmasına neden olan MQDISC çağrısı da başarısız olur; **CompCode** ve **Reason** parametreleri uygun değerlere ayarlanır.

Aboneliği kaydet-MQ_SUB_EXIT

MQ_SUB_EXIT, *bundan önce* ve *bundan sonra* aboneliği yeniden kaydetmeyi gerçekleştirmek için bir çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_SUB işlev tanıtıcısını kullanarak *bundan önce* ve *bundan sonra* abonelik kayıt çağrısı çıkış işlevlerini kaydedin.

Bu işleve ilişkin arabirim:

```
MQ_SUB_EXIT (&ExitParms, &ExitContext, &Hconn, &pSubDesc, &pHobj, &pHsub, &CompCode, &Reason)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş/çıkış

Bağlantı tanıtıcısı.

pSubGiriş/çıkış

Öznitelik seçicilerin dizisi.

pHobj -giriş/çıkış

Nesne tanıtıcısı

pHsub (MQHOBJ) giriş/çıkış

Abonelik tanıtıcısı

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
MQAXP    ExitParms;    /* Exit parameter structure */
```

```

MQAXC      ExitContext;    /* Exit context structure */
MQHCONN    Hconn;         /* Connection handle */
PMQSD      pSubDesc;      /* Subscription descriptor */
PMQHOBJS   pHobj;        /* Object Handle */
PMQHOBJS   pSub;         /* Subscription handle */
MQLONG     CompCode;     /* Completion code */
MQLONG     Reason;       /* Reason code qualifying completion code */

```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```

MQ_SUB_EXIT (&ExitParms, &ExitContext, &Hconn, &pSubDesc, &pHobj, &pSub,
             &CompCode, &Reason);

```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```

PMQAXP      pExitParms;    /* Exit parameter structure */
PMQAXC      pExitContext;  /* Exit context structure */
PMQHCONN    pHconn;       /* Connection handle */
PPMQSD      ppSubDesc;    /* Subscription descriptor */
PPMQHOBJS   ppHobj;       /* Object Handle */
PPMQHOBJS   ppSub;        /* Subscription handle */
PMQLONG     pCompCode;    /* Completion code */
PMQLONG     pReason;      /* Reason code qualifying completion code */

```

Abonelik isteği-MQ_SUBRQ_EXIT

MQ_SUBRQ_EXIT, abonelik isteği işlendikten *önce* ve *sonra* gerçekleştirmek için bir abonelik isteği çıkış işlevi sağlar. MQXF_SUBRQ işlev tanıtıcısını, MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle kullanarak *bundan önce* ve *bundan sonra* abonelik isteği çağrı çıkışı işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```

MQ_SUBRQ_EXIT (&ExitParms, &ExitContext, &Hconn, &pSub, &Action, &pSubRqOpts,
              &CompCode, &Reason)

```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş/çıkış

Bağlantı tanıtıcısı.

pHsub (MQHOBJS) giriş/çıkış

Abonelik tanıtıcısı

İşlem (MQLONG) giriş/çıkış

Eylem

pSubRqOpts (MQSRO) giriş/çıkış

CompCode (MQLONG)-giriş/çıkış

Tamamlanma kodu, geçerli değerler şunlardır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu

Neden (MQLONG)-giriş/çıkış

Tamamlanma kodunu niteleyen neden kodu.

Tamamlanma kodu MQCC_OK ise, tek geçerli değer:

MQRC_NONE

(0, x '000') Rapor etmek için bir neden yok.

Tamamlanma kodu MQCC_FAILED ya da MQCC_WARNING ise, çıkış işlevi neden kodu alanını geçerli bir MQRC_* değerine ayarlayabilir.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
MQ_SUBRQ_EXIT (&ExitParms, &ExitContext, &Hconn, &pHsub, &Action, &pSubRqOpts,
               &CompCode, &Reason);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
PPMQHOBJS 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, xa_close işlemlerinden önce ve sonra gerçekleştirilecek bir xa_close çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XACLOSE işlev tanıtıcısını kullanarak, xa_close çağrısından önce ve sonra çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
XA_CLOSE_EXIT (&ExitParms, &ExitContext, &Hconn, &pXa_info, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXa_info (PMQCHAR)-giriş/çıkış

Yönetim ortamına özgü kaynak yöneticisi bilgileri.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_CLOSE_EXIT (&ExitParms, &ExitContext, &Hconn, &pXa_info, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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, xa_commit işlemesinden önce ve sonra gerçekleştirilecek bir xa_commit çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XACOMMIT işlev tanıtıcısını kullanarak, xa_commit çağırısından önce ve sonra çıkış işlevlerini kaydettirir.

Bu işleve ilişkin arabirim:

```
XA_COMMIT_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXID (MQPTR)-giriş/çıkış

İşlem dalı tanıtıcısı.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```

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 */

```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_COMMIT_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```

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, xa_complete işlemesinden önce ve sonra gerçekleştirilecek bir xa_complete çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XACOMPLETE işlev tanıtıcısını kullanarak, xa_complete çağırından önce ve sonra çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
XA_COMPLETE_EXIT (&ExitParms, &ExitContext, &Hconn, &pHandle, &pRetVal, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pHandle (PMQLONG)-giriş/çıkış

Zamanuyumsuz işleme işaret eden gösterge.

pRetVal (PMQLONG)-giriş/çıkış

Zamanuyumsuz işlemin dönüş değeri.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```

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 */

```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```

XA_COMPLETE_EXIT (&ExitParms, &ExitContext, &Hconn, &pHandle, &pRetval, &Rmid, &Flags,
&XARetCode);

```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```

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, xa_end işlemesinden önce ve sonra gerçekleştirilecek bir xa_end çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XAEND işlev tanıtıcısını kullanarak, xa_end çağırısından önce ve sonra çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```

XA_END_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)

```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXID (MQPTR)-giriş/çıkış

İşlem dalı tanıtıcısı.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```

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 */

```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_END_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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_unut-XA_FORGET_EXIT

XA_FORGET_EXIT, xa_unut işlemesinden önce ve sonra gerçekleştirilecek bir xa_unut çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XAFORGET işlev tanıtıcısını kullanarak xa_unut çağrı çıkışı işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
XA_FORGET_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXID (MQPTR)-giriş/çıkış

İşlem dalı tanıtıcısı.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_FORGET_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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, xa_open işlemesinden önce ve sonra gerçekleştirilecek bir xa_open çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XAOPEN işlev tanıtıcısını kullanarak, xa_open çağrısından önce ve sonra çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
XA_OPEN_EXIT (&ExitParms, &ExitContext, &Hconn, &pXa_info, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXa_info (PMQCHAR)-giriş/çıkış

Yönetim ortamına özgü kaynak yöneticisi bilgileri.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_OPEN_EXIT (&ExitParms, &ExitContext, &Hconn, &pXa_info, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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, xa_prepare işlemesinden önce ve sonra gerçekleştirilecek bir xa_prepare çıkış işlevi sağlar. xa_prepare çağrı çıkışı işlevlerini kaydettirmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XAPREPARE işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```
XA_PREPARE_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXID (MQPTR)-giriş/çıkış

İşlem dalı tanıtıcısı.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_PREPARE_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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, xa_recover komutunun işlenmesinden önce ve sonra gerçekleştirilecek bir xa_recover çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XARECOVER işlev tanıtıcısını kullanarak, xa_recover çağrısından önce ve sonra çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
XA_RECOVER_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Count, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXID (MQPTR)-giriş/çıkış

İşlem dalı tanıtıcısı.

Sayı (MQLONG)-giriş/çıkış

XID dizisindeki XID sayısı üst sınırı

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_RECOVER_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Count, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
    PMQPTR  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, xa_rollback işleminin önce ve sonra gerçekleştirilecek bir xa_rollback çıkış işlevi sağlar. xa_rollback çağrısından önce ve sonra çıkış işlevlerini kaydettirmek için MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XAROLLBACK işlev tanıtıcısını kullanın.

Bu işleve ilişkin arabirim:

```
XA_ROLLBACK_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXID (MQPTR)-giriş/çıkış

İşlem dalı tanıtıcısı.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_ROLLBACK_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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, xa_start işleminin önce ve sonra gerçekleştirilecek bir xa_start çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_XASTART işlev tanıtıcısını kullanarak, xa_start çağrısından önce ve sonra çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
XA_START_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXID (MQPTR)-giriş/çıkış

İşlem dalı tanıtıcısı.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
XA_START_EXIT (&ExitParms, &ExitContext, &Hconn, &pXID, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
  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_reg-AX_REG_EXIT

AX_REG_EXIT, ax_reg işlemesinden önce ve sonra gerçekleştirilecek bir ax_reg çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_AXREG işlev tanıtıcısını kullanarak, ax_reg çağrısından önce ve sonra çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
AX_REG_EXIT (&ExitParms, &ExitContext, &pXID, &Rmid, &Flags, &XARetCode)
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Hconn (MQHCONN)-giriş

Bağlantı tanıtıcısı.

pXID (MQPTR)-giriş/çıkış

İşlem dalı tanıtıcısı.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
AX_REG_EXIT (&ExitParms, &ExitContext, &pXID, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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, ax_unreg işlemesinden önce ve sonra gerçekleştirilecek bir ax_unreg çıkış işlevi sağlar. MQXR_BEFORE ve MQXR_AFTER çıkış nedenleriyle MQXF_AXUNREG işlev tanıtıcısını kullanarak, ax_unreg çağırısından önce ve sonra çıkış işlevlerini kaydettirin.

Bu işleve ilişkin arabirim:

```
AX_UNREG_EXIT (&ExitParms, &ExitContext, &Rmid, &Flags, &XARetCode);
```

burada parametreler şunlardır:

ExitParms (MQAXP)-giriş/çıkış

Parametre yapısından çık.

ExitContext (MQAXC)-giriş/çıkış

Bağlam yapısından çık.

Rmid (MQLONG)-giriş/çıkış

Kaynak yöneticisi tanıtıcısı.

İşaretler (MQLONG)-giriş/çıkış

Kaynak yöneticisi seçenekleri.

XARetCode (MQLONG)-giriş/çıkış

XA aramasından yanıt.

C dili çağırma

Kuyruk yöneticisi aşağıdaki değişkenleri mantıksal olarak tanımlar:

```
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 */
```

Kuyruk yöneticisi daha sonra, çıkışı aşağıdaki gibi mantıksal olarak çağırır:

```
AX_UNREG_EXIT (&ExitParms, &ExitContext, &Rmid, &Flags, &XARetCode);
```

Çıkışınız aşağıdaki C işlevi prototipiyle eşleşmelidir:

```
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 */
```

Çıkış işlevlerinin çağırılmasıyla ilgili genel bilgiler

Bu konu, özellikle hataların ve beklenmeyen olayların işlenmesiyle ilgili olarak, çıkışlarınızı planlamanıza yardımcı olacak genel bir kılavuz sağlar.

Çıkış hatası

Bir çıkış işlevi yıkıcı, uyumsuz bir MQGET çağrısından sonra olağandışı bir şekilde sonlandırılırsa, ancak ileti uygulamaya iletilmeden önce, çıkış işleyicisi hatadan kurtulup denetimi uygulamaya geçirebilir.

Bu durumda, ileti kaybolabilir. Bu, bir uygulama kuyruktan ileti aldıktan hemen sonra başarısız olduğunda ne olur gibi bir şey.

MQGET çağrısı MQCC_FAILED ve MQRC_API_EXIT_ERROR ile tamamlanabilir.

Bir *before* API çağrısı çıkış işlevi olağandışı bitirse, çıkış işleyicisi başarısızlıktan kurtulabilir ve denetimi API çağrısını işlemeyen uygulamaya geçirebilir. Bu durumda, çıkış işlevinin sahip olduğu kaynakları kurtarması gerekir.

Zincirleme çıkışlar kullanılırsa, başarıyla yönlendirilen herhangi bir *bundan önce* API çağrısı çıkışına ilişkin *bundan sonra* API çağrısı çıkışları kendileri yönlendirilebilir. API çağrısı MQCC_FAILED ve MQRC_API_EXIT_ERROR ile başarısız olabilir.

Çıkış işlevleri için örnek hata işleme

Aşağıdaki çizgede noktalar gösterilmektedir (e N) hataların ortaya çıkabileceği bir yerde. Bu yalnızca çıkışların nasıl davrandığını ve aşağıdaki çizelgeyle birlikte okunması gerektiğini gösteren bir örnektir. Bu örnekte, zincirleme çıkışlarla davranışı göstermek için her API çağrısından önce ve sonra iki çıkış işlevi çağrılır.

Application	ErrPt	Exit function	API call
-----	-----	-----	-----
Start			
MQCONN	-->		
	e1		
		MQ_INIT_EXIT	
	e2		

```

before MQ_CONNX_EXIT 1
e3 before MQ_CONNX_EXIT 2
e4 --> MQCONN
e5 after MQ_CONNX_EXIT 2
e6 after MQ_CONNX_EXIT 1
e7
MQOPEN <--
--> before MQ_OPEN_EXIT 1
e8 before MQ_OPEN_EXIT 2
e9 --> MQOPEN
e10 after MQ_OPEN_EXIT 2
e11 after MQ_OPEN_EXIT 1
e12
MQPUT <--
--> before MQ_PUT_EXIT 1
e13 before MQ_PUT_EXIT 2
e14 --> MQPUT
e15 after MQ_PUT_EXIT 2
e16 after MQ_PUT_EXIT 1
e17
MQCLOSE <--
--> before MQ_CLOSE_EXIT 1
e18 before MQ_CLOSE_EXIT 2
e19 --> MQCLOSE
e20 after MQ_CLOSE_EXIT 2
e21 after MQ_CLOSE_EXIT 1
e22
MQDISC <--
--> before MQ_DISC_EXIT 1
e23 before MQ_DISC_EXIT 2
e24 --> MQDISC
e25 after MQ_DISC_EXIT 2
e26 after MQ_DISC_EXIT 1
e27
<--
end

```

Aşağıdaki çizelge, her hata noktasında yapılacak işlemleri listeler. Burada gösterilen kurallar diğerlerine uygulanabileceğinden, yalnızca hata noktalarının bir alt kümesi kapsandı. Bu, her bir vakada amaçlanan davranışı belirten işlemidir.

Çizelge 837. API çıkış hataları ve yapılacak uygun işlemler

Err Pt	Açıklama	İşlemler
e1	Ortam kurulumu ayarlanırken hata oluştu.	<ol style="list-style-type: none"> 1. Ortam kurulumunu gerektiği gibi geri al 2. Sürücü çıkış işlevi yok 3. MQCC_FAILED, MQRC_API_EXIT_LOAD_ERROR ile MQCONN başarısız oldu
e2	MQ_INIT_EXIT işlevi şununla tamamlanır: <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için: <ol style="list-style-type: none"> 1. Ortamı temizle 2. MQCC_FAILED, MQRC_API_EXIT_INIT_ERROR ile MQCONN başarısız oldu • MQXCC_* için <ol style="list-style-type: none"> 1. MQXCC_* ve MQXR2_*¹ değerleri için işlem yapın 2. Ortamı temizle
e3	Önce MQ_CONNX_EXIT 1 işlevi şununla tamamlanır: <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için: <ol style="list-style-type: none"> 1. Drive MQ_TERM_EXIT işlevi 2. Ortamı temizle 3. MQCC_FAILED, MQRC_API_EXIT_ERROR ile MQCONN çağrısı başarısız oldu • MQXCC_* için <ol style="list-style-type: none"> 1. MQXCC_* ve MQXR2_*¹ değerleri için işlem yapın 2. Gerekliyse, sürücü MQ_TERM_EXIT işlevi 3. Gerekliyse ortamı temizle
e4	Önce MQ_CONNX_EXIT 2 işlevi aşağıdaki işlemlerle tamamlanır: <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için: <ol style="list-style-type: none"> 1. Drive <i>bundan sonra</i> MQ_CONNX_EXIT 1 işlevi 2. Drive MQ_TERM_EXIT işlevi 3. Ortamı temizle 4. MQCC_FAILED, MQRC_API_EXIT_ERROR ile MQCONN çağrısı başarısız oldu • MQXCC_* için <ol style="list-style-type: none"> 1. MQXCC_* ve MQXR2_*¹ değerleri için işlem yapın 2. Çıkış engellenmediyse <i>bundan sonra</i> MQ_CONNX_EXIT 1 işlevi 3. Gerekliyse, sürücü MQ_TERM_EXIT işlevi 4. Gerekliyse ortamı temizle

Çizelge 837. API çıkış hataları ve yapılacak uygun işlemler (devamı var)

Err Pt	Açıklama	İşlemler
e5	MQCONN çağrısı başarısız oldu.	<ol style="list-style-type: none"> 1. MQCONN CompCode ve Nedeni Geçir 2. <i>bundan önce</i> MQ_CONNX_EXIT 2 başarılı olursa ve çıkış engellenmezse <i>bundan sonra</i> MQ_CONNX_EXIT 2 işlevi 3. <i>bundan önce</i> MQ_CONNX_EXIT 1 başarılı olursa ve çıkış engellenmezse <i>bundan sonra</i> MQ_CONNX_EXIT 1 işlevi 4. Drive MQ_TERM_EXIT işlevi 5. Ortamı temizle
e6	<p><i>Sonra</i> MQ_CONNX_EXIT 2 işlevi şununla tamamlanır:</p> <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için: <ol style="list-style-type: none"> 1. Drive <i>bundan sonra</i> MQ_CONNX_EXIT 1 işlevi 2. MQCC_FAILED, MQRC_API_EXIT_ERROR ile MQCONN çağrısı tamamlayın • MQXCC_* için <ol style="list-style-type: none"> 1. MQXCC_* ve MQXR2_*¹ değerleri için işlem yapın 2. Sürücü <i>bundan sonra</i> MQ_CONNX_EXIT 1 işlevi gerekiyorsa
e7	<p><i>Sonra</i> MQ_CONNX_EXIT 1 işlevi şununla tamamlanır:</p> <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için, MQCC_FAILED, MQRC_API_EXIT_ERROR ile MQCONN çağrısı tamamlayın • MQXCC_* için, MQXCC_* ve MQXR2_*¹ değerlerini kullanın
e8	<p><i>Önce</i> MQ_OPEN_EXIT 1 işlevi şununla tamamlanır:</p> <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için, MQCC_FAILED, MQRC_API_EXIT_ERROR ile MQOPEN çağrısı tamamlayın • MQXCC_* için, MQXCC_* ve MQXR2_*¹ değerlerini kullanın
e9	<p><i>Önce</i> MQ_OPEN_EXIT 2 işlevi şununla tamamlanır:</p> <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için: <ol style="list-style-type: none"> 1. Drive <i>bundan sonra</i> MQ_OPEN_EXIT 1 işlevi 2. MQCC_FAILED, MQRC_API_EXIT_ERROR ile eksiksiz MQOPEN çağrısı • MQXCC_* için, MQXCC_* ve MQXR2_*¹ değerlerini kullanın
e10	MQOPEN çağrısı başarısız oldu	<ol style="list-style-type: none"> 1. MQOPEN CompCode ve Nedeni Geçir 2. Çıkış engellenmediyse <i>bundan sonra</i> MQ_OPEN_EXIT 2 işlevi 3. <i>bundan sonra</i> MQ_OPEN_EXIT 1 sürücüsü çıkışı engellenmediyse ve zincirleme çıkışlar engellenmediyse

Çizelge 837. API çıkış hataları ve yapılacak uygun işlemler (devamı var)		
Err Pt	Açıklama	İşlemler
e1 1	Sonra MQ_OPEN_EXIT 2 işlevi şununla tamamlanır: <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için: <ol style="list-style-type: none"> 1. Drive <i>bundan sonra</i> MQ_OPEN_EXIT 1 işlevi 2. MQCC_FAILED, MQRC_API_EXIT_ERROR ile eksiksiz MQOPEN çağrısı • MQXCC_* için <ol style="list-style-type: none"> 1. MQXCC_* ve MQXR2_*¹ değerleri için işlem yapın 2. Çıkış engellenmezse <i>bundan sonra</i> MQ_OPEN_EXIT 1 işlevi
e2 5	Sonra MQ_DISC_EXIT 2 işlevi şununla tamamlanır: <ul style="list-style-type: none"> • MQXCC_FAILED • MQXCC_* 	<ul style="list-style-type: none"> • MQXCC_FAILED için: <ol style="list-style-type: none"> 1. Sürücü , MQ_DISC_EXIT 1 işlevinden sonra 2. Drive MQ_TERM_EXIT işlevi 3. Çıkış yürütme ortamını temizle 4. MQCC_FAILED, MQRC_API_EXIT_ERROR ile eksiksiz MQDISC çağrısı • MQXCC_* için <ol style="list-style-type: none"> 1. MQXCC_* ve MQXR2_*¹ değerleri için işlem yapın 2. Drive MQ_TERM_EXIT işlevi 3. Çıkış yürütme ortamını temizle

Not:

1. MQXCC_* ve MQXR2_* değerleri ve bunlara karşılık gelen işlemler Kuyruk yöneticilerinin çıkış işlevlerini nasıl işlediği alanında tanımlanır.

ExitResponse alanları yanlış ayarlandı

Bu konuda, ExitResponse alanı desteklenen değerler dışında bir değere ayarlandığında ne olacağı hakkında bilgi verilir.

ExitResponse (ExitResponse) alanı, desteklenen değerlerden biri dışında bir değere ayarlandıysa, aşağıdaki işlemler geçerlidir:

- *bundan önce* MQCONN ya da MQDISC API çıkış işlevi için:
 - ExitResponse2 değeri yoksayılr.
 - Çıkış zincirinde (varsa) çıkış işlevlerinden başka *önce* çağrılmaz; API çağrısı kendisine yayınlanmaz.
 - Başarılı bir şekilde çağrılan *before* çıkışları için, *after* (sonraki) çıkışları ters sırayla çağrılır.
 - Kayıtlıysa, zincirdeki başarıyla çağrılan *bundan önce* MQCONN ya da MQDISC çıkış işlevlerine ilişkin sonlandırma çıkış işlevleri, bu çıkış işlevlerinden sonra temizlenmek üzere yönlendirilir.
 - MQCONN ya da MQDISC çağrısı MQRC_API_EXIT_ERROR ile başarısız oluyor.
- MQCONN ya da MQDISC dışındaki bir *için* IBM MQ API çıkış işlevinden önce:
 - ExitResponse2 değeri yoksayılr.
 - Çıkış zincirinde (varsa) başka *bundan önce* ya da *bundan sonra* veri dönüştürme işlevi çağrılmaz.
 - Başarılı bir şekilde çağrılan *before* çıkışları için, *after* (sonraki) çıkışları ters sırayla çağrılır.
 - IBM MQ API çağrısını kendisi yayınlanmadı.
 - IBM MQ API çağrısı MQRC_API_EXIT_ERROR ile başarısız olur.

- *bundan sonra* MQCONN ya da MQDISC API çıkış işlevi için:
 - ExitResponse2 değeri yoksayılr.
 - API çağrılmadan önce başarıyla çağrılan kalan çıkış işlevleri ters sırayla çağrılır.
 - Kayıtlıysa, başarıyla çağrılan zincirdeki *bundan önce* ya da *bundan sonra* MQCONN ya da MQDISC çıkış işlevleri için sonlandırma çıkışı işlevleri, çıkıştan sonra temizlenecek şekilde yönlendirilir.
 - MQCC_WARNING ve çıkışın döndürdüğü CompCode daha ciddi bir CompCode , uygulamaya döndürülür.
 - Uygulamaya bir MQRC_API_EXIT_ERROR nedeni döndürüldü.
 - IBM MQ API çağrısı başarıyla yayınlandı.
 - MQCONN ya da MQDISC dışında bir *bundan sonra* IBM MQ API çağrı çıkışı işlevi için:
 - ExitResponse2 değeri yoksayılr.
 - API çağrılmadan önce başarıyla çağrılan kalan çıkış işlevleri ters sırayla çağrılır.
 - MQCC_WARNING ve çıkışın döndürdüğü CompCode daha ciddi bir CompCode , uygulamaya döndürülür.
 - Uygulamaya bir MQRC_API_EXIT_ERROR nedeni döndürüldü.
 - IBM MQ API çağrısı başarıyla yayınlandı.
 - Get exit işlevinde *before* veri dönüştürmesi için:
 - ExitResponse2 değeri yoksayılr.
 - API çağrılmadan önce başarıyla çağrılan kalan çıkış işlevleri ters sırayla çağrılır.
 - İleti dönüştürülmez ve dönüştürülmemiş ileti uygulamaya döndürülür.
 - MQCC_WARNING ve çıkışın döndürdüğü CompCode daha ciddi bir CompCode , uygulamaya döndürülür.
 - Uygulamaya bir MQRC_API_EXIT_ERROR nedeni döndürüldü.
 - IBM MQ API çağrısı başarıyla yayınlandı.
- Not:** Hata çıkışta olduğu için, MQRC_API_EXIT_ERROR döndürmek, MQRC_NOT_CONVERTED döndürmekten daha iyidir.

Bir çıkış işlevi ExitResponse2 alanını desteklenen değerlerden biri dışında bir değere ayarlarsa, bunun yerine MQXR2_DEFAULT_CONTINUATION değeri kabul edilir.

Kurulabilir hizmetler arabirimi başvuru bilgileri

Bu konu derlemi, kurulabilir hizmetlere ilişkin başvuru bilgilerini içerir.

İşlevler ve veri tipleri, her hizmet tipi için grup içinde alfabetik sırayla listelenir.

İlgili kavramlar

ALW [UNIX, Linux ve Windows için kurulabilir hizmetler ve bileşenler](#)

IBM i [IBM i için kurulabilir hizmetler ve bileşenler](#)

IBM i [IBMi için kurulabilir hizmetler arabirimi başvuru bilgileri](#)

İlgili görevler

[Kuyruk yöneticisi olanaklarını genişletme](#)

ALW [Kurulabilir hizmetlerin yapılandırılması](#)

İşlevlerin nasıl gösterileceği

Kurulabilir hizmet işlevlerinin nasıl belgelendiğini.

Her işlev için, işlev tanıtıcısını (MQZEP için) içeren bir tanım vardır.

Değiştirgeler , gerçekleştirmeleri gereken sırayla gösterilir. Hepsi orada olmalı.

Her parametre adını kendi veri tipi izler. Bunlar, [“Temel veri tipleri” sayfa 234](#) içinde açıklanan temel veri tipleridir.

Parametrelerin açıklamasından sonra C dili çağırısı da verilir.

MQZ_AUTHENTICATE_USER-Kullanıcının Kimliğini Doğrula

Bu işlev bir MQZAS_VERSION_5 yetkilendirme hizmeti bileşeni tarafından sağlanır ve bir kullanıcının kimliğini doğrulamak ya da kimlik bağlamı alanlarını ayarlamak için kuyruk yöneticisi tarafından çağrılır. IBM MQ kullanıcı uygulaması bağlamı oluşturulduğunda çağrılır.

Uygulama bağlamı, uygulamanın kullanıcı bağlamının başlatıldığı noktada ve uygulamanın kullanıcı bağlamının değiştirildiği her noktada bağlantı çağrıları sırasında oluşturulur. Her bağlanma çağırısı yapıldığında, *IdentityContext* alanında uygulamanın kullanıcı bağlamı bilgileri yeniden alınır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_AUTHENTICATE_USER.

Sözdizimi

MQZ_AUTHENTICATE_USER (*QMgrName* , *SecurityParms* , *ApplicationContext* , *IdentityContext* , *CorrelationPtr* , *ComponentData* , *Devam* , *CompCode* , *Reason* (Neden))

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

SecurityParms

Tip: MQCSP-giriş

Güvenlik parametreleri. Kullanıcı kimliği, parola ve kimlik doğrulama tipiyle ilgili veriler. MQCSP yapısının AuthenticationType özniteliği MQCSP_AUTH_USER_ID_AND_PWD olarak belirtilirse, kullanıcı kimliği ve parola, eşleşip eşleşmediklerini belirlemek için IdentityContext (MQZIC) değiştirgesindeki eşdeğer alanlarla karşılaştırılır. Daha fazla bilgi için bkz [“MQCSP-Güvenlik değiştirgeleri” sayfa 334](#).

MQCONN MQI çağırısı sırasında bu değiştirge boş değer ya da varsayılan değerler içeriyor.

ApplicationContext

Tip: MQZAC-giriş

Uygulama bağlamı. Çağıran uygulamayla ilgili veriler. Ayrıntılar için bkz. [MQZAC-Uygulama bağlamı](#) .

Her MQCONN ya da MQCONNX MQI çağırısı sırasında, MQZAC yapısındaki kullanıcı bağlamı bilgileri yeniden alınır.

IdentityContext

Tip: MQZIC-giriş/çıkış

Kimlik bağlamı. Kimlik doğrulama kullanıcı işlevine girişte bu, geçerli kimlik bağlamını tanımlar. Kullanıcı kimliğini doğru işlevi bunu değiştirebilir; bu noktada kuyruk yöneticisi yeni kimlik bağlamını benimser. MQZIC yapısıyla ilgili daha fazla ayrıntı için bkz. [MQZIC-Identity context](#) .

CorrelationPtr

Tip: MQPTR-çıkış

İlinti göstergesi. İlinti verilerinin adresini belirler. Bu işaretçi daha sonra diğer OAM çağrılarında aktarılır.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının ComponentDataLength deęiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Devam işareti. Aşağıdaki deęerleri belirtebilirsiniz:

MQZCI_DEFAULT

Dięer bileşenlere baęlı olarak devam.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki deęerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağrısı

```
MQZ_AUTHENTICATE_USER (QMgrName, SecurityParms, ApplicationContext,
                        IdentityContext, &CorrelationPtr, ComponentData,
                        &Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiştirgeleri aşağıdaki gibi bildirin:

```
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-Yetki denetimi

Bu işlev bir MQZAS_VERSION_1 yetkilendirme hizmeti bileşeni tarafından sağlanır ve bir varlığın belirli bir nesne üzerinde belirli bir işlem ya da işlem gerçekleştirme yetkisi olup olmadığını denetlemek için kuyruk yöneticisi tarafından başlatılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_CHECK_AUTHORITY.

Sözdizimi

MQZ_CHECK_AUTHORITY(QMgrName , EntityName , EntityType , ObjectName , ObjectType , Authority , ComponentData , Continuation , CompCode , Reason)

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

EntityName

Tip: MQCHAR12 -giriş

Varlık adı. Nesne için yetkisi denetlenecek varlığın adı. Dizginin uzunluk üst sınırı 12 karakterdir; uzunluktan daha kısaysa, sağ tarafta boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

Bu varlığın temel güvenlik hizmeti tarafından bilinmesi gerekli değildir. Bilinmezse, denetim için özel **hiç kimse** grubunun (tüm varlıkların ait olduğu varsayılır) yetkileri kullanılır. Tümü boş bir ad geçerlidir ve bu şekilde kullanılabilir.

EntityType

Tip: MQLONG-input

Varlık tipi. EntityName tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRP

Grup.

ObjectName

Tip: MQCHAR48 -giriş

Nesne adı. Erişim gerektiren nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

Authority

Tip: MQLONG-input

Denetlenecek yetki. Bir yetki denetleniyorsa, bu alan uygun yetkilendirme işlemine (MQZAO_ * değişmezi) eşittir. Birden çok yetki denetleniyorsa, bu, ilgili MQZAO_ * değişmezlerinin bit bakımından OR işlecidir.

MQI çağrılarının kullanımı için aşağıdaki yetkiler geçerlidir:

MQZAO_CONNECT

MQCONN çağrısı kullanma yeteneği.

MQZAO_GÖZ AT

MQGET çağrısı bir göz atma seçeneğiyle kullanma yeteneği.

Bu, MQGET çağrısında MQGMO_BROWSE_FIRST, MQGMO_BROWSE_MSG_UNDER_CURSOR ya da MQGMO_BROWSE_NEXT seçeneğinin belirtilmesini sağlar.

MQZAO_INPUT

Müdür bey. MQGET çağrısı bir giriş seçeneğiyle kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_INPUT_SHARED, MQOO_INPUT_EXCLUSIVE ya da MQOO_INPUT_AS_Q_DEF seçeneğinin belirtilmesini sağlar.

MQZAO_OUTPUT

MQPUT çağrısı kullanma yeteneği.

MQOPEN çağrısında MQOO_OUTPUT seçeneğinin belirtilmesini sağlar.

MQZAO_INQUIRE

MQINQ çağrısı kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_INQUIRE seçeneğinin belirtilmesini sağlar.

MQZAO_KÜMESİ

MQSET çağrısı kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_SET seçeneğinin belirtilmesini sağlar.

MQZAO_PASS_IDENTITY_CONTEXT

Kimlik bağlamını geçme yeteneği.

Bu, MQOPEN çağrısında MQOO_PASS_IDENTITY_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_PASS_IDENTITY_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_PASS_ALL_CONTEXT

Tüm bağlamı geçme becerisi.

Bu, MQOPEN çağrısında MQOO_PASS_ALL_CONTEXT seçeneğinin belirtilmesini ve MQPUT ve MQPUT1 çağrılarında MQPMO_PASS_ALL_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_SET_IDENTITY_CONTEXT

Kimlik bağlamını ayarlama yeteneği.

Bu, MQOPEN çağrısında MQOO_SET_IDENTITY_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_SET_IDENTITY_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_SET_ALL_CONTEXT

Tüm bağlamı ayarlama yeteneği.

Bu, MQOPEN çağrısında MQOO_SET_ALL_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_SET_ALL_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_ALTERNATE_USER_AUTHORITY

Diğer kullanıcı yetkisini kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_ALTERNATE_USER_AUTHORITY seçeneğinin ve MQPUT1 çağrısında MQPMO_ALTERNATE_USER_AUTHORITY seçeneğinin belirtilmesini sağlar.

MQZAO_ALL_MQI

Tüm MQI yetkileri.

Bu, tüm yetkileri etkinleştirir.

Bir kuyruk yöneticisinin yönetimi için aşağıdaki yetkiler geçerlidir:

MQZAO_OLUŞTURMA

Belirtilen tipte nesnelere yaratma yeteneği.

MQZAO_DELETE

Belirtilen bir nesneyi silme yeteneği.

MQZAO_DISPLAY

Belirtilen bir nesnenin özniteliklerini görüntüleme yeteneği.

MQZAO_CHANGE

Belirtilen bir nesnenin özniteliklerini değiştirme yeteneği.

MQZAO_CLEAR

Belirtilen bir kuyruktan tüm iletileri silme yeteneği.

MQZAO_AUTHORIZE

Belirli bir nesne için diğer kullanıcılara yetki verme yeteneği.

MQZAO_CONTROL

Bir dinleyici, hizmet ya da istemci dışı kanal nesnesini başlatma ya da durdurma ve istemci olmayan bir kanal nesnesine ping komutu gönderebilme yeteneği.

MQZAO_CONTROL_EXTENDED

Bir sıra numarasını sıfırlama ya da istemci olmayan bir kanal nesnesindeki belirsiz bir iletiyi çözme yeteneği.

MQZAO_ALL_ADMIN

Kimlik bağlamını ayarlama yeteneği.

MQZAO_CREATE dışındaki tüm denetim yetkileri.

Aşağıdaki yetkiler hem MQI kullanımı hem de bir kuyruk yöneticisinin yönetimi için geçerlidir:

MQZAO_Tümü

MQZAO_CREATE dışındaki tüm yetkiler.

MQZAO_NONE

Yetki yok.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlemlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlemlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değeriyle değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

Bir bileşene çağrı başarısız olursa (*CompCode* , MQCC_FAILED değerini döndürür) ve *Devam* parametresi MQZCI_DEFAULT ya da MQZCI_CONTINUE ise, kuyruk yöneticisi varsa diğer bileşenleri çağdırmaya devam eder.

Çağrı başarılı olursa (*CompCode* , MQCC_OK değerini döndürür), *Devam* ayarı ne olursa olsun başka hiçbir bileşen çağdırılmaz.

Çağrı başarısız olursa ve *Devam* parametresi MQZCI_STOP ise, başka hiçbir bileşen çağdırılmaz ve hata kuyruk yöneticisine döndürülür. Bileşenlerin önceki çağrılar hakkında bilgisi olmadığı için, *Devam* parametresi her zaman çağrıdan önce MQZCI_DEFAULT olarak ayarlanır.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağrısı

```
MQZ_CHECK_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
Objectype, Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen değıştirgeler aşağıdaki gibi bildirilir:

```

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_AUTHORITY_2 -Yetki denetimi (genişletilmiş)

Bu işlev bir MQZAS_VERSION_2 yetkilendirme hizmeti bileşeni tarafından sağlanır ve bir varlığın belirli bir nesne üzerinde belirli bir işlem ya da işlem gerçekleştirme yetkisi olup olmadığını denetlemek için kuyruk yöneticisi tarafından başlatılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_CHECK_AUTHORITY.

MQZ_CHECK_AUTHORITY_2 , MQZ_CHECK_AUTHORITY gibi, ancak **EntityName** parametresi **EntityData** parametresiyle değiştirildi.

Sözdizimi

MQZ_CHECK_AUTHORITY_2(*QMgrName* , *EntityData* , *EntityType* , *ObjectName* , *ObjectType* , *Authority* , *ComponentData* , *Continuation* , *CompCode* , *Reason*)

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

EntityData

Tip: MQZED-giriş

Varlık verileri. Denetlenecek nesne için yetkilendirmeye sahip varlıkla ilgili veriler. Ayrıntılar için bkz. [“MQZED-Varlık tanımlayıcı” sayfa 1651.](#)

Bu varlığın temel güvenlik hizmeti tarafından bilinmesi gerekli değildir. Bilinmezse, denetim için özel **hiç kimse** grubunun (tüm varlıkların ait olduğu varsayılır) yetkileri kullanılır. Tümü boş bir ad geçerlidir ve bu şekilde kullanılabilir.

EntityType

Tip: MQLONG-input

Varlık tipi. *EntityData* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRP

Grup.

ObjectName

Tip: MQCHAR48 -giriş

Nesne adı. Erişim gerektiren nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmaz.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

Authority

Tip: MQLONG-input

Denetlenecek yetki. Bir yetki denetleniyorsa, bu alan uygun yetkilendirme işlemine (MQZAO_* değişmezi) eşittir. Birden çok yetki denetleniyorsa, bu, ilgili MQZAO_* değişmezlerinin bit bakımından OR işlecidir.

MQI çağrılarının kullanımı için aşağıdaki yetkiler geçerlidir:

MQZAO_CONNECT

MQCONN çağrısı kullanma yeteneği.

MQZAO_GÖZ AT

MQGET çağrısı bir göz atma seçeneğiyle kullanma yeteneği.

Bu, MQGET çağrısında MQGMO_BROWSE_FIRST, MQGMO_BROWSE_MSG_UNDER_CURSOR ya da MQGMO_BROWSE_NEXT seçeneğinin belirtilmesini sağlar.

MQZAO_INPUT

Müdür bey. MQGET çağrısı bir giriş seçeneğiyle kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_INPUT_SHARED, MQOO_INPUT_EXCLUSIVE ya da MQOO_INPUT_AS_Q_DEF seçeneğinin belirtilmesini sağlar.

MQZAO_OUTPUT

MQPUT çağrısı kullanma yeteneği.

MQOPEN çağrısında MQOO_OUTPUT seçeneğinin belirtilmesini sağlar.

MQZAO_INQUIRE

MQINQ çağrısı kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_INQUIRE seçeneğinin belirtilmesini sağlar.

MQZAO_KÜMESİ

MQSET çağrısı kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_SET seçeneğinin belirtilmesini sağlar.

MQZAO_PASS_IDENTITY_CONTEXT

Kimlik bağlamını geçme yeteneği.

Bu, MQOPEN çağrısında MQOO_PASS_IDENTITY_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_PASS_IDENTITY_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_PASS_ALL_CONTEXT

Tüm bağlamı geçme becerisi.

Bu, MQOPEN çağrısında MQOO_PASS_ALL_CONTEXT seçeneğinin belirtilmesini ve MQPUT ve MQPUT1 çağrılarında MQPMO_PASS_ALL_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_SET_IDENTITY_CONTEXT

Kimlik bağlamını ayarlama yeteneği.

Bu, MQOPEN çağrısında MQOO_SET_IDENTITY_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_SET_IDENTITY_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_SET_ALL_CONTEXT

Tüm bağlamı ayarlama yeteneği.

Bu, MQOPEN çağrısında MQOO_SET_ALL_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_SET_ALL_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_ALTERNATE_USER_AUTHORITY

Diğer kullanıcı yetkisini kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_ALTERNATE_USER_AUTHORITY seçeneğinin ve MQPUT1 çağrısında MQPMO_ALTERNATE_USER_AUTHORITY seçeneğinin belirtilmesini sağlar.

MQZAO_ALL_MQI

Tüm MQI yetkileri.

Bu, tüm yetkileri etkinleştirir.

Bir kuyruk yöneticisinin yönetimi için aşağıdaki yetkiler geçerlidir:

MQZAO_OLUŞTURMA

Belirtilen tipte nesnelere yaratma yeteneği.

MQZAO_DELETE

Belirtilen bir nesneyi silme yeteneği.

MQZAO_DISPLAY

Belirtilen bir nesnenin özniteliklerini görüntüleme yeteneği.

MQZAO_CHANGE

Belirtilen bir nesnenin özniteliklerini değiştirme yeteneği.

MQZAO_CLEAR

Belirtilen bir kuyruktan tüm iletileri silme yeteneği.

MQZAO_AUTHORIZE

Belirli bir nesne için diğer kullanıcılara yetki verme yeteneği.

MQZAO_CONTROL

Bir dinleyici, hizmet ya da istemci dışı kanal nesnesini başlatma ya da durdurma ve istemci olmayan bir kanal nesnesine ping komutu gönderebilme yeteneği.

MQZAO_CONTROL_EXTENDED

Bir sıra numarasını sıfırlama ya da istemci olmayan bir kanal nesnesindeki belirsiz bir iletiyi çözme yeteneği.

MQZAO_ALL_ADMIN

Kimlik bağlamını ayarlama yeteneği.

MQZAO_CREATE dışındaki tüm denetim yetkileri.

Aşağıdaki yetkiler hem MQI kullanımı hem de bir kuyruk yöneticisinin yönetimi için geçerlidir:

MQZAO_Tümü

MQZAO_CREATE dışındaki tüm yetkiler.

MQZAO_NONE

Yetki yok.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlemlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlemlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değıştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşğıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşğıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağrısı

MQZ_CHECK_AUTHORITY_2 (QMgıName, &EntityData, EntityType,

```
ObjectName, ObjectType, Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Kullanıcının ayrıcalıklı olup olmadığını denetleyin

Bu işlev bir MQZAS_VERSION_6 yetkilendirme hizmeti bileşeni tarafından sağlanır ve belirtilen kullanıcının ayrıcalıklı bir kullanıcı olup olmadığını saptamak için kuyruk yöneticisi tarafından çağrılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_CHECK_PRIVILEGED.

Sözdizimi

```
MQZ_CHECK_PRIVILEGED( QMgrName , EntityData , EntityType , ComponentData ,  
Continuation , CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

EntityData

Tip: MQZED-giriş

Varlık verileri. Denetlenecek varlıkla ilgili veriler. Daha fazla bilgi için bkz. [“MQZED-Varlık tanımlayıcı” sayfa 1651.](#)

EntityType

Tip: MQLONG-input

Varlık tipi. EntityDatatarafından belirtilen varlık tipi. Ařaęıdaki deęerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRUP

Grup.

ComponentData

Tip: MQBYTExComponentDataLength -giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan deęiřiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılıřında sunulur.

Bu veri alanının uzunluęu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** deęiřtirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

Bir bileşene çağrı başarısız olursa (*CompCode* , MQCC_FAILED değerini döndürür) ve *Devam* parametresi MQZCI_DEFAULT ya da MQZCI_CONTINUE ise, kuyruk yöneticisi varsa diğer bileşenleri çağdırmaya devam eder.

Çağrı başarılı olursa (*CompCode* , MQCC_OK değerini döndürür), *Devam* ayarı ne olursa olsun başka hiçbir bileşen çağrıılmaz.

Çağrı başarısız olursa ve *Devam* parametresi MQZCI_STOP ise, başka hiçbir bileşen çağrıılmaz ve hata kuyruk yöneticisine döndürülür. Bileşenlerin önceki çağrılar hakkında bilgisi olmadığı için, *Devam* parametresi her zaman çağrıdan önce MQZCI_DEFAULT olarak ayarlanır.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_PRIVILEGED

(2584, X'A18') Bu kullanıcı ayrıcalıklı bir kullanıcı kimliği değil.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağrısı

```
MQZ_CHECK_PRIVILEGED (QMgrName, &EntityData, EntityType,  
ComponentData, &Continuation,  
&CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Tüm yetkiyi kopyala

Bu iřlev bir yetkilendirme hizmeti bileřeni tarafından saęlanır. Kuyruk yöneticisi tarafından, bir bařvuru nesnesi için yürürlükte olan tüm yetkileri bařka bir nesneye kopyalamak için bařlatılır.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP için) MQZID_COPY_ALL_AUTHORITY 'dir.

Sözdizimi

MQZ_COPY_ALL_AUTHORITY(QMgrName , RefObjectName , ObjectName , ObjectType , ComponentData , Continuation , CompCode , Reason)

Parametreler

QMgrName

Tip: MQCHAR48 -giriř

Kuyruk yöneticisi adı. Bileřeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileřene aktarılır; yetkilendirme hizmeti arabirimi, bileřenin bileřeni herhangi bir řekilde kullanmasını gerektirmez.

RefObjectAdı

Tip: MQCHAR48 -giriř

Bařvuru nesnesi adı. Referans nesnesinin adı, kopyalanacak yetkiler. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, saęa boşluklarla doldurulur. Ad boş deęerli bir karakterle sonlandırılmaz.

ObjectName

Tip: MQCHAR48 -giriř

Nesne adı. Eriřimleri ayarlanacak nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, saęa boşluklarla doldurulur. Ad boş deęerli bir karakterle sonlandırılmaz.

ObjectType

Tip: MQLONG-input

Nesne tipi. *RefObjectName* ve *ObjectName* tarafından belirtilen varlık tipi. Ařaęıdaki deęerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doęrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci baęlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

ComponentData

Tip: MQBYTEExComponentDataLength -giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının ComponentDataLength değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

MQRC_UNKNOWN_REF_OBJECT

(2294, X'8F6') Başvuru nesnesi bilinmiyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_COPY_ALL_AUTHORITY (QMgrName, RefObjectName, ObjectName, ObjectType,  
                          ComponentData, &Continuation, &CompCode,  
                          &Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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-Silme yetkisi

Bu işlev bir yetkilendirme hizmeti bileşeni tarafından sağlanır ve kuyruk yöneticisi tarafından, belirtilen nesneyle ilişkili tüm yetkileri silmek için başlatılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_DELETE_AUTHORITY.

Sözdizimi

```
MQZ_DELETE_AUTHORITY( QMgrName , ObjectName , ObjectType , ComponentData ,  
Continuation , CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

ObjectName

Tip: MQCHAR48 -giriş

Nesne adı. Erişimleri silinecek nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının ComponentDataLength değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_DELETE_AUTHORITY (QMgrName, ObjectName, ObjectType, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
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-Yetki verilerini numaralandırma

Bu işlev bir MQZAS_VERSION_4 yetkilendirme hizmeti bileşeni tarafından sağlanır ve kuyruk yöneticisi tarafından, ilk çağrıda belirtilen seçim ölçütleriyle eşleşen tüm yetki verilerini almak için arka arkaya başlatılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_ENUMERATE_AUTHORITY_DATA.

Sözdizimi

```
MQZ_ENUMERATE_AUTHORITY_DATA( QMgrName , StartEnumeration , Filter ,  
AuthorityBufferLength , AuthorityBuffer , AuthorityDataLength , ComponentData ,  
Continuation , CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

StartEnumeration

Tip: MQLONG-input

Aramanın numaralandırmayı başlatıp başlatamayacağını gösteren işaret. Bu, çağrının yetki verilerinin numaralandırmasını başlatıp başlatamayacağını ya da önceki bir MQZ_ENUMERATE_AUTHORITY_DATA çağrısıyla başlatılan yetki verilerinin numaralandırmasına devam edip edemeyeceğini belirtir. Değer, aşağıdaki değerlerden biridir:

MQZSE_START

Numaralandırmayı başlat. Çağrı, yetki verilerinin numaralandırmasını başlatmak için bu değerle başlatılır. **Filter** parametresi, bu ve ardışık çağrılar tarafından döndürülen yetki verilerini seçmek için kullanılacak seçim ölçütlerini belirtir.

MQZSE_CONTINUE

Numaralandırmaya devam et. Yetki verileri sıralamasına devam etmek için çağrı bu değerle başlatılır. Bu durumda **Filter** değiştirgesi yoksayılr ve boş değerli gösterge olarak belirtilebilir (seçim ölçütleri, *StartEnumeration* ' u MQZSE_START olarak ayarlı olan çağrı tarafından belirtilen **Filter** değiştirgesiyle belirlenir).

Filtrele

Tip: MQZAD-giriş

Filtrele. *StartEnumeration* MQZSE_START ise, *Filter* döndürülecek yetki verilerini seçmek için kullanılacak seçim ölçütlerini belirtir. *Filter* boş değerli işaretçiye, seçim ölçütü kullanılmaz; yani, tüm yetki verileri döndürülür. Kullanılabilecek seçim ölçütlerine ilişkin ayrıntılar için bkz. "MQZAD-Yetki verileri" sayfa 1648 .

StartEnumeration MQZSE_CONTINUE ise, *Filter* yoksayılr ve boş değerli gösterge olarak belirtilebilir.

AuthorityBufferUzunluğu

Tip: MQLONG-input

AuthorityBuffer uzunluğu. Bu, **AuthorityBuffer** parametresinin bayt cinsinden uzunluğudur. Yetki arabelleği, döndürülecek verileri alabilecek büyüklükte olmalıdır.

AuthorityBuffer

Tip: MQZAD-çıkış

Yetki verileri. Bu, yetki verilerinin döndürüldüğü arabellektir. Arabellek, MQZAD yapısını, MQZED yapısını ve en uzun varlık adını ve tanımlanan en uzun etki alanı adını barındıracak kadar büyük olmalıdır.

Not: Not: Bu değiştirge MQZAD olarak tanımlanır; MQZAD arabelleğin başında her zaman oluşur. Ancak, arabellek MQZAD olarak bildirildiyse, arabellek çok küçük olur; MQZAD, MQZED ve varlık ve etki alanı adlarını barındırabilmesi için arabellek bir MQZAD ' dan büyük olmalıdır.

AuthorityDataUzunluğu

Tip: MQLONG-output

AuthorityBuffer içinde döndürülen verilerin uzunluğu. Yetki arabelleği çok küçükse, *AuthorityDataLength* gereken arabelleğin uzunluğuna ayarlanır ve çağrı tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_BUFFER_LENGTH_ERROR döndürür.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının ComponentDataLength değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_ENUMERATE_AUTHORITY_DATA için bu, MQZCI_CONTINUE ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

MQRC_NO_DATA_VAR

(2379, X'94B') Kullanılabilir veri yok.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_ENUMERATE_AUTHORITY_DATA (QMgrName, StartEnumeration, &Filter,  
                               AuthorityBufferLength,  
                               &AuthorityBuffer,  
                               &AuthorityDataLength, ComponentData,  
                               &Continuation, &CompCode,  
                               &Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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-Serbest kullanıcı

Bu işlev bir MQZAS_VERSION_5 yetki hizmeti bileşeni tarafından sağlanır ve ilişkili ayrılmış kaynağı serbest bırakmak için kuyruk yöneticisi tarafından başlatılır.

Bir uygulama tüm kullanıcı bağlamları altında (örneğin, bir MQDISC MQI çağırısı sırasında) çalışmayı bitirdiğinde başlatılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_FREE_USER.

Sözdizimi

```
MQZ_FREE_USER( QMgrName , FreeParms , ComponentData , Continuation , CompCode ,  
Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

FreeParms

Tip: MQZFP-giriş

Serbest parametreler. Serbest bırakılacak kaynakla ilgili verileri içeren bir yapı. Ayrıntılar için bkz. [“MQZFP-Serbest deęiřtirgeler” sayfa 1653.](#)

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriř/çıkıř

Bileřen verileri. Bu veriler, bu belirli bileřen adına kuyruk yöneticisi tarafından tutulur; bu bileřen tarafından saęlanan iřlevlerden herhangi biri tarafından yapılan deęiřiklikler korunur ve bu bileřen iřlevlerinden biri bir sonraki çağrılıřında sunulur.

Bu veri alanının uzunluęu, MQZ_INIT_AUTHORITY çağrısının ComponentDataLength deęiřtirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Devam iřareti. Ařaęıdaki deęerler belirtilebilir:

MQZCI_DEFAULT

Dięer bileřenlere baęlı olarak devam.

MQZCI_DURUřU

Sonraki bileřenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Ařaęıdaki deęerlerden biri olmalıdır:

MQCC_OK

Bařarıyla tamamlandı.

MQCC_FAILED

Arama bařarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete eriřilirken beklenmeyen bir hata oluřtu.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları.](#)

C çağrısı

MQZ_AUTHENTICATE_USER (QMgrName, SecurityParms, ApplicationContext,

```
IdentityContext, CorrelationPtr, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Yetki al

Bu iřlev bir MQZAS_VERSION_1 yetkilendirme hizmeti bileřeni tarafından saęlanır ve kuyruk yneticisi, birincil kullanıcının yesi olduęu gruplar tarafından sahip olunan (varlık bir birincil kullanıcıysa) yetkiler de dahil olmak zere, bir varlıęın belirtilen nesneye eriřmesi iin gereken yetkiyi almak zere kuyruk yneticisi tarafından bařlatılır. Genel tanıtımlardaki yetkiler, dndrlen yetki kmesine eklenir.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP iin) MQZID_GET_AUTHORITY.

Szdizimi

MQZ_GET_AUTHORITY(*QMgrName* , *EntityName* , *EntityType* , *ObjectName* ,
ObjectType , *Authority* , *ComponentData* , *Continuation* , *CompCode* , *Reason*)

Parametreler

QMgrName

Tip: MQCHAR48 -giriř

Kuyruk yneticisi adı. Bileřeni aęıran kuyruk yneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar bořluklarla doldurulur; ad boř karakterle sonlandırılmaz.

Kuyruk yneticisi adı, bilgi iin bileřene aktarılır; yetkilendirme hizmeti arabirimi, bileřenin bileřeni herhangi bir Őekilde kullanmasını gerektirmez.

EntityName

Tip: MQCHAR12 -giriř

Varlık adı. Nesneye eriřimi alınacak varlıęın adı. Dizginin uzunluk st sınırı 12 karakterdir; uzunluktan daha kısaysa, saę tarafta bořluklarla doldurulur. Ad boř deęerli bir karakterle sonlandırılmadı.

EntityType

Tip: MQLONG-input

Varlık tipi. *EntityName* tarafından belirtilen varlık tipi. Ařaęıdaki deęerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Mdr bey.

MQZAET_GRP

Grup.

ObjectName

Tip: MQCHAR48 -giriř

Nesne adı. Eriřimin alınacaęı nesnenin adı. Dizilimin uzunluk st sınırı 48 karakterdir; uzunluktan daha kısaysa, saęa bořluklarla doldurulur. Ad boř deęerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Ařaęıdaki deęerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

Authority

Tip: MQLONG-input

Varlığın yetkisi. Varlığın tek bir yetkisi varsa, bu alan uygun yetkilendirme işlemine (MQZAO_* değişmezi) eşittir. Birden çok yetkisi varsa, bu alan ilgili MQZAO_* sabitlerinin bit bakımından OR işlecidir.

ComponentData

Tip: MQBYTE xComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirtgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_GET_AUTHORITY için bu, MQZCI_CONTINUE ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılamıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_GET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
                   ObjectType, &Authority, ComponentData,  
                   &Continuation, &CompCode, &Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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 -Yetki al (genişletilmiş)

Bu işlev bir MQZAS_VERSION_2 yetkilendirme hizmeti bileşeni tarafından sağlanır ve bir varlığın belirtilen nesneye erişme yetkisini almak için kuyruk yöneticisi tarafından başlatılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_GET_AUTHORITY.

MQZ_GET_AUTHORITY_2 , MQZ_GET_AUTHORITY gibi, ancak **EntityName** parametresi **EntityData** parametresiyle değiştirildi.

Sözdizimi

```
MQZ_GET_AUTHORITY_2( QMgrName , EntityData , EntityType , ObjectName ,  
ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )
```


Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

EntityData

Tip: MQZED-giriş

Varlık verileri. Nesne için yetkilendirmenin alınacağı varlıkla ilgili veriler. Ayrıntılar için bkz. [“MQZED-Varlık tanımlayıcı” sayfa 1651.](#)

EntityType

Tip: MQLONG-input

Varlık tipi. *EntityData* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRUP

Grup.

ObjectName

Tip: MQCHAR48 -giriş

Nesne adı. Varlık yetkisinin alınacağı nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

Authority

Tip: MQLONG-input

Varlığın yetkisi. Varlığın tek bir yetkisi varsa, bu alan uygun yetkilendirme işlemine (MQZAO_* değişmezi) eşittir. Birden çok yetkisi varsa, bu alan ilgili MQZAO_* sabitlerinin bit bakımından OR işlecidir.

ComponentData

Tip: MQBYTE xComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılamıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_GET_AUTHORITY_2 (QMgrName, &EntityData, EntityType, ObjectName,  
Objectype, &Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQZED EntityData;          /* Entity data */  
MQLONG EntityType;        /* Entity type */  
MQCHAR48 ObjectName;      /* Object name */  
MQLONG Objectype;        /* 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-Belirtik yetkiyi al

Bu iřlev bir MQZAS_VERSION_1 yetkilendirme hizmeti bileřeni tarafından saęlanır ve kuyruk yneticisi, birincil kullanıcının yesi olduęu gruplar tarafından sahip olunan (varlık bir birincil kullanıcıysa) yetkiler de dahil olmak zere, bir varlıęın belirtilen nesneye eriřmesi iin gereken yetkiyi almak zere kuyruk yneticisi tarafından bařlatılır. Genel tanıtımlardaki yetkiler, dndrlen yetki kmesine eklenir.

AIX and Linux zerinde, yerleřik IBM MQ nesne yetkisi yneticisi (OAM) iin, dndrlen yetki yalnızca birincil kullanıcının birincil grubunun sahip olduęu yetkilidir.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP iin) MQZID_GET_BELIRIT_AUTHORITY.

Szdizimi

```
MQZ_GET_EXPLICIT_AUTHORITY( QMgrName , EntityName , EntityType , ObjectName ,  
Objectype , Authority , ComponentData , Continuation , CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriř

Kuyruk yneticisi adı. Bileřeni aęıran kuyruk yneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar bořluklarla doldurulur; ad boř karakterle sonlandırılmaz.

Kuyruk yneticisi adı, bilgi iin bileřene aktarılır; yetkilendirme hizmeti arabirimi, bileřenin bileřeni herhangi bir Őekilde kullanmasını gerektirmez.

EntityName

Tip: MQCHAR12 -giriř

Varlık adı. Nesneye eriřimin alınacaęı varlıęın adı. Dizginin uzunluk st sınırı 12 karakterdir; uzunluktan daha kısaysa, saę tarafta bořluklarla doldurulur. Ad boř deęerli bir karakterle sonlandırılmadı.

EntityType

Tip: MQLONG-input

Varlık tipi. *EntityName* tarafından belirtilen varlık tipi. Ařaęıdaki deęerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Mdr bey.

MQZAET_GRP

Grup.

ObjectName

Tip: MQCHAR48 -giriş

Nesne adı. Varlık yetkisinin alınacağı nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

Authority

Tip: MQLONG-input

Varlığın yetkisi. Varlığın tek bir yetkisi varsa, bu alan uygun yetkilendirme işlemine (MQZAO_* değişmezi) eşittir. Birden çok yetkisi varsa, bu alan ilgili MQZAO_* sabitlerinin bit bakımından OR işlecidir.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlemlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlemlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_GET_AUTHORITY için bu, MQZCI_CONTINUE ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılamıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_GET_EXPLICIT_AUTHORITY (QMgrName, EntityName, EntityType,  
ObjectName, ObjectType, &Authority,  
ComponentData, &Continuation,  
&CompCode, &Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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 -Belirtik yetkiyi al (genişletilmiş)

Bu işlev bir MQZAS_VERSION_2 yetki hizmeti bileşeni tarafından sağlanır ve kuyruk yöneticisi tarafından, adı belirtilen bir grubun belirtilen bir nesneye (ancak **kimse** grubunun ek yetkisi olmadan) erişmesi için

gereken yetkiyi ya da adı belirtilen birincil grubun belirtilen bir nesneye erişmesi için gereken yetkiyi almak üzere başlatılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_GET_BELIRIT_AUTHORITY.

MQZ_GET_EXPLICIT_AUTHORITY_2 , MQZ_GET_BELIRIT_AUTHORITY gibi, ancak **EntityName** değiştirgesi **EntityData** değiştirgesiyle değiştirildi.

Sözdizimi

MQZ_GET_EXPLICIT_AUTHORITY_2(*QMgrName* , *EntityData* , *EntityType* , *ObjectName* , *ObjectType* , *Authority* , *ComponentData* , *Continuation* , *CompCode* , *Reason*)

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

EntityData

Tip: MQZED-giriş

Varlık verileri. Nesne için yetkisi alınacak varlıkla ilgili veriler. Ayrıntılar için bkz. "[MQZED-Varlık tanımlayıcı](#)" sayfa 1651.

EntityType

Tip: MQLONG-input

Varlık tipi. *EntityData* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRUP

Grup.

ObjectName

Tip: MQCHAR48 -giriş

Nesne adı. Varlık yetkisinin alınacağı nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

Authority

Tip: MQLONG-input

Varlığın yetkisi. Varlığın tek bir yetkisi varsa, bu alan uygun yetkilendirme işlemine (MQZAO_* değişmezi) eşittir. Birden çok yetkisi varsa, bu alan ilgili MQZAO_* sabitlerinin bit bakımından OR işlemidir.

ComponentData

Tip: MQBYTE xComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlemlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlemlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değıştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılamıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_GET_EXPLICIT_AUTHORITY_2 (QMgrName, &EntityData, EntityType,  
ObjectName, ObjectType, &Authority,  
ComponentData, &Continuation,  
&CompCode, &Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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-Yetkilendirme hizmetini kullanıma hazırla

Bu işlev bir yetkilendirme hizmeti bileşeni tarafından sağlanır ve bileşenin yapılandırılması sırasında kuyruk yöneticisi tarafından başlatılır. Kuyruk yöneticisine bilgi sağlamak için MQZEP ' nin çağırılması beklenir.

Bu işleve ilişkin işlev tanıtcısı (MQZEP için) MQZID_INIT_AUTHORITY.

Sözdizimi

```
MQZ_INIT_AUTHORITY( Hconfig , Options , QMgrName , ComponentDataLength ,  
ComponentData , Version , CompCode , Reason )
```

Parametreler

Hconfig (Yapılandırma)

Tip: MQHCONFIG-giriş

Yapılandırma tanıtcısı. Bu tanıtcı, kullanıma hazırlanmakta olan bileşeni gösterir. MQZEP işleviyle kuyruk yöneticisi çağırılırken bileşen tarafından kullanılır.

Seçenekler

Tip: MQLONG-input

Başlatma seçenekleri. Aşağıdaki değerlerden biri olmalıdır:

MQZIO_PRIMARY

Birincil başlatma.

MQZIO_IKINCIL

İkincil başlatma.

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

ComponentDataUzunluğu

Tip: MQLONG-input

Bileşen verilerinin uzunluğu. *ComponentData* alanının bayt cinsinden uzunluğu. Bu uzunluk, bileşen yapılandırma verilerinde tanımlanır.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu, bileşen birincil kullanıma hazırlama işlevi çağrılmadan önce tüm sıfırlar için başlatılır. Bu veriler, bu bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerin (kullanıma hazırlama işlevi de içinde olmak üzere) herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Sürüm

Tip: MQLONG-giriş/çıkış

Sürüm numarası. Kullanıma hazırlama işlevine girişte, kuyruk yöneticisinin desteklediği en yüksek sürüm numarasını gösterir. Başlatma işlevi, gerekiyorsa, bunu, desteklediği arabirim sürümüyle değiştirmelidir. Dönüşte kuyruk yöneticisi, bileşenin döndürdüğü sürümü desteklemiyorsa, MQZ_TERM_AUTHORITY işlevini çağırır ve bu bileşeni artık kullanmaz.

Aşağıdaki değerler desteklenir:

MQZAS_VERSION_1

Sürüm 1.

MQZAS_VERSION_2

Sürüm 2.

MQZAS_VERSION_3

Sürüm 3.

MQZAS_VERSION_4

Sürüm 4.

MQZAS_VERSION_5

Sürüm 5.

MQZAS_VERSION_6

IBM WebSphere MQ 6.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_INITIALIZATION_FAILED

(2286, X'8EE') Tanımlanmamış bir nedenle kullanıma hazırlama başarısız oldu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_INIT_AUTHORITY (Hconfig, Options, QMgrName, ComponentDataLength,  
                    ComponentData, &Version, &CompCode,  
                    &Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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-Yetki hizmetini sorgula

Bu işlev bir MQZAS_VERSION_5 yetkilendirme hizmeti bileşeni tarafından sağlanır ve desteklenen işlevselliği sorgulamak için kuyruk yöneticisi tarafından başlatılır.

Birden çok hizmet bileşeni kullanıldığında, hizmet bileşenleri, kuruldukları sıraya göre ters sırayla çağrılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_INQUIRE 'dir.

Sözdizimi

```
MQZ_INQUIRE( QMgrName , SelectorCount , Selectors , IntAttrCount , IntAttrs ,  
CharAttrLength , CharAttrs , SelectorReturned , ComponentData , Continuation ,  
CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

SelectorCount

Tip: MQLONG-input

Seçici sayısı. **Selectors** değiştirgesinde sağlanan seçicilerin sayısı.

Değer 0-256 aralığında olmalıdır.

Seçiciler

Tip: MQLONGxSelectorSayı-giriş

Seçicilerin dizisi. Her seçici gerekli bir özniteliği tanımlar ve aşağıdakilerden biri olmalıdır:

- MQIACF_INTERFACE_VERSION (tamsayı)
- MQIACF_USER_ID_SUPPORT (tamsayı)
- MQCACF_SERVICE_COMPONENT (karakter)

Seçiciler herhangi bir sırada belirtilebilir. Dizideki seçicilerin sayısı **SelectorCount** parametresiyle gösterilir.

Seçiciler tarafından tanımlanan tamsayı öznitelikleri, **IntAttr**s parametresinde *Selectors* içinde görüldükleri sırayla döndürülür.

Seçiciler tarafından tanımlanan karakter öznitelikleri, **CharAttr**s değiştirgesinde, görüldükleri sırayla döndürülür *Selectors*.

IntAttrSayısı

Tip: MQLONG-input

IntAttr's değiştirgesinde verilen tamsayı özniteliklerinin sayısı.

Değer 0-256 aralığında olmalıdır.

IntAttr

Tip: MQLONG x IntAttrSayı-çıkış

Tamsayı öznitelikleri. Tamsayı öznitelikleri dizisi. Tamsayı öznitelikleri, *Selectors* dizisindeki karşılık gelen tamsayı seçicileri ile aynı sırada döndürülür.

CharAttrSayısı

Tip: MQLONG-input

Karakter öznitelikleri arabelleğinin uzunluğu. **CharAttr**s parametresinin bayt cinsinden uzunluğu.

Değer, en azından istenen karakter özniteliklerinin uzunluklarının toplamı olmalıdır. Herhangi bir karakter özniteliği istenmezse, sıfır geçerli bir değerdir.

CharAttr

Tip: MQLONG x CharAttrSayı-çıkış

Karakter öznitelikleri arabelleği. Karakter özniteliklerini içeren arabellek, birleştirildi. Karakter öznitelikleri, *Selectors* dizisindeki karşılık gelen karakter seçiciler ile aynı sırayla döndürülür.

Arabelleğin uzunluğu CharAttrCount parametresi tarafından verilir.

SelectorReturned

Tip: MQLONG x SelectorCount -giriş

Seçici döndürüldü. Seçiciler parametresindeki seçiciler tarafından istenen kümeden hangi özniteliklerin döndürüldüğünü tanımlayan değerler dizisi. Bu dizideki değerlerin sayısı **SelectorCount** parametresiyle gösterilir. Dizideki her değer, Seçiciler dizisindeki karşılık gelen konumdan seçiciyle ilgilidir. Her değer aşağıdakilerden biridir:

MQZSL_DÖNDÜRÜLDÜ

Selectors değiştirgesinde ilgili seçici tarafından istenen öznitelik döndürüldü.

MQZSL_NOT_DÖNDÜRÜLMEDİ

Selectors değiştirgesinde ilgili seçici tarafından istenen öznitelik döndürülmedi.

Dizi, tüm değerlerle *MQZSL_NOT_RETURNED* olarak başlatılır. Bir yetkilendirme hizmeti bileşeni bir öznitelik döndürdüğünde, dizideki uygun değeri *MQZSL_NOT_RETURNED* olarak ayarlar. Bu, sorgunun yapıldığı diğer yetkilendirme hizmeti bileşenlerinin hangi özniteliklerin önceden döndürüldüğünü tanımlamasına olanak sağlar.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değıştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşğıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşğıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_CHAR_ATTRS_TOO_SHORT

Karakter öznitelikleri için yeterli alan yok.

MQRC_INT_COUNT_TOO_SMALL

Tamsayı öznitelikleri için yeterli alan yok.

CompCode MQCC_FAILED ise:

MQRC_SELECTOR_COUNT_HATA

Seçici sayısı geçerli değil.

MQRC_SELECTOR_ERROR

Öznitelik seçici geçerli değil.

MQRC_SELECTOR_LIMIT_AŞILDI

Çok fazla seçici belirtildi.

MQRC_INT_ATTR_COUNT_HATA

Tamsayı özniteliklerinin sayısı geçerli değil.

MQRC_INT_ATTRS_ARRAY_HATA

Tamsayı öznitelikleri dizisi geçersiz.

MQRC_CHAR_ATTR_LENGTH_HATA

Karakter öznitelikleri sayısı geçersiz.

MQRC_CHAR_ATTRS_HATA

Karakter öznitelikleri dizgisi geçerli değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmet erişilirken beklenmeyen bir hata oluştu.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_INQUIRE (QMgrName, SelectorCount, Selectors, IntAttrCount,  
              &IntAttrs, CharAttrLength, &CharAttrs,  
              SelectorReturned, ComponentData, &Continuation,  
              &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Tüm yetkileri yenile

Bu iřlev bir MQZAS_VERSION_3 yetkilendirme hizmeti bileřeni tarafından saęlanır ve kuyruk yöneticisi tarafından bileřen tarafından ieride tutulan yetkiler listesini yenilemek için çağırılır.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP için) MQZID_REFRESH_CACHE (8L).

Sözdizimi

```
MQZ_REFRESH_CACHE( QMgrName , ComponentData , Continuation , CompCode ,  
Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriř

Kuyruk yöneticisi adı. Bileřeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileřene aktarılır; yetkilendirme hizmeti arabirimi, bileřenin tanımlanmış bir řekilde kullanılmasını gerektirmez.

ComponentData

Tip: MQBYTE xComponentDataUzunluk-giriř/ıkıř

Bileřen verileri. Bu veriler, kuyruk yöneticisi tarafından bu bileřen adına tutulur; bu bileřen tarafından saęlanan iřlevlerden herhangi biri tarafından yapılan deęiřiklikler korunur ve bu bileřenin iřlevlerinden biri bir sonraki çağrılıřında sunulur.

Bu veri alanının uzunluęu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** deęiřtirgesinde kuyruk yöneticisi tarafından geirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

C çağırısı

```
MQZ_REFRESH_CACHE (QMgrName, ComponentData,  
                   &Continuation, &CompCode, &Reason);
```

Değiştireleri aşağıdaki gibi bildirin:

```
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-Yetki ayarla

Bu işlev bir MQZAS_VERSION_1 yetkilendirme hizmeti bileşeni tarafından sağlanır ve bir varlığın belirtilen nesneye erişme yetkisini ayarlamak için kuyruk yöneticisi tarafından başlatılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_SET_AUTHORITY.

Not: Bu işlev, var olan yetkileri geçersiz kılar. Var olan yetkileri korumak için, bunları bu işlevle yeniden ayarlamanız gerekir.

Sözdizimi

```
MQZ_SET_AUTHORITY( QMgrName , EntityName , EntityType , ObjectName ,  
                  ObjectType , Authority , ComponentData , Continuation , CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

EntityName

Tip: MQCHAR12 -giriş

Varlık adı. Nesneye erişimin alınacağı varlığın adı. Dizginin uzunluk üst sınırı 12 karakterdir; uzunluktan daha kısaysa, sağ tarafta boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmaz.

EntityType

Tip: MQLONG-input

Varlık tipi. *EntityName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRUP

Grup.

ObjectName

Tip: MQCHAR48 -giriş

Nesne adı. Erişim gerektiren nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmaz.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

Authority

Tip: MQLONG-input

Varlığın yetkisi. Bir yetki belirleniyorsa, bu alan uygun yetkilendirme işlemine (MQZAO_ * değişmezi) eşittir. Birden çok yetki belirleniyorsa, bu alan ilgili MQZAO_ * değişmezlerinin bit bakımından OR işlecidir.

ComponentDatarname>

Tip: MQBYTEComponentDataLength -giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_GET_AUTHORITY için bu, MQZCI_CONTINUE ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılamıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_SET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
ObjectTypes, Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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 -Yetki ayarla (geniřletilmiř)

Bu iřlev bir MQZAS_VERSION_2 yetkilendirme hizmeti bileřeni tarafından saęlanır ve bir varlıęın belirtilen nesneye eriřme yetkisini ayarlamak iin kuyruk yneticisi tarafından bařlatılır.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP iin) MQZID_SET_AUTHORITY.

Not: Bu iřlev, var olan yetkileri geersiz kılar. Var olan yetkileri korumak iin, bunları bu iřlevle yeniden ayarlamanız gerekir.

MQZ_SET_AUTHORITY_2 , MQZ_SET_AUTHORITY gibi, ancak **EntityName** parametresi **EntityData** parametresiyle deęiřtirildi.

Szdizimi

```
MQZ_SET_AUTHORITY_2( QMgrName , EntityData , EntityType , ObjectName ,  
ObjectTypes , Authority , ComponentData , Continuation , CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriř

Kuyruk yneticisi adı. Bileřeni aęıran kuyruk yneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar bořluklarla doldurulur; ad boř karakterle sonlandırılmaz.

Kuyruk yneticisi adı, bilgi iin bileřene aktarılır; yetkilendirme hizmeti arabirimi, bileřenin bileřeni herhangi bir řekilde kullanmasını gerektirmez.

EntityData

Tip: MQZED-giriř

Varlık verileri. Nesne iin yetkilendirmesi ayarlanacak varlıkla ilgili veriler. Ayrıntılar iin bkz. "[MQZED-Varlık tanımlayıcı](#)" sayfa 1651.

EntityType

Tip: MQLONG-input

Varlık tipi. *EntityData* tarafından belirtilen varlık tipi. Ařaęıdaki deęerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Mdr bey.

MQZAET_GRUP

Grup.

ObjectName

Tip: MQCHAR48 -giriş

Nesne adı. Varlık yetkisinin ayarlanacağı nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi. *ObjectName* tarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

MQOT_KONU

Konu.

Authority

Tip: MQLONG-input

Varlığın yetkisi. Bir yetki belirleniyorsa, bu alan uygun yetkilendirme işlemine (MQZAO_ * değişmezi) eşittir. Birden çok yetki belirleniyorsa, bu alan ilgili MQZAO_ * değişmezlerinin bit bakımından OR işlemdir.

ComponentData

Tip: MQBYTE xComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlemlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlemlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılamıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_SET_AUTHORITY_2 (QMgrName, &EntityData, EntityType, ObjectName,  
ObjectTypes, Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Yetkilendirme hizmetini sonlandır

Bu iřlev bir yetkilendirme hizmeti bileřeni tarafından saęlanır ve artık bu bileřenin hizmetlerini gerektirmedięi zaman kuyruk yoneticisi tarafından bařlatılır. İřlev, bileřenin gerektirdięi tüm temizleme iřlemlerini geręekleřtirmelidir.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_TERM_AUTHORITY.

Sözdizimi

MQZ_TERM_AUTHORITY(*Hconfig* , *Options* , *QMgrName* , *ComponentData* , *CompCode* , *Reason*)

Parametreler

Hconfig (Yapılandırma)

Tip: MQHCONFIG-giriş

Yapılandırma tanıtıcısı. Bu tanıtıcı, sonlandırılmakta olan bileşeni gösterir. MQZEP işleviyle kuyruk yöneticisi çağrılırken bileşen tarafından kullanılır.

Seçenekler

Tip: MQLONG-input

Sonlandırma seçenekleri. Aşağıdaki değerlerden biri olmalıdır:

MQZTO_BİRİNCİL

Birincil sonlandırma.

MQZTO_IKİNCİL

İkincil sonlandırma.

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısındaki ComponentDataLength değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

MQZ_TERM_AUTHORITY çağrısı tamamlandığında, kuyruk yöneticisi bu verileri atar.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

MQRC_TERMINATION_FAILED

(2287, X'8FF') Sonlandırma, tanımlanmamış bir nedenle başarısız oldu.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_TERM_AUTHORITY (Hconfig, Options, QMgrName, ComponentData,  
&CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Adı sil

Bu iřlev bir ad hizmeti bileřeni tarafından saęlanır ve kuyruk yneticisi tarafından, belirtilen kuyruk iin bir giriři silmek zere bařlatılır.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP iin) MQZID_DELETE_NAME.

Szdizimi

```
MQZ_DELETE_NAME( QMgrName , QName , ComponentData , Continuation , CompCode ,  
Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriř

Kuyruk yneticisi adı. Bileřeni aęıran kuyruk yneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar bořluklarla doldurulur; ad boř karakterle sonlandırılmaz.

Kuyruk yneticisi adı, bilgi iin bileřene aktarılır; yetkilendirme hizmeti arabirimi, bileřenin bileřeni herhangi bir řekilde kullanmasını gerektirmez.

QName

Tip: MQCHAR48 -giriř

Kuyruk adı. Bir giriřin silineceęi kuyruęun adı. Bu ad, parametrenin tam uzunluęuna kadar bořluklarla doldurulur; ad boř karakterle sonlandırılmaz.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriř/ıkıř

Bileřen verileri. Bu veriler, bu belirli bileřen adına kuyruk yneticisi tarafından tutulur; bu bileřen tarafından saęlanan iřlevlerden herhangi biri tarafından yapılan deęiřiklikler korunur ve bu bileřen iřlevlerinden biri bir sonraki aęırılıřında sunulur.

Bu veri alanının uzunluęu, MQZ_INIT_NAME aęırısındaki ComponentDataLength deęiřtirgesinde kuyruk yneticisi tarafından geirilir.

Devam

Tip: MQLONG-output

Bileřen tarafından ayarlanan devam gstergesi. Ařaęıdaki deęerlerden biri olmalıdır:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

MQZ_DELETE_NAME komutu için, kuyruk yöneticisi **Continuation** değiştirilmesinde ne döndürülürse döndürülsün, başka bir bileşeni başlatma girişiminde bulunmaz.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Uyarı (kısmi tamamlama).

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_UNKNOWN_NAME

(2288, X'8F0') Kuyruk adı bulunamadı.

Not: Temel hizmet bu vaka için başarılı bir şekilde yanıt verirse, bu kod döndürülemeyebilir.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_DELETE_NAME (QMgrName, QName, ComponentData, &Continuation,  
&CompCode, &Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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-Ad hizmetini kullanıma hazırla

Bu işlev bir ad hizmeti bileşeni tarafından sağlanır ve bileşenin yapılandırılması sırasında kuyruk yöneticisi tarafından başlatılır. Kuyruk yöneticisine bilgi sağlamak için MQZEP ' nin çağrılması beklenir.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_INIT_NAME.

Sözdizimi

MQZ_INIT_NAME(Hconfig , Options , QMgrName , ComponentDataLength , ComponentData , Version , CompCode , Reason)

Parametreler

Hconfig (Yapılandırma)

Tip: MQHCONFIG-giriş

Yapılandırma tanıtıcısı. Bu tanıtıcı, kullanıma hazırlanmakta olan bileşeni gösterir. MQZEP işleviyle kuyruk yöneticisi çağrılırken bileşen tarafından kullanılır.

Seçenekler

Tip: MQLONG-input

Başlatma seçenekleri. Aşağıdaki değerlerden biri olmalıdır:

MQZIO_PRIMARY

Birincil başlatma.

MQZIO_IKINCIL

İkincil başlatma.

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

ComponentDataUzunluğu

Tip: MQLONG-input

Bileşen verilerinin uzunluğu. *ComponentData* alanının bayt cinsinden uzunluğu. Bu uzunluk, bileşen yapılandırma verilerinde tanımlanır.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu, bileşen birincil kullanıma hazırlama işlevi çağrılmadan önce tüm sıfırlar için başlatılır. Bu veriler, bu bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerin (kullanıma hazırlama işlevi de içinde olmak üzere) herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Sürüm

Tip: MQLONG-giriş/çıkış

Sürüm numarası. Kullanıma hazırlama işlevine girişte, kuyruk yöneticisinin desteklediği en yüksek sürüm numarasını gösterir. Başlatma işlevi, gerekiyorsa, bunu, desteklediği arabirim sürümüyle değiştirmelidir. Dönüşte kuyruk yöneticisi, bileşenin döndürdüğü sürümü desteklemiyorsa, bileşen MQZ_TERM_NAME işlevini çağırır ve bu bileşeni artık kullanmaz.

Aşağıdaki değerler desteklenir:

MQZAS_VERSION_1

Sürüm 1.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_INITIALIZATION_FAILED

(2286, X'8EE') Tanımlanmamış bir nedenle kullanıma hazırlama başarısız oldu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_INIT_NAME (Hconfig, Options, QMgrName, ComponentDataLength,  
ComponentData, &Version, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Ad ekle

Bu iřlev bir ad hizmeti bileřeni tarafından saęlanır ve kuyruk yneticisi tarafından, kuyruęun iyesi olan kuyruk yneticisinin adını ieren bir giriř eklemek zere bařlatılır. Kuyruk hizmette nceden tanımlanmıřsa, aęrı bařarısız olur.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP iin) MQZID_INSERT_NAME.

Szdizimi

```
MQZ_INSERT_NAME( QMgrName , QName , ResolvedQMgrName , ComponentData ,  
Continuation , CompCode , Reason )
```


Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

QName

Tip: MQCHAR48 -giriş

Kuyruk adı. Bir girişin ekleneceği kuyruğun adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

ResolvedQMgrAdı

Tip: MQCHAR48 -giriş

Çözülmüş kuyruk yöneticisi adı. Kuyruğun çözüldüğü kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

ComponentData

Tip: MQBYTE xComponentDataUzunluk-giriş/çıkış

Bileşen verileri. Bu veriler, bu bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerin (kullanıma hazırlama işlevi de içinde olmak üzere) herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_NAME çağrısının **ComponentDataLength** değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-giriş/çıkış

Bileşen tarafından ayarlanan devam göstergesi. MQZ_INSERT_NAME için, kuyruk yöneticisi **Continuation** değiştirgesinde döndürülen her neyse, başka bir bileşeni başlatmayı denemez.

Aşağıdaki değerler desteklenir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_Q_ALREADY_EXISTS

(2290, X'8F2') Kuyruk nesnesi zaten var.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_INSERT_NAME (QMgrName, QName, ResolvedQMgrName, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Arama adı

Bu iřlev bir ad hizmeti bileřeni tarafından saęlanır ve kuyruk yoneticisi tarafından, belirtilen bir kuyruk iin sahip olan kuyruk yoneticisinin adını almak uzere bařlatılır.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP iin) MQZID_LOOKUP_NAME.

Sözdizimi

```
MQZ_LOOKUP_NAME( QMgrName , QName , ResolvedQMgrName , ComponentData ,  
Continuation , CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriř

Kuyruk yoneticisi adı. Bileřeni aęıran kuyruk yoneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar bořluklarla doldurulur; ad boř karakterle sonlandırılmaz.

Kuyruk yoneticisi adı, bilgi iin bileřene aktarılır; yetkilendirme hizmeti arabirimi, bileřenin bileřeni herhangi bir řekilde kullanmasını gerektirmez.

QName

Tip: MQCHAR48 -giriř

Kuyruk adı. Bir giriřin özömleneceęi kuyruęun adı. Bu ad, parametrenin tam uzunluęuna kadar bořluklarla doldurulur; ad boř karakterle sonlandırılmaz.

ResolvedQMgrAdı

Tip: MQCHAR48 -ıkıř

özölmüř kuyruk yoneticisi adı. İřlev bařarıyla tamamlanırsa, bu, kuyruęun iyeyi olan kuyruk yoneticisinin adıdır.

Hizmet bileřeni tarafından döndürölen ad, parametrenin tam uzunluęuna kadar bořluklarla saęda doldurulacaktır; ad boř karakterle sonlandırılmamalı ya da bařında ya da iinde bořluk bulunmamalıdır.

ComponentData

Tip: MQBYTEXComponentDataLength -giriş/çıkış

Bileşen verileri. Bu veriler, bu bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerin (kullanıma hazırlama işlevi de içinde olmak üzere) herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_NAME çağrısının **ComponentDataLength** değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. MQZ_LOOKUP_NAME için kuyruk yöneticisi, aşağıdaki gibi başka bir ad hizmeti bileşenin başlatılıp başlatılmayacağını belirtir:

- *CompCode* MQCC_OK ise, başka bileşen başlatılmazsa, *Devam* alanında hangi değer döndürülürse o değer döndürülür.
- *CompCode* MQCC_OK değilse, *Continuation* MQZCI_STOP değilse başka bir bileşen başlatılır.

Aşağıdaki değerler desteklenir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

MQRC_UNKNOWN_Q_NAME

(2288, X'8F0') Kuyruk adı bulunamadı.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_LOOKUP_NAME (QMgrName, QName, ResolvedQMgrName, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Ad hizmetini sonlandır

Bu iřlev bir ad hizmeti bileřeni tarafından saęlanır ve artık bu bileřenin hizmetlerini gerektirmedięi zaman kuyruk yoneticisi tarafından bařlatılır. İřlev, bileřenin gerektirdięi tüm temizleme iřlemlerini geręekleřtirmelidir.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP için) MQZID_TERM_NAME.

Sözdizimi

```
MQZ_TERM_NAME( Hconfig , Options , QMgrName , ComponentData , CompCode ,  
Reason )
```

Parametreler

Hconfig (Yapılandırma)

Tip: MQHCONFIG-giriř

Yapılandırma tanıtıcısı. Bu tanıtıcı, sonlandırılmakta olan bileřeni gösterir. Bileřen, kuyruk yoneticisini MQZEP iřleviyle çağırırken kullanır.

Seęenekler

Tip: MQLONG-input

Sonlandırma seęenekleri. Ařaęıdaki deęerlerden biri olmalıdır:

MQZTO_BİRİNCİL

Birincil sonlandırma.

MQZTO_İKİNCİL

İkincil sonlandırma.

QMgrName

Tip: MQCHAR48 -giriř

Kuyruk yoneticisi adı. Bileřeni çağırın kuyruk yoneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yoneticisi adı, bilgi için bileřene aktarılır; yetkilendirme hizmeti arabirimi, bileřenin bileřeni herhangi bir řekilde kullanmasını gerektirmez.

ComponentData

Tip: MQBYTE x ComponentDataUzunluk-giriř/çıkıř

Bileřen verileri. Bu veriler, bu bileřen adına kuyruk yoneticisi tarafından tutulur; bu bileřen tarafından saęlanan iřlevlerin (kullanıma hazırlama iřlevi de içinde olmak üzere) herhangi biri tarafından yapılan deęiřiklikler korunur ve bu bileřen iřlevlerinden biri bir sonraki çağırılıřında sunulur.

Bileřen verileri, tüm süreçlerin eriřebileceęi paylařılan bellekte.

Bu veri alanının uzunluğu, MQZ_INIT_NAME çağrısının **ComponentDataLength** deęiřtirgesinde kuyruk yöneticisi tarafından geirilir.

MQZ_TERM_NAME çağrısı tamamlandığında, kuyruk yöneticisi bu verileri atar.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Ařaęıdaki deęerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_TERMINATION_FAILED

(2287, X'8FF') Sonlandırma, tanımlanmamış bir nedenle başarısız oldu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağrısı

```
MQZ_TERM_NAME (Hconfig, Options, QMgrName, ComponentData, &CompCode,
                &Reason);
```

Hizmete geirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Uygulama baęlamı

MQZAC yapısı, *ApplicationContext* parametresi için MQZ_AUTHENTICATE_USER çağrısında kullanılır. Bu parametre, çağırılan uygulamayla ilgili verileri belirtir.

Tablo 1, yapıdaki alanları özetler.

Çizelge 838. MQZAC içindeki alanlar	
Alan	Açıklama
<u>StrucId</u>	Yapı tanıtıcısı
<u>Sürüm</u>	Yapı sürümü numarası
<u>ProcessId</u>	Süreç tanıtıcısı
<u>ThreadId</u>	İř parçacıęı tanıtıcısı

Çizelge 838. MQZAC içindeki alanlar (devamı var)	
Alan	Açıklama
<u>ApplName</u>	Uygulama Adı
<u>UserID</u>	Kullanıcı kimliği
<u>EffectiveUserKimliği</u>	Etkin kullanıcı kimliği
<u>Ortam</u>	Çevre
<u>CallerType</u>	Çağırın tipi
<u>AuthenticationType</u>	Kimlik doğrulama tipi
<u>BindType</u>	Bağ Tipi

Alanlar

StrucId

Tip: MQCHAR4 -giriş

Yapı tanıtıcısı. Değer aşağıdaki gibidir:

MQZAC_STRUC_ID

Uygulama bağlamı yapısına ilişkin tanıtıcı.

C programlama dili için, MQZAC_STRUC_ID_ARRAY değişmezi de tanımlanır; bu değer MQZAC_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Sürüm

Tip: MQLONG-input

Yapı sürümü numarası. Değer aşağıdaki gibidir:

MQZAC_VERSION_1

Version-1 uygulama bağlamı yapısı. MQZAC_CURRENT_VERSION değişmezi, yürürlükteki sürümün sürüm numarasını belirtir.

ProcessId

Tip: MQPID-giriş

Uygulamanın süreç tanıtıcısı.

ThreadId

Tip: MQTID-giriş

Uygulamanın iş parçacığı tanıtıcısı.

ApplName

Tip: MQCHAR28 -giriş

Uygulama adı.

UserID

Tip: MQCHAR12 -giriş

Kullanıcı kimliği. AIX and Linux üzerinde bu alan, uygulamanın gerçek kullanıcı kimliğini belirtir. Windows üzerinde bu alan, uygulamanın kullanıcı kimliğini belirtir.

EffectiveUserKimliği

Tip: MQCHAR12 -giriş

Etkili kullanıcı kimliği. AIX and Linux üzerinde bu alan, uygulamanın geçerli kullanıcı kimliğini belirtir. Windows üzerinde bu alan boştur.

Çevre

Tip: MQLONG-input

Çevre. Bu alan, aramanın yapıldığı ortamı belirtir. Alan aşağıdaki değerlerden biridir:

MQXE_COMMAND_SERVER

Komut sunucusu

MQXE_MQSC

runmqsc komut yorumlayıcısı

MQXE_MCA

İleti kanalı aracısı MQXE_DİĞER

MQXE_DİĞER

Tanımlanmamış ortam

CallerType

Tip: MQLONG-input

Arayan Tipi. Bu alan, çağrıyı yapan programın tipini belirler. Alan aşağıdaki değerlerden biridir:

MQXACT_EXTERNAL

Çağrı, kuyruk yöneticisinin dışında.

MQXACT_INTERNAL

Çağrı, kuyruk yöneticisinin içidir.

AuthenticationType

Tip: MQLONG-input

Kimlik Doğrulama Tipi. Bu alan, gerçekleştirilmekte olan kimlik doğrulamasının tipini belirtir. Alan aşağıdaki değerlerden biridir:

MQZAT_INITIAL_CONTEXT

Kimlik doğrulama çağrısı, kullanıcı bağlamının kullanıma hazırlanmasından kaynaklanıyor. Bu değer bir MQCONN ya da MQCONNX çağrısı sırasında kullanılır.

MQZAT_CHANGE_CONTEXT

Kimlik doğrulama çağrısı, kullanıcı bağlamının değiştirilmesinden kaynaklanıyor. Bu değer, MCA kullanıcı bağlamını değiştirdiğinde kullanılır. Üst konu: MQZAC-

BindType

Tip: MQLONG-input

Bağ Tanımı Tipi. Bu alan, kullanılmakta olan bağ tanımının tipini belirtir. Alan aşağıdaki değerlerden biridir:

MQCNO_FASTPATH_BINDING

Hızlı yol bağlama.

MQCNO_SHARED_BINDING

Paylaşılan bağ tanımı.

MQCNO_ISOLATED_BINDING

Yalıtılmış bağ.

C bildirimini

Yapının alanlarını aşağıdaki gibi bildirin:

```
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 */
};
```

MQZAD-Yetki verileri

MQZAD yapısı, biri giriş, biri çıkış olmak üzere iki deęiřtirge için MQZ_ENUMERATE_AUTHORITY_DATA çağrısında kullanılır.

Filter ve **AuthorityBuffer** parametreleriyle ilgili daha fazla bilgi için bkz.

“MQZ_ENUMERATE_AUTHORITY_DATA-Yetki verilerini numaralandırma” sayfa 1610 :

- MQZAD, çağrıya giriş olan **Filter** deęiřtirgesi için kullanılır. Bu parametre, çağrı tarafından döndürülen yetki verilerini seçmek için kullanılacak seçim ölçütlerini belirtir.
- MQZAD, çağrıdan çıkış olan **AuthorityBuffer** deęiřtirgesi için de kullanılır. Bu parametre, profil adı, nesne tipi ve varlığın bir birleşimine ilişkin yetkileri belirtir.

Tablo 1. yapıdaki alanları özetler.

Çizelge 839. MQZAD içindeki alanlar	
Alan	Açıklama
<u>StrucId</u>	Yapı tanıtıcısı
<u>Sürüm</u>	Yapı sürümü numarası
<u>ProfileName</u>	Tanıtım adı
<u>ObjectType</u>	Nesne tipi
<u>Authority</u>	Authority
<u>EntityDataPtr</u>	Varlık verilerini gösteren işaretçi
<u>EntityType</u>	Varlık tipi
<u>Seçenekler</u>	Seçenekler

Alanlar

StrucId

Tip: MQCHAR4 -giriş

Yapı tanıtıcısı. Deęer ařaęıdaki gibidir:

MQZAD_STRUC_ID

Yetki veri yapısına ilişkin tanıtıcı.

C programlama dili için, MQZAD_STRUC_ID_ARRAY deęiřmezi de tanımlanır; bu deęer MQZAD_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Sürüm

Tip: MQLONG-input

Yapı sürümü numarası. Deęer ařaęıdaki gibidir:

MQZAD_VERSION_1

Version-1 uygulama baęlamı yapısı. MQZAD_CURRENT_VERSION deęiřmezi, yürürlükteki sürümün sürüm numarasını belirtir.

Ařaęıdaki deęiřmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZAD_CURRENT_VERSION

Yetki veri yapısının yürürlükteki sürümü.

ProfileName

Tip: MQCHAR48 -giriş

Profil adı.

Filter parametresi için bu alan, yetki verilerinin gerekli olduğu profil adıdır. Ad, alanın sonuna ya da ilk boş karakterine kadar tamamen boşsa, tüm tanımlı adlarına ilişkin yetki verileri döndürülür.

AuthorityBuffer parametresi için bu alan, belirtilen seçim ölçütüyle eşleşen bir profilin adıdır.

ObjectType

Tip: MQLONG-input

Nesne tipi.

Filter parametresi için bu alan, yetki verilerinin gerekli olduğu nesne türüdür. Değer MQOT_ALL ise, tüm nesne tipleri için yetki verileri döndürülür.

AuthorityBuffer değiştirgesi için bu alan, **ProfileName** değiştirgesiyle tanımlı tanımın geçerli olduğu nesne tipidir.

Değer aşağıdakilerden biridir; **Filter** değiştirgesi için MQOT_ALL değeri de geçerlidir:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri

MQOT_KANAL

Kanal

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı

MQOT_LISTENER

Dinleyici

MQOT_NAMELIST

Ad listesi

MQOT_PROCESS

Süreç tanımlaması

MQOT_Q

Kuyruk

MQOT_Q_MGR

Kuyruk yöneticisi

MQOT_SERVICE

Hizmet

Authority

Tip: MQLONG-input

Otorite.

Filter parametresi için bu alan yoksayılr.

AuthorityBuffer parametresi için bu alan, varlığın **ProfileName** ve **ObjectType** ile tanımlanan nesnelere için sahip olduğu yetkileri temsil eder. Varlığın tek bir yetkisi varsa, alan uygun yetki değerine (MQZAO_* değişmezi) eşittir. Varlığın birden çok yetkisi varsa, alan ilgili MQZAO_* sabitlerinin bit bakımından OR işlecidir.

EntityDataPtr

Tip: PMQZED-giriş

Bir varlığı tanımlayan MQZED yapısının adresi.

Filter değiştirgesi için bu alan, yetki verilerinin gerekli olduğu varlığı tanımlayan bir MQZED yapısını gösterir. **EntityDataPtr** boş değerli işaretçiye, tüm varlıklar için yetki verileri döndürülür.

AuthorityBuffer değiştirgesi için bu alan, yetki verilerinin döndürüldüğü varlığı tanımlayan bir MQZED yapısını gösterir.

EntityType

Tip: MQLONG-input

Varlık tipi.

Filter parametresi için bu alan, yetki verilerinin gerekli olduğu varlık tipini belirtir. Değer MQZAET_NONE ise, tüm varlık tipleri için yetki verileri döndürülür.

AuthorityBuffer değiştirgesi için bu alan, **EntityDataPtr** değiştirgesinin gösterdiği MQZED yapısıyla tanıtilan varlığın tipini belirtir.

Değer aşağıdakilerden biridir; **Filter** parametresi için MQZAET_NONE değeri de geçerlidir:

MQZAET_BIRINCIL KULLANICISI

Müdür

MQZAET_GRUP

Grup

Seçenekler

Tip: MQAUTHOPT-giriş

Seçenekler. Bu alan, görüntülenen tanıtlar üzerinde denetim sağlayan seçenekleri belirler. Aşağıdaki değerlerden biri belirtilmelidir:

MQAUTHOPT_NAME_ALL_EŞLEŞME

Tüm tanıtları görüntüler

MQAUTHOPT_NAME_EXPLICIT

ProfileName alanında belirtilenle tam olarak aynı ada sahip tanıtları görüntüler.

Ayrıca, aşağıdakilerden biri de belirtilmelidir:

MQAUTHOPT_ENTITY_SET

Varlığın **ProfileName** parametresiyle belirtilen nesne üzerinde sahip olduğu birikmeli yetkiyi hesaplamak için kullanılan tüm profilleri görüntüleyin. **ProfileName** parametresi herhangi bir genel arama karakteri içermemelidir.

- Belirtilen varlık bir birincil kullanıcıysa, kümenin her üyesi için {entity, groups} nesne için geçerli olan en uygulanabilir profil görüntülenir.
- Belirtilen varlık bir grupsa, nesneye uygulanan gruptaki en uygulanabilir profil görüntülenir.
- Bu değer belirtilirse, **ProfileName**, **ObjectType**, **EntityType** değerlerinin ve **EntityDataPtr** MQZED yapısında belirtilen varlık adının tümü boş olmamalıdır.

MQAUTHOPT_NAME_ALL_EŞLEŞME belirttiyseniz, aşağıdaki değeri de belirtebilirsiniz:

MQAUTHOPT_ENTITY_EXPLICIT

EntityDataPtr MQZED yapısında belirtilen varlık adıyla tam olarak aynı varlık adına sahip profilleri görüntüler.

C bildiri

```
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-Varlık tanımlayıcı

MQZED yapısı, yetkilendirmenin denetleneceği varlığı belirtmek için bir dizi yetkilendirme hizmeti çağrısında kullanılır.

Tablo 1. yapıdaki alanları özetler.

Çizelge 840. MQZED 'deki Alanlar	
Alan	Açıklama
<u>StrucId</u>	Yapı tanıttıcısı
<u>Sürüm</u>	Sürüm
<u>EntityName Ptr</u>	Varlık Adı
<u>EntityDomainPtr</u>	Varlık etki alanı işaretçisi
<u>SecurityId</u>	Güvenlik tanıttıcısı
<u>CorrelationPtr</u>	İlinti göstergesi

Alanlar

StrucId

Tip: MQCHAR4 -giriş

Yapı tanıttıcısı. Değer aşağıdaki gibidir:

MQZED_STRUC_ID

Varlık tanımlayıcı yapısına ilişkin tanıttıcı.

C programlama dili için, MQZED_STRUC_ID_ARRAY değişmezi de tanımlanır; bu değer MQZED_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Sürüm

Tip: MQLONG-input

Yapı sürümü numarası. Değer aşağıdaki gibidir:

MQZED_VERSION_1

Version-1 varlık tanımlayıcı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZED_CURRENT_VERSION

Varlık tanımlayıcı yapısının yürürlükteki sürümü.

EntityNamePtr

Tip: PMQCHAR-giriş

Profil adı.

Varlık adının adresi. Bu, yetkilendirmesi denetlenecek varlığın adını gösteren bir işaretçidir.

EntityDomainPtr

Tip: PMQCHAR-giriş

Varlık etki alanı adının adresi. Bu, yetkisi denetlenecek varlığın tanımını içeren etki alanının adını gösteren bir işaretçidir.

SecurityId

Tip: MQBYTE40 -giriş

Otorite.

Güvenlik tanıttıcısı. Bu, yetkisi denetlenecek olan güvenlik tanıttıcısıdır.

CorrelationPtr

Tip: MQPTR-giriş

İlinti göstergesi. Bu, kimlik doğrulama kullanıcı işlevi ile diğer uygun OAM işlevleri arasında korelasyon verilerinin aktarılmasını kolaylaştırır.

C bildirim

```
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-Bileşen giriş noktası ekle

Bir hizmet bileşeni, başlatma sırasında, o hizmet bileşenine ilişkin giriş noktası vektörüne bir giriş noktası eklemek için bu işlevi başlatır.

Sözdizimi

MQZEP (*Hconfig* , *İşlev* , *EntryPoint* , *CompCode* , *Neden*)

Parametreler

Hconfig (Yapılandırma)

Tip: MQHCONFIG-giriş

Yapılandırma tanıttıcısı. Bu tanıttıcı, bu kurulabilir hizmet için yapılandırılmakta olan bileşeni gösterir. Bileşen kullanıma hazırlama çağrısında kuyruk yöneticisi tarafından bileşen yapılandırma işlevine geçirilen bileşenle aynı olmalıdır.

İşlev

Tip: MQLONG-input

İşlev tanıttıcısı. Bu hizmet için geçerli değerler, her kurulabilir hizmet için tanımlanır.

MQZEP aynı işlev için bir kereden fazla çağrılırsa, yapılan son çağrı kullanılan giriş noktasını sağlar.

EntryPoint

Tip: PMQFUNC-giriş

İşlev giriş noktası. Bu, işlevi gerçekleştirmek için bileşen tarafından sağlanan giriş noktasının adresidir.

NULL değeri geçerli ve işlevin bu bileşen tarafından sağlanmadığını gösteriyor. MQZEP kullanılarak tanımlanmamış giriş noktaları için NULL kabul edilir.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_FUNCTION_HATA

(2281, X'8E9') İşlev tanıtıcısı geçersiz.

MQRC_HCONFIG_HATA

(2280, X'8E8') Yapılandırma tanıtıcısı geçersiz.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZEP (Hconfig, Function, EntryPoint, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
MQHCONFIG Hconfig; /* Configuration handle */
MQLONG Function; /* Function identifier */
PMQFUNC EntryPoint; /* Function entry point */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Reason code qualifying CompCode */
```

MQZFP-Serbest değiştirgeler

MQZFP yapısı, *FreeParms* parametresi için MQZ_FREE_USER çağırısında kullanılır. Bu değiştirge, serbest bırakılacak kaynakla ilgili verileri belirtir.

Tablo 1. yapıdaki alanları özetler.

Çizelge 841. MQZFP 'deki Alanlar	
Alan	Açıklama
<u>StrucId</u>	Yapı tanıtıcısı
<u>Sürüm</u>	Sürüm
<u>Ayrıldı</u>	Ayrılmış alan
<u>CorrelationPtr</u>	İlinti göstergesi

Alanlar

StrucId

Tip: MQCHAR4 -giriş

Yapı tanıtıcısı. Değer aşağıdaki gibidir:

MQZIC_STRUC_ID

Kimlik bağlamı yapısına ilişkin tanıtıcı. C programlama dili için, MQZIC_STRUC_ID_ARRAY değişmezi de tanımlanır; bu değer MQZIC_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Sürüm

Tip: MQLONG-input

Yapı sürümü numarası. Değer aşağıdaki gibidir:

MQZFP_VERSION_1

Version-1 serbest değiştirge yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZFP_CURRENT_VERSION

Serbest parametre yapısının geçerli sürümü.

Ayrıldı

Tip: MQBYTE8 -giriş

Ayrılmış alan. İlk değer boş.

CorrelationPtr

Tip: MQPTR-giriş

İlinti göstergesi. Serbest bırakılacak kaynakla ilgili ilinti verilerinin adresi.

C bildirimi

```
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-Kimlik bağlamı

MQZIC yapısı, *IdentityContext* değiştirgesi için MQZ_AUTHENTICATE_USER çağrısında kullanılır.

MQZIC yapısı, iletiyi kuyruğa ilk koyan uygulamanın kullanıcıyı tanıtan kimlik bağlamı bilgilerini içerir:

- Kuyruk yöneticisi, *UserIdentifier* alanını kullanıcıyı tanıtan bir adla doldurur; kuyruk yöneticisinin bunu yapabilme şekli, uygulamanın çalıştığı ortama bağlıdır.
- Kuyruk yöneticisi, *AccountingToken* alanını, iletiyi koyan uygulamadan belirlediği bir simge ya da numarayla doldurur.
- Uygulamalar, kullanıcı hakkında (örneğin, şifrelenmiş bir parola) eklemek istedikleri ek bilgiler için *AppIdentityData* alanını kullanabilir.

Uygun yetkili uygulamalar, MQZ_AUTHENTICATE_USER işlevini kullanarak kimlik bağlamını ayarlayabilir.

IBM MQ for Windowsaltında bir ileti oluşturulduğunda, *AccountingToken* alanında bir Windows sistem güvenlik tanıtıcısı (SID) saklanır. SID, *UserIdentifier* alanını tamamlamak ve bir kullanıcının kimlik bilgilerini oluşturmak için kullanılabilir.

Tablo 1. yapıdaki alanları özetler.

Çizelge 842. MQZIC 'deki Alanlar	
Alan	Açıklama
<u>StrucId</u>	Yapı tanıtıcısı
<u>Sürüm</u>	Sürüm
<u>UserIdentifier</u>	Kullanıcı kimliği
<u>AccountingToken</u>	Muhasebe simgesi
<u>AppIdentityVerileri</u>	Uygulama kimlik verileri

Alanlar

StrucId

Tip: MQCHAR4 -giriş

Yapı tanıtıcısı. Değer aşağıdaki gibidir:

MQZIC_STRUC_ID

Kimlik bağlamı yapısına ilişkin tanıtıcı. C programlama dili için, MQZIC_STRUC_ID_ARRAY değişmezi de tanımlanır; bu değer MQZIC_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Sürüm

Tip: MQLONG-input

Yapı sürümü numarası. Değer aşağıdaki gibidir:

MQZIC_VERSION_1

Version-1 kimlik bağlamı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZIC_CURRENT_VERSION

Kimlik bağlamı yapısının yürürlükteki sürümü.

UserIdentifier

Tip: MQCHAR12 -giriş

Kullanıcı kimliği. Bu, iletinin kimlik bağlamının bir parçasıdır. *UserIdentifier*, iletiyi oluşturan uygulamanın kullanıcı kimliğini belirtir. Kuyruk yöneticisi bu bilgileri karakter verileri olarak kabul eder, ancak biçimini tanımlamaz. *UserIdentifier* alanıyla ilgili daha fazla bilgi için bkz. [“MQMD için UserIdentifier \(MQCHAR12\)” sayfa 455.](#)

AccountingToken

Tip: MQBYTE32 -giriş

Hesap simgesi. Bu, iletinin kimlik bağlamının bir parçasıdır. *AccountingToken*, bir uygulamanın, iletinin sonucu olarak yapılan işlerin uygun şekilde ücretlendirilmesine neden olmasını sağlar. Kuyruk yöneticisi bu bilgileri bir bit dizgesi olarak işler ve içeriğini denetlemez. *AccountingToken* alanıyla ilgili daha fazla bilgi için bkz. [“MQMD için AccountingToken \(MQBYTE32\)” sayfa 456.](#)

ApplIdentityVerileri

Tip: MQCHAR32 -giriş

Kimlikle ilgili uygulama verileri. Bu, iletinin kimlik bağlamının bir parçasıdır. ApplIdentityVerileri, iletinin kökenine ilişkin ek bilgi sağlamak için uygulama grubu tarafından tanımlanan bilgilerdir. Örneğin, kimlik verilerinin güvenilir olup olmadığını belirtmek için uygun kullanıcı yetkisiyle çalışan uygulamalar tarafından ayarlanabilir. ApplIdentityData alanı hakkında daha fazla bilgi için bkz. [“MQMD için ApplIdentityVerileri \(MQCHAR32\)” sayfa 458.](#)

C bildiri

```
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 */
};
```

IBM i

IBM i üzerinde kurulabilir hizmetler arabirimi başvuru bilgileri

IBM için kurulabilir hizmetlere ilişkin başvuru bilgilerini anlamak için bu bilgileri kullanın.

Her işlev için, işlev tanıtıcısını (MQZEP için) içeren bir tanım vardır.

Değiştirgeler, gerçekleştirmeleri gereken sırayla gösterilir. Hepsi orada olmalı.

Her parametre adını parantez içinde veri tipi izler. Bunlar, [“Temel veri tipleri” sayfa 982](#) içinde açıklanan temel veri tipleridir.

Parametrelerin açıklamasından sonra C dili çağırısı da verilir.

İlgili kavramlar

IBM i IBM i için kurulabilir hizmetler ve bileşenler

ALW UNIX, Linux ve Windows için kurulabilir hizmetler ve bileşenler

İlgili başvurular

“Kurulabilir hizmetler arabirimi başvuru bilgileri” sayfa 1593

Bu konu derlemi, kurulabilir hizmetlere ilişkin başvuru bilgilerini içerir.

IBM i IBM i üzerinde MQZEP (Bileşen giriş noktası ekle)

Bu işlev, başlatma sırasında bir hizmet bileşeni tarafından, o hizmet bileşenine ilişkin giriş noktası vektörüne bir giriş noktası eklemek için çağrılır.

Sözdizimi

```
MQZEP (Hconfig, Function, EntryPoint, CompCode, Reason)
```

Parametreler

MQZEP çağrısı aşağıdaki parametrelere sahiptir.

Hconfig (MQHCONFIG)-giriş

Yapılandırma tanıtıcısı.

Bu tanıtıcı, bu kurulabilir hizmet için yapılandırılmakta olan bileşeni gösterir. Bileşen kullanıma hazırlama çağrısında kuyruk yöneticisi tarafından bileşen yapılandırma işlevine geçirilen işlevle aynı olmalıdır.

İşlev (MQLONG)-giriş

İşlev tanıtıcısı.

Bu hizmet için geçerli değerler, her kurulabilir hizmet için tanımlanır. MQZEP aynı işlev için bir kereden fazla çağrılırsa, yapılan son çağrı kullanılan giriş noktasını sağlar.

EntryPoint (PMQFUNC)-giriş

İşlev giriş noktası.

Bu, işlevi gerçekleştirmek için bileşen tarafından sağlanan giriş noktasının adresidir. NULL değeri geçerli ve işlevin bu bileşen tarafından sağlanmadığını gösteriyor. MQZEP kullanılarak tanımlanmamış giriş noktaları için NULL varsayılır.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_FUNCTION_HATA

(2281, X'8E9') İşlev tanıtıcısı geçersiz.

MQRC_HCONFIG_HATA

(2280, X'8E8') Yapılandırma tanıtıcısı geçersiz.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZEP (Hconfig, Function, EntryPoint, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
MQHCONFIG Hconfig; /* Configuration handle */
MQLONG Function; /* Function identifier */
PMQFUNC EntryPoint; /* Function entry point */
MQLONG CompCode; /* Completion code */
MQLONG Reason; /* Reason code qualifying CompCode */
```

IBM i MQHCONFIG (Yapılandırma tanıtıcısı)- IBM i

MQHCONFIG veri tipi, belirli bir kurulabilir hizmet için yapılandırılmakta olan bir yapılandırma tanıtıcısını gösterir. Bir yapılandırma tanıtıcısı, doğal sınırında hizalanmalıdır.

Uygulamalar, yalnızca eşitlik için bu tipteki değişkenleri test etmek zorundadır.

C bildirimi

```
typedef void MQPOINTER MQHCONFIG;
```

IBM i IBM i üzerinde PMQFUNC (İşaretçi işlevi)

Bir işleve ilişkin gösterge.

C bildirimi

```
typedef void MQPOINTER PMQFUNC;
```

IBM i MQZ_AUTHENTICATE_USER (Kullanıcının kimliğini doğrula)- IBM i

Bu işlev bir MQZAS_VERSION_5 yetkilendirme hizmeti bileşeni tarafından sağlanır. Bir kullanıcının kimliğini doğrulamak ya da kimlik bağlamı alanlarını ayarlamak için kuyruk yöneticisi tarafından çağrılır.

Bir IBM MQ kullanıcı uygulaması bağlamı oluşturulduğunda çağrılır. Bu, uygulamanın kullanıcı bağlamının başlatıldığı noktada ve uygulamanın kullanıcı bağlamının değiştirildiği her noktada bağlanma çağrılarının sırasında gerçekleşir. Her bağlanma çağrısı yapıldığında, *IdentityContext* alanında uygulamanın kullanıcı bağlamı bilgileri yeniden alınır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_AUTHENTICATE_USER.

Sözdizimi

MQZ_AUTHENTICATE_USER (QMgrName, SecurityParms, ApplicationContext, IdentityContext, CorrelationPtr, ComponentData, Continuation, CompCode, Reason)

Parametreler

MQZ_AUTHENTICATE_USER çağırısı aşağıdaki parametreleri içeriyor.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz. Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

SecurityParms (MQCSP)-giriş

Güvenlik parametreleri.

Kullanıcı kimliği, parola ve kimlik doğrulama tipiyle ilgili veriler.

MQCONN MQI çağrısı sırasında bu değiştirge boş değer ya da varsayılan değerler içeriyor.

ApplicationContext (MQZAC)-giriş

Uygulama bağlamı.

Çağıran uygulamayla ilgili veriler. Ayrıntılar için bkz. [“IBM i üzerinde MQZAC \(Uygulama bağlamı\)” sayfa 1687](#). Her MQCONN ya da MQCONNX MQI çağrısı sırasında, MQZAC yapısındaki kullanıcı bağlamı bilgileri yeniden alınır.

IdentityContext (MQZIC)-giriş/çıkış

Kimlik bağlamı.

Kimlik doğrulama kullanıcı işlevine girişte bu, geçerli kimlik bağlamını tanımlar. Kullanıcı kimliğini doğru işlevi bunu değiştirebilir; bu noktada kuyruk yöneticisi yeni kimlik bağlamını benimser. MQZIC yapısıyla ilgili daha fazla ayrıntı için bkz. [“IBM i üzerinde MQZIC \(Kimlik bağlamı\)” sayfa 1693](#).

CorrelationPtr (MQPTR)-çıkış

İlinti göstergesi.

İlinti verilerinin adresini belirler. Bu işaretçi daha sonra diğer OAM çağrılarına aktarılır.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenlerden birinin bir sonraki çağrılışında sunulur. Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Devam işareti.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Diğer bileşenlere bağlı olarak devam.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZ_AUTHENTICATE_USER (QMgrName, SecurityParms, ApplicationContext,  
IdentityContext, &CorrelationPtr, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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

IBM i üzerinde MQZ_CHECK_AUTHORITY (Yetki denetimi)

Bu işlev bir MQZAS_VERSION_1 yetkilendirme hizmeti bileşeni tarafından sağlanır ve bir varlığın belirli bir nesne üzerinde belirli bir işlem ya da işlem gerçekleştirme yetkisi olup olmadığını denetlemek için kuyruk yöneticisi tarafından çağırılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_CHECK_AUTHORITY.

Sözdizimi

MQZ_CHECK_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName, ObjectType, Authority, ComponentData, Continuation, CompCode, Reason)

Parametreler

MQZ_CHECK_AUTHORITY çağırısı aşağıdaki parametreleri içeriyor.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz. Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

EntityName (MQCHAR12)-giriş

Varlık adı.

Nesne için yetkisi denetlenecek varlığın adı. Dizginin uzunluk üst sınırı 12 karakterdir; uzunluktan daha kısaysa, sağ tarafta boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

Bu varlığın temel güvenlik hizmeti tarafından bilinmesi gerekli değildir. Bilinmezse, denetim için özel **hiç kimse** grubunun yetkileri (tüm varlıkların ait olduğu varsayılır) kullanılır. Tümü boş bir ad geçerlidir ve bu şekilde kullanılabilir.

EntityType (MQLONG)-giriş

Varlık tipi.

EntityName tarafından belirtilen varlık tipi. Aşağıdakilerden biridir:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRUP

Grup.

ObjectName (MQCHAR48)-giriş

Nesne adı.

Erişim gerektiren nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType (MQLONG)-giriş

Nesne tipi.

ObjectName tarafından belirtilen varlık tipi. Aşağıdakilerden biridir:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

Yetki (MQLONG)-giriş

Denetlenecek yetki.

Bir yetki denetleniyorsa, bu alan uygun yetkilendirme işlemine (MQZAO_* değişmezi) eşittir. Birden çok yetki denetleniyorsa, bu, ilgili MQZAO_* değişmezlerinin bit bakımından OR işlecidir.

MQI çağrılarının kullanımı için aşağıdaki yetkiler geçerlidir:

MQZAO_CONNECT

MQCONN çağrısı kullanma yeteneği.

MQZAO_GÖZ AT

MQGET çağrısı bir göz atma seçeneğiyle kullanma yeteneği.

Bu, MQGET çağrısında MQGMO_BROWSE_FIRST, MQGMO_BROWSE_MSG_UNDER_CURSOR ya da MQGMO_BROWSE_NEXT seçeneğinin belirtilmesini sağlar.

MQZAO_INPUT

MQGET çağrısı bir giriş seçeneğiyle kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_INPUT_SHARED, MQOO_INPUT_EXCLUSIVE ya da MQOO_INPUT_AS_Q_DEF seçeneğinin belirtilmesini sağlar.

MQZAO_OUTPUT

MQPUT çağrısı kullanma yeteneği.

MQOPEN çağrısında MQOO_OUTPUT seçeneğinin belirtilmesini sağlar.

MQZAO_INQUIRE

MQINQ çağrısı kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_INQUIRE seçeneğinin belirtilmesini sağlar.

MQZAO_KÜMESİ

MQSET çağrısı kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_SET seçeneğinin belirtilmesini sağlar.

MQZAO_PASS_IDENTITY_CONTEXT

Kimlik bağlamını geçme yeteneği.

Bu, MQOPEN çağrısında MQOO_PASS_IDENTITY_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_PASS_IDENTITY_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_PASS_ALL_CONTEXT

Tüm bağlamı geçme becerisi.

Bu, MQOPEN çağrısında MQOO_PASS_ALL_CONTEXT seçeneğinin belirtilmesini ve MQPUT ve MQPUT1 çağrılarında MQPMO_PASS_ALL_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_SET_IDENTITY_CONTEXT

Kimlik bağlamını ayarlama yeteneği.

Bu, MQOPEN çağrısında MQOO_SET_IDENTITY_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_SET_IDENTITY_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_SET_ALL_CONTEXT

Tüm bağlamı ayarlama yeteneği.

Bu, MQOPEN çağrısında MQOO_SET_ALL_CONTEXT seçeneğinin ve MQPUT ve MQPUT1 çağrılarında MQPMO_SET_ALL_CONTEXT seçeneğinin belirtilmesini sağlar.

MQZAO_ALTERNATE_USER_AUTHORITY

Diğer kullanıcı yetkisini kullanma yeteneği.

Bu, MQOPEN çağrısında MQOO_ALTERNATE_USER_AUTHORITY seçeneğinin ve MQPUT1 çağrısında MQPMO_ALTERNATE_USER_AUTHORITY seçeneğinin belirtilmesini sağlar.

MQZAO_ALL_MQI

Tüm MQI yetkileri.

Bu, daha önce açıklanan tüm yetkileri etkinleştirir.

Bir kuyruk yöneticisinin yönetimi için aşağıdaki yetkiler geçerlidir:

MQZAO_OLUŞTURMA

Belirtilen tipte nesnelere yaratma yeteneği.

MQZAO_DELETE

Belirtilen bir nesneyi silme yeteneği.

MQZAO_DISPLAY

Belirtilen bir nesnenin özniteliklerini görüntüleme yeteneği.

MQZAO_CHANGE

Belirtilen bir nesnenin özniteliklerini değiştirme yeteneği.

MQZAO_CLEAR

Belirtilen bir kuyruktan tüm iletileri silme yeteneği.

MQZAO_AUTHORIZE

Belirli bir nesne için diğer kullanıcılara yetki verme yeteneği.

MQZAO_CONTROL

İstemci olmayan bir kanal nesnesini başlatma, durdurma ya da ping işlemi gerçekleştirme yeteneği.

MQZAO_CONTROL_EXTENDED

Bir sıra numarasını sıfırlama ya da istemci olmayan bir kanal nesnesindeki belirsiz bir iletiyi çözme yeteneği.

MQZAO_ALL_ADMIN

MQZAO_CREATE dışındaki tüm denetim yetkileri.

Aşağıdaki yetkiler hem MQI kullanımı hem de bir kuyruk yöneticisinin yönetimi için geçerlidir:

MQZAO_Tümü

MQZAO_CREATE dışındaki tüm yetkiler.

MQZAO_NONE

Yetki yok.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Bileşen tarafından ayarlanan devam göstergesi.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZ_CHECK_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
Objectype, Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
MQCHAR48 QMgrName;          /* Queue manager name */  
MQCHAR12 EntityName;       /* Entity name */  
MQLONG   EntityType;       /* Entity type */  
MQCHAR48 ObjectName;       /* Object name */  
MQLONG   Objectype;        /* 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-Kullanıcının ayrıcalıklı olup olmadığını denetleyin

Bu işlev bir MQZAS_VERSION_6 yetkilendirme hizmeti bileşeni tarafından sağlanır ve belirtilen kullanıcının ayrıcalıklı bir kullanıcı olup olmadığını saptamak için kuyruk yöneticisi tarafından çağırılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_CHECK_PRIVILEGED.

Sözdizimi

```
MQZ_CHECK_PRIVILEGED( QMgrName , EntityData , EntityType , ComponentData ,  
Continuation , CompCode , Reason )
```

Parametreler

QMgrName

Tip: MQCHAR48 -giriş

Kuyruk yöneticisi adı. Bileşeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

EntityData

Tip: MQZED-giriş

Varlık verileri. Denetlenecek varlıkla ilgili veriler. Daha fazla bilgi için bkz. [“MQZED-Varlık tanımlayıcı” sayfa 1651.](#)

EntityType

Tip: MQLONG-input

Varlık tipi. EntityDatatarafından belirtilen varlık tipi. Aşağıdaki değerlerden biri olmalıdır:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRPUP

Grup.

ComponentData

Tip: MQBYTEExComponentDataLength -giriş/çıkış

Bileşen verileri. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlemlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşen işlemlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam

Tip: MQLONG-output

Bileşen tarafından ayarlanan devam göstergesi. Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_CHECK_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

Bir bileşene çağrı başarısız olursa (*CompCode* , MQCC_FAILED değerini döndürür) ve *Devam* parametresi MQZCI_DEFAULT ya da MQZCI_CONTINUE ise, kuyruk yöneticisi varsa diğer bileşenleri çağrıya devam eder.

Çağrı başarılı olursa (*CompCode* , MQCC_OK değerini döndürür), *Devam* ayarı ne olursa olsun başka hiçbir bileşen çağrılmaz.

Çağrı başarısız olursa ve *Devam* parametresi MQZCI_STOP ise, başka hiçbir bileşen çağrılmaz ve hata kuyruk yöneticisine döndürülür. Bileşenlerin önceki çağrılar hakkında bilgisi olmadığı için, *Devam* parametresi her zaman çağrıdan önce MQZCI_DEFAULT olarak ayarlanır.

CompCode

Tip: MQLONG-output

Tamamlanma kodu. Aşağıdaki değerlerden biri olmalıdır:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden

Tip: MQLONG-output

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_PRIVILEGED

(2584, X'A18') Bu kullanıcı ayrıcalıklı bir kullanıcı kimliği değil.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarıyla ilgili daha fazla bilgi için bkz. [API tamamlama ve neden kodları](#).

C çağırısı

```
MQZ_CHECK_PRIVILEGED (QMgrName, &EntityData, EntityType,  
                      ComponentData, &Continuation,  
                      &CompCode, &Reason);
```

Hizmete geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
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 */
```

IBM i

MQZ_COPY_ALL_AUTHORITY (IBM i üzerinde tüm yetkiyi kopyala)

Bu işlev bir yetkilendirme hizmeti bileşeni tarafından sağlanır. Kuyruk yöneticisi tarafından, bir başvuru nesnesi için yürürlükte olan tüm yetkileri başka bir nesneye kopyalamak için çağırılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_COPY_ALL_AUTHORITY 'dir.

Sözdizimi

MQZ_COPY_ALL_AUTHORITY (QMgrName, RefObjectName, ObjectName, ObjectName, ObjectType, ComponentData, Continuation, CompCode, Reason)

Parametreler

MQZ_COPY_ALL_AUTHORITY çağırısı aşağıdaki parametreleri içeriyor.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

RefObjectAdı (MQCHAR48)-giriş

Başvuru nesnesi adı.

Referans nesnesinin adı, kopyalanacak yetkiler. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmaz.

ObjectName (MQCHAR48)-giriş

Nesne adı.

Erişimleri ayarlanacak nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmaz.

ObjectType (MQLONG)-giriş

Nesne tipi.

RefObjectName ve *ObjectName* tarafından belirtilen nesne tipi. Aşağıdakilerden biridir:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Bileşen tarafından ayarlanan devam göstergesi.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_COPY_ALL_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmette erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

MQRC_UNKNOWN_REF_OBJECT

(2294, X'8F6') Başvuru nesnesi bilinmiyor.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZ_COPY_ALL_AUTHORITY (QMgrName, RefObjectName, ObjectName, ObjectType,  
ComponentData, &Continuation, &CompCode,  
&Reason);
```

Hizmete geçirilen değişirgeler aşağıdaki gibi bildirilir:

```
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 */
```

IBM i

IBM i üzerinde MQZ_DELETE_AUTHORITY (Silme yetkisi)

Bu işlev bir yetkilendirme hizmeti bileşeni tarafından sağlanır ve belirtilen nesneyle ilişkili tüm yetkileri silmek için kuyruk yöneticisi tarafından çağırılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_DELETE_AUTHORITY.

Sözdizimi

**MQZ_DELETE_AUTHORITY (QMgrName, ObjectName, ObjectType,
ComponentData, Continuation, CompCode, Reason)**

Parametreler

MQZ_DELETE_AUTHORITY çağırısı aşağıdaki parametreleri içeriyor.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

ObjectName (MQCHAR48)-giriş

Nesne adı.

Erişimleri silinecek nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmaz.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType (MQLONG)-giriş

Nesne tipi.

ObjectName tarafından belirtilen varlık tipi. Aşağıdakilerden biridir:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Bileşen tarafından ayarlanan devam göstergesi.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_DELETE_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZ_DELETE_AUTHORITY (QMgrName, ObjectName, ObjectType, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen değıştirmeler ařağıdaki gibi bildirilir:

```
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 */
```

IBM i

IBM i üzerinde MQZ_ENUMERATE_AUTHORITY_DATA (Yetki verilerini numaralandır)

Bu işlev bir MQZAS_VERSION_4 yetkilendirme hizmeti bileşeni tarafından sağlanır ve kuyruk yöneticisi tarafından, ilk çağrıda belirtilen seçim ölçütleriyle eşleşen tüm yetki verilerini almak için arka arkaya çağrılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_ENUMERATE_AUTHORITY_DATA.

Sözdizimi

MQZ_ENUMERATE_AUTHORITY_DATA (QMgrName, StartEnumeration, Filter, AuthorityBufferLength, AuthorityBuffer, AuthorityDataLength, ComponentData, Continuation, CompCode, Reason)

Parametreler

MQZ_ENUMERATE_AUTHORITY_DATA çağırısı ařağıdaki parametreleri içeriyor.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağırın kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

StartEnumeration (MQLONG)-giriş

Aramanın numaralandırmayı başlatıp başlatmayacağını gösteren işaret.

Bu, çağırının yetki verilerinin numaralandırmasını başlatıp başlatmayacağını ya da önceki bir MQZ_ENUMERATE_AUTHORITY_DATA çağırısıyla başlatılan yetki verilerinin numaralandırmasına devam edip etmeyeceğini belirtir. Değer ařağıdakilerden biridir:

MQZSE_START

Numaralandırmayı başlat.

Çağrı, yetki verilerinin numaralandırmasını başlatmak için bu değerle çağrılır. **Filter** parametresi, bu ve ardışık çağrılar tarafından döndürülen yetki verilerini seçmek için kullanılacak seçim ölçütlerini belirtir.

MQZSE_CONTINUE

Numaralandırmaya devam et.

Çağrı, yetki verilerinin numaralandırmasına devam etmek için bu değerle çağrılır. Bu durumda **Filter** değiştirgesi yoksayılr ve boş değerli gösterge olarak belirtilebilir (seçim ölçütleri, *StartEnumeration* ' u MQZSE_START olarak ayarlı olan çağrı tarafından belirtilen **Filter** değiştirgesiyle belirlenir).

Süzgeç (MQZAD)-giriş

Filtrele.

StartEnumeration MQZSE_START ise, *Filter* döndürülecek yetki verilerini seçmek için kullanılacak seçim ölçütlerini belirtir. *Filter* boş değerli işaretçiye, seçim ölçütü kullanılmaz; yani, tüm yetki verileri döndürülür. Kullanılabilecek seçim ölçütlerine ilişkin ayrıntılar için bkz. "[IBM i üzerinde MQZAD \(Yetki verileri\)" sayfa 1689](#) .

StartEnumeration MQZSE_CONTINUE ise, *Filter* yoksayılr ve boş değerli gösterge olarak belirtilebilir.

AuthorityBufferUzunluğu (MQLONG)-giriş

*AuthorityBuffer*uzunluğu.

Bu, **AuthorityBuffer** parametresinin bayt cinsinden uzunluğudur. Yetki arabelleği, döndürülecek verileri alabilecek büyüklükte olmalıdır.

AuthorityBuffer (MQZAD)-çıkış

Yetki verileri.

Bu, yetki verilerinin döndürüldüğü arabellektir. Arabellek, bir MQZAD yapısını, bir MQZED yapısını ve en uzun varlık adını ve tanımlanan en uzun etki alanı adını barındıracak kadar büyük olmalıdır.

Not: Bu değiştirge MQZAD olarak tanımlanır; MQZAD arabelleğin başında her zaman oluşur. Ancak, arabellek gerçekte bir MQZAD olarak bildirildiyse, arabellek çok küçük olur; MQZAD, MQZED ve etki alanı adlarını barındırabilmesi için arabellek bir MQZAD ' dan büyük olmalıdır.

AuthorityDataUzunluğu (MQLONG)-çıkış

*AuthorityBuffer*içinde döndürülen verilerin uzunluğu.

Bu, *AuthorityBuffer*içinde döndürülen verilerin uzunluğudur. Yetki arabelleği çok küçükse, *AuthorityDataLength* gereken arabelleğin uzunluğuna ayarlanır ve çağrı tamamlanma kodu MQCC_FAILED ve neden kodu MQRC_BUFFER_LENGTH_ERROR döndürür.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Bileşen tarafından ayarlanan devam göstergesi.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_ENUMERATE_AUTHORITY_DATA için bu, MQZCI_CONTINUE ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_BUFFER_LENGTH_ERROR

(2005, X'7D5') Arabellek uzunluğu parametresi geçersiz.

MQRC_NO_DATA_VAR

(2379, X'94B') Kullanılabilir veri yok.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZ_ENUMERATE_AUTHORITY_DATA (QMgrName, StartEnumeration, &Filter,
                                AuthorityBufferLength,
                                &AuthorityBuffer,
                                &AuthorityDataLength, ComponentData,
                                &Continuation, &CompCode,
                                &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Serbest kullanıcı

Bu iřlev bir MQZAS_VERSION_5 yetkilendirme hizmeti bileřeni tarafından saęlanır ve iliřkili ayrılmıř kaynaęı serbest bırakmak için kuyruk yneticisi tarafından çağırılır. Bir uygulama tm kullanıcı baęlıamları altında çalıřmayı bitirdięinde (örneęin, bir MQDISC MQI çağırısı sırasında) çağırılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_FREE_USER.

IBM i üzerinde MQZ_GET_AUTHORITY (Yetki al)

Bu işlev bir MQZAS_VERSION_1 yetkilendirme hizmeti bileşeni tarafından sağlanır ve bir varlığın belirtilen nesneye erişmek için sahip olduğu yetkiyi almak üzere kuyruk yöneticisi tarafından çağrılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_GET_AUTHORITY.

Sözdizimi

MQZ_GET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName, ObjectType, Authority, ComponentData, Continuation, CompCode, Reason)

Parametreler

MQZ_GET_AUTHORITY çağrısı aşağıdaki parametreleri içeriyor.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

EntityName (MQCHAR12)-giriş

Varlık adı.

Nesneye erişimi alınacak varlığın adı. Dizginin uzunluk üst sınırı 12 karakterdir; uzunluktan daha kısaysa, sağ tarafta boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmaz.

EntityType (MQLONG)-giriş

Varlık tipi.

EntityName tarafından belirtilen varlık tipi. Aşağıdaki değer belirtilebilir:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRUP

Grup.

ObjectName (MQCHAR48)-giriş

Nesne adı.

Varlığın yetkisinin alınacağı nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmaz.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType (MQLONG)-giriş

Nesne tipi.

*ObjectNam*e tarafından belirtilen varlık tipi. Aşağıdakilerden biridir:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

Yetki (MQLONG)-çıkış

Varlığın yetkisi.

Varlığın tek bir yetkisi varsa, bu alan uygun yetkilendirme işlemine (MQZAO_ * değişmezi) eşittir. Birden çok yetkisi varsa, bu alan ilgili MQZAO_ * sabitlerinin bit bakımından OR işlemidir.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değıştirtgesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Bileşen tarafından ayarlanan devam göstergesi.

Aşğıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_GET_AUTHORITY için bu, MQZCI_CONTINUE ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşğıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmette erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZ_GET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
ObjectTyp e, &Authority, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
MQCHAR48 QMgrName; /* Queue manager name */  
MQCHAR12 EntityName; /* Entity name */  
MQLONG EntityType; /* Entity type */  
MQCHAR48 ObjectName; /* Object name */  
MQLONG ObjectTyp e; /* 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

IBM i üzerinde MQZ_GET_AÇIKIT_AUTHORITY (Belirtik yetki al)

Bu işlev bir MQZAS_VERSION_1 yetkilendirme hizmeti bileşeni tarafından sağlanır ve kuyruk yöneticisi, adı belirtilen bir grubun belirtilen bir nesneye (ancak **kimse** grubunun ek yetkisi olmadan) ya da adı belirtilen birincil grubun belirtilen bir nesneye erişmesi gereken yetkiyi almak için kuyruk yöneticisi tarafından çağrılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_GET_BELIRIT_AUTHORITY.

Sözdizimi

MQZ_GET_BELIRIT_YETKISI (*QMgrName, EntityName, EntityType, ObjectName, ObjectTyp e, Authority, ComponentData, Continuation, CompCode, Reason*)

Parametreler

MQZ_GET_BELIRIT_AUTHORITY çağırısı ařaęıdaki deęiřtirgeleri ieriyor.

QMgrName (MQCHAR48)-giriř

Kuyruk yöneticisi adı.

Bileşeni çağır an kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluęuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

EntityName (MQCHAR12)-giriř

Varlık adı.

Nesneye erişimin alınacaęı varlığın adı. Dizginin uzunluk üst sınırı 12 karakterdir; uzunluktan daha kısaysa, saę tarafta boşluklarla doldurulur. Ad boş deęerli bir karakterle sonlandırılmadı.

EntityType (MQLONG)-giriş

Varlık tipi.

EntityName tarafından belirtilen varlık tipi. Aşağıdaki değer belirtilebilir:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRUP

Grup.

ObjectName (MQCHAR48)-giriş

Nesne adı.

Varlığın yetkisinin alınacağı nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType (MQLONG)-giriş

Nesne tipi.

ObjectName tarafından belirtilen varlık tipi. Aşağıdakilerden biridir:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

Yetki (MQLONG)-çıkış

Varlığın yetkisi.

Varlığın tek bir yetkisi varsa, bu alan uygun yetkilendirme işlemine (MQZAO_* değişmezi) eşittir. Birden çok yetkisi varsa, bu alan ilgili MQZAO_* sabitlerinin bit bakımından OR işlecidir.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Bileşen tarafından ayarlanan devam göstergesi.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_GET_BELIRIT_AUTHORITY için bu, MQZCI_CONTINUE ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZ_GET_EXPLICIT_AUTHORITY (QMgrName, EntityName, EntityType,  
                             ObjectName, ObjectType, &Authority,  
                             ComponentData, &Continuation,  
                             &CompCode, &Reason);
```

Hizmete geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
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 üzerinde MQZ_INIT_AUTHORITY (yetkilendirme hizmetini kullanıma hazırla)

Bu işlev bir yetkilendirme hizmeti bileşeni tarafından sağlanır ve bileşenin yapılandırılması sırasında kuyruk yöneticisi tarafından çağrılır. Kuyruk yöneticisine bilgi sağlamak için MQZEP ' nin çağrılması beklenir.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_INIT_AUTHORITY.

Sözdizimi

MQZ_INIT_AUTHORITY (*Hconfig, Options, QMgrName, ComponentDataLength, ComponentData, Version, CompCode, Reason*)

Parametreler

MQZ_INIT_AUTHORITY çağrısı aşağıdaki parametreleri içeriyor.

Hconfig (MQHCONFIG)-giriş

Yapılandırma tanıtıcısı.

Bu tanıtıcı, kullanıma hazırlanmakta olan bileşeni gösterir. MQZEP işleviyle kuyruk yöneticisi çağrılırken bileşen tarafından kullanılır.

Seçenekler (MQLONG)-giriş

Başlatma seçenekleri.

Aşağıdakilerden biridir:

MQZIO_PRIMARY

Birincil başlatma.

MQZIO_IKINCIL

İkincil başlatma.

QMGrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

ComponentDataUzunluğu (MQLONG)-giriş

Bileşen verilerinin uzunluğu.

ComponentData alanının bayt cinsinden uzunluğu. Bu uzunluk, bileşen yapılandırma verilerinde tanımlanır.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu, bileşenin birincil kullanıma hazırlama işlevi çağrılmadan önce tüm sıfırlar olarak başlatılır. Bu veriler, bu belirli bileşen adına kuyruk yöneticisi tarafından tutulur; bu bileşen tarafından sağlanan işlevlerin (kullanıma hazırlama işlevi de içinde olmak üzere) herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Sürüm (MQLONG)-giriş/çıkış

Sürüm numarası.

Kullanıma hazırlama işlevine girişte bu, kuyruk yöneticisinin desteklediği *en yüksek* sürüm numarasını belirtir. Kullanıma hazırlama işlevi, gerekirse, bunu *desteklediği* arabirimin sürümüyle değiştirmelidir. Dönüşte kuyruk yöneticisi, bileşenin döndürdüğü sürümü desteklemiyorsa, bileşenin MQZ_TERM_AUTHORITY işlevini çağırır ve bu bileşeni artık kullanmaz.

Aşağıdaki değerler desteklenir:

MQZAS_VERSION_1

Sürüm 1.

MQZAS_VERSION_2

Sürüm 2.

MQZAS_VERSION_3

Sürüm 3.

MQZAS_VERSION_4

Sürüm 4.

MQZAS_VERSION_5

Sürüm 5.

MQZAS_VERSION_6

IBM WebSphere MQ 6.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_INITIALIZATION_FAILED

(2286, X'8EE') Tanımlanmamış bir nedenle kullanıma hazırlama başarısız oldu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılamıyor.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağırısı

```
MQZ_INIT_AUTHORITY (Hconfig, Options, QMgrName, ComponentDataLength,  
                    ComponentData, &Version, &CompCode,  
                    &Reason);
```

Hizmete geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
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 */
```

IBM i üzerinde MQZ_INQUIRE (sorgu yetkilendirme hizmeti)

Bu işlev bir MQZAS_VERSION_5 yetkilendirme hizmeti bileşeni tarafından sağlanır ve kuyruk yöneticisi tarafından desteklenen işlevselliği sorgulamak için çağrılır. Birden çok hizmet bileşeni kullanıldığında, hizmet bileşenleri, kuruldukları sıraya göre ters sırayla çağrılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_INQUIRE 'dir.

Sözdizimi

MQZ_INQUIRE

(QMgrName, SelectorCount, Selectors, IntAttrCount, IntAttrs, CharAttrLength, CharAttrs, SelectorReturned, ComponentData, Continuation, CompCode, Reason)

Parametreler

MQZ_INQUIRE çağırısı aşağıdaki parametreleri içeriyor.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

SelectorCount (MQLONG)-giriş

Seçici sayısı.

Selectors parametresinde sağlanan seçici sayısı.

Değer sıfır ile 256 arasında olmalıdır.

Seçiciler (MQLONG x SelectorCount)-giriş

Seçiciler.

Seçicilerin dizisi. Her seçici gerekli bir özniteliği tanımlar ve aşağıdaki tiplerden biri olmalıdır:

- MQIACF_ * (tamsayı)
- MQCACF_ * (karakter)

Seçiciler herhangi bir sırada belirtilebilir. Dizideki seçicilerin sayısı SelectorCount değiştirilmesiyle gösterilir.

Seçiciler tarafından tanımlanan tamsayı öznitelikleri, IntAttrs değiştirilmesinde Seçiciler 'de görüldükleri sırayla döndürülür.

Seçiciler tarafından tanımlanan karakter öznitelikleri, CharAttrs değiştirilmesinde, Seçiciler görüldükleriyle aynı sırayla döndürülür.

IntAttrSayı (MQLONG)-giriş

Tamsayı özniteliklerinin sayısı.

IntAttrs değiştirilmesinde verilen tamsayı özniteliklerinin sayısı.

Değer 0-256 aralığında olmalıdır.

IntAttrs (MQLONG x IntAttrSayısı)-çıkış

Tamsayı öznitelikleri.

Tamsayı öznitelikleri dizisi. Tamsayı öznitelikleri, Seçiciler dizisindeki karşılık gelen tamsayı seçiciler ile aynı sırada döndürülür.

CharAttrSayı (MQLONG)-giriş

Karakter öznitelikleri arabelleğinin uzunluğu.

CharAttrs parametresinin bayt cinsinden uzunluğu.

Değer, istenen karakter özniteliklerinin uzunluklarının en az toplamı olmalıdır. Herhangi bir karakter özniteliği istenmezse, sıfır geçerli bir değerdir.

CharAttrs (MQLONG x CharAttrSayısı)-çıkış

Karakter öznitelikleri arabelleği.

Karakter özniteliklerini içeren arabellek, birleştirildi. Karakter öznitelikleri, Seçiciler dizisindeki karşılık gelen karakter seçiciler ile aynı sırayla döndürülür.

Arabelleğin uzunluğu CharAttrCount parametresi tarafından verilir.

SelectorReturned (MQLONGxSelectorSayısı)-giriş

Seçici döndürüldü.

Seçiciler parametresindeki seçiciler tarafından istenen kümeden hangi özniteliklerin döndürüldüğünü tanımlayan değerler dizisi. Bu dizideki değerlerin sayısı SelectorCount değiştirgesiyle gösterilir. Dizideki her değer, Seçiciler dizisindeki karşılık gelen konumdan seçiciyle ilgilidir. Her değer aşağıdakilerden biridir:

MQZSL_DÖNDÜRÜLDÜ

Selectors parametresindeki ilgili seçici tarafından istenen öznitelik döndürüldü.

MQZSL_NOT_DÖNDÜRÜLMEDI

Selectors parametresindeki ilgili seçici tarafından istenen öznitelik döndürülmedi.

Dizi, *MQZSL_NOT_İADE* olarak tüm değerlerle kullanıma hazırlandı. Bir yetkilendirme hizmeti bileşeni bir öznitelik döndürdüğünde, dizideki uygun değeri *MQZSL_RETURNS* olarak ayarlar. Bu, sorgunun yapıldığı diğer yetkilendirme hizmeti bileşenlerinin hangi özniteliklerin önceden döndürüldüğünü tanımlamasına olanak sağlar.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirgesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Devam işareti.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Diğer bileşenlere bağlı olarak devam.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_UYARISI

Kısmi tamamlama.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_WARNING ise:

MQRC_CHAR_ATTRS_TOO_SHORT

Karakter öznitelikleri için yeterli alan yok.

MQRC_INT_COUNT_TOO_SMALL

Tamsayı öznitelikleri için yeterli alan yok.

CompCode MQCC_FAILED ise:

MQRC_SELECTOR_COUNT_HATA

Seçici sayısı geçerli değil.

MQRC_SELECTOR_ERROR

Öznitelik seçici geçerli değil.

MQRC_SELECTOR_LIMIT_AŞILDI

Çok fazla seçici belirtildi.

MQRC_INT_ATTR_COUNT_HATA

Tamsayı özniteliklerinin sayısı geçerli değil.

MQRC_INT_ATTRS_ARRAY_HATA

Tamsayı öznitelikleri dizisi geçersiz.

MQRC_CHAR_ATTR_LENGTH_HATA

Karakter öznitelikleri sayısı geçersiz.

MQRC_CHAR_ATTRS_HATA

Karakter öznitelikleri dizisi geçerli değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

C çağırısı

```
MQZ_INQUIRE (QMgrName, SelectorCount, Selectors, IntAttrCount,  
             &IntAttrs, CharAttrLength, &CharAttrs,  
             SelectorReturned, ComponentData, &Continuation,  
             &CompCode, &Reason);
```

Hizmete geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
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 */
```

IBM i

MQZ_REFRESH_CACHE (IBM i üzerinde tüm yetkileri yenile)

Bu işlev bir MQZAS_VERSION_3 yetkilendirme hizmeti bileşeni tarafından sağlanır. Kuyruk yöneticisi tarafından, bileşen tarafından içeride tutulan yetkiler listesini yenilemek için çağrılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_REFRESH_CACHE (8L).

Sözdizimi

MQZ_REFRESH_CACHE

(QMgrName, ComponentData, Devam, CompCode, Neden)

Parametreler

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin bileşeni herhangi bir şekilde kullanmasını gerektirmez.

ComponentData (MQBYTE x ComponentDataLength) -giriş/çıkış

Bileşen verileri.

Bu veriler, bu bileşen adına kuyruk yöneticisi tarafından alıkonur. Bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bileşenin bir sonraki işlevi çağrıldığında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının ComponentDataLength değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Bileşen tarafından ayarlanan devam göstergesi.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_REFRESH_CACHE için bu, MQZCI_CONTINUE ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici CompCode.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

C çağrısı

```
MQZ_REFRESH_CACHE (QMgrName, ComponentData,  
&Continuation, &CompCode, &Reason);
```

Değiştirgeleri aşağıdaki gibi bildirin:

```
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

IBM i üzerinde MQZ_SET_AUTHORITY (Yetki ayarla)

Bu işlev bir MQZAS_VERSION_1 yetkilendirme hizmeti bileşeni tarafından sağlanır ve bir varlığın belirtilen nesneye erişmek için sahip olduğu yetkiyi ayarlamak üzere kuyruk yöneticisi tarafından çağrılır.

Bu işleve ilişkin işlev tanıtıcısı (MQZEP için) MQZID_SET_AUTHORITY.

Not: Bu işlev, var olan yetkileri geçersiz kılar. Var olan yetkileri korumak için, bunları bu işlevle yeniden ayarlamanız gerekir.

Sözdizimi

MQZ_SET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName, Object Type, Authority, ComponentData, Continuation, CompCode, Reason)

Parametreler

MQZ_SET_AUTHORITY çağırısı aşağıdaki parametreleri içeriyor.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

EntityName (MQCHAR12)-giriş

Varlık adı.

Nesneye erişimin ayarlanacağı varlığın adı. Dizinin uzunluk üst sınırı 12 karakterdir; uzunluktan daha kısaysa, sağ tarafta boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

EntityType (MQLONG)-giriş

Varlık tipi.

EntityName tarafından belirtilen varlık tipi. Aşağıdaki değer belirtilebilir:

MQZAET_BIRINCIL KULLANICISI

Müdür bey.

MQZAET_GRUP

Grup.

ObjectName (MQCHAR48)-giriş

Nesne adı.

Erişim gerektiren nesnenin adı. Dizilimin uzunluk üst sınırı 48 karakterdir; uzunluktan daha kısaysa, sağa boşluklarla doldurulur. Ad boş değerli bir karakterle sonlandırılmadı.

ObjectType MQOT_Q_MGR ise, bu ad *QMgrName* ile aynıdır.

ObjectType (MQLONG)-giriş

Nesne tipi.

ObjectName tarafından belirtilen varlık tipi. Aşağıdakilerden biridir:

MQOT_AUTH_INFO

Kimlik doğrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sıraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

Yetki (MQLONG)-giriş

Denetlenecek yetki.

Bir yetki belirleniyorsa, bu alan uygun yetkilendirme işlemine (MQZAO_ * değişmezi) eşittir. Birden çok yetki tanımlanıyorsa, bu, ilgili MQZAO_ * değişmezlerinin bit bakımından OR işlemidir.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısının **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

Devam (MQLONG)-çıkış

Bileşen tarafından ayarlanan devam göstergesi.

Aşağıdaki değerler belirtilebilir:

MQZCI_DEFAULT

Devam, kuyruk yöneticisine bağlıdır.

MQZ_SET_AUTHORITY için bu, MQZCI_STOP ile aynı etkiye sahiptir.

MQZCI_CONTINUE

Sonraki bileşenle devam edin.

MQZCI_DURUŞU

Sonraki bileşenle devam etme.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_NOT_YETKILI

(2035, X'7F3') Erişim için yetkili değil.

MQRC_SERVICE_HATA

(2289, X'8F1') Hizmete erişilirken beklenmeyen bir hata oluştu.

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

MQRC_UNKNOWN_ENTITY

(2292, X'8F4') Hizmet için varlık bilinmiyor.

C çağırısı

```
MQZ_SET_AUTHORITY (QMgrName, EntityName, EntityType, ObjectName,  
                   ObjectType, Authority, ComponentData,  
                   &Continuation, &CompCode, &Reason);
```

Hizmete geçirilen deęiřtirgeler ařaęıdaki gibi bildirilir:

```
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-Yetkilendirme hizmetini sonlandır

Bu iřlev bir yetkilendirme hizmeti bileřeni tarafından saęlanır ve artık bu bileřenin hizmetlerini gerektirmedięi zaman kuyruk yneticisi tarafından çağırılır. İřlev, bileřenin gerektirdięi tm temizleme iřlemlerini gerçekleřtirmelidir.

Bu iřleve iliřkin iřlev tanıtıcısı (MQZEP için) MQZID_TERM_AUTHORITY.

Szdizimi

MQZ_TERM_AUTHORITY (*Hconfig, Options, QMgrName, ComponentData, CompCode, Reason*)

Parametreler

MQZ_TERM_AUTHORITY çağırısı ařaęıdaki parametreleri ieriyor.

Hconfig (MQHCONFIG)-giriř

Yapılandırma tanıtıcısı.

Bu tanıtıcı, sonlandırılmakta olan bileřeni gsterir.

Seçenekler (MQLONG)-giriş

Sonlandırma seçenekleri.

Aşağıdakilerden biridir:

MQZTO_BİRİNCİL

Birincil sonlandırma.

MQZTO_IKİNCİL

İkincil sonlandırma.

QMgrName (MQCHAR48)-giriş

Kuyruk yöneticisi adı.

Bileşeni çağıran kuyruk yöneticisinin adı. Bu ad, parametrenin tam uzunluğuna kadar boşluklarla doldurulur; ad boş karakterle sonlandırılmaz.

Kuyruk yöneticisi adı, bilgi için bileşene aktarılır; yetkilendirme hizmeti arabirimi, bileşenin tanımlanmış bir şekilde kullanılmasını gerektirmez.

ComponentData (MQBYTE x ComponentDataUzunluğu)-giriş/çıkış

Bileşen verileri.

Bu veriler, kuyruk yöneticisi tarafından bu bileşen adına tutulur; bu bileşen tarafından sağlanan işlevlerden herhangi biri tarafından yapılan değişiklikler korunur ve bu bileşenin işlevlerinden biri bir sonraki çağrılışında sunulur.

Bu veri alanının uzunluğu, MQZ_INIT_AUTHORITY çağrısında **ComponentDataLength** değiştirilmesinde kuyruk yöneticisi tarafından geçirilir.

MQZ_TERM_AUTHORITY çağrısı tamamlandığında, kuyruk yöneticisi bu verileri atar.

CompCode (MQLONG)-çıkış

Tamamlanma kodu.

Aşağıdakilerden biridir:

MQCC_OK

Başarıyla tamamlandı.

MQCC_FAILED

Arama başarısız oldu.

Neden (MQLONG)-çıkış

Neden kodu niteleyici *CompCode*.

CompCode MQCC_OK ise:

MQRC_NONE

(0, X'000 ') Raporlamak için bir neden yok.

CompCode MQCC_FAILED ise:

MQRC_SERVICE_KULLANILAMIYOR

(2285, X'8ED') Alt hizmet kullanılmıyor.

MQRC_TERMINATION_FAILED

(2287, X'8FF') Sonlandırma, tanımlanmamış bir nedenle başarısız oldu.

Bu neden kodlarına ilişkin ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

C çağrısı

```
MQZ_TERM_AUTHORITY (Hconfig, Options, QMgrName, ComponentData,  
&CompCode, &Reason);
```

Hizmete geçirilen değiştirgeler aşağıdaki gibi bildirilir:

```
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 **IBM i üzerinde MQZAC (Uygulama bağlamı)**

Bu parametre, çağırılan uygulamayla ilgili verileri belirtir.

MQZAC yapısı, **ApplicationContext** değiştirgesi için MQZ_AUTHENTICATE_USER çağrısında kullanılır.

Alanlar

StrucId (MQCHAR4)

Yapı tanıtıcısı.

Değer:

MQZAC_STRUC_ID

Uygulama bağlamı yapısına ilişkin tanıtıcı.

C programlama dili için, MQZAC_STRUC_ID_ARRAY değişmezi de tanımlanır; bu değer MQZAC_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu, hizmete ilişkin bir giriş alanıdır.

Sürüm (MQLONG)

Yapı sürümü numarası.

Değer:

MQZAC_VERSION_1

Version-1 uygulama bağlamı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZAC_CURRENT_VERSION

Uygulama bağlamı yapısının yürürlükteki sürümü.

Bu, hizmete ilişkin bir giriş alanıdır.

ProcessId (MQPID)

Süreç tanıtıcısı.

Uygulamanın süreç tanıtıcısı.

ThreadId (MQTID)

İş parçacığı tanıtıcısı.

Uygulamanın iş parçacığı tanıtıcısı.

ApplName (MQCHAR28)

Uygulama adı.

Uygulama adı.

UserID (MQCHAR12)

Kullanıcı kimliği.

IBM i sistemleri için, uygulama işinin yaratıldığı kullanıcı tanıtımı. (IBM i' da, uygulama işinde QWTSETP API ile bir tanıtım değiş tokuşu yapıldığında, yürürlükteki kullanıcı tanıtımı döndürülür).

EffectiveUserKimliği (MQCHAR12)

Etkili kullanıcı kimliği.

IBM i sistemleri için, uygulama işinin yürürlükteki kullanıcı tanıtımı.

Ortam (MQLONG)

Çevre.

Bu alan, aramanın yapıldığı ortamı belirtir.

Bu, aşağıdaki değerlerden birine sahip olabilir:

MQXE_COMMAND_SERVER

Komut sunucusu.

MQXE_MQSC

runmqsc komut yorumlayıcısı.

MQXE_MCA

İleti kanalı aracısı

MQXE_DİĞER

Tanımlanmamış ortam

CallerType (MQLONG)

Arayan Tipi.

Bu alan, çağrıyı yapan programın tipini belirler.

Bu, aşağıdaki değerlerden birine sahip olabilir:

MQXACT_EXTERNAL

Çağrı, kuyruk yöneticisinin dışında.

MQXACT_INTERNAL

Çağrı, kuyruk yöneticisinin içidir.

AuthenticationType (MQLONG)

Kimlik Doğrulama Tipi.

Bu alan, gerçekleştirilmekte olan kimlik doğrulamasının tipini belirtir.

Bu, aşağıdaki değerlerden birine sahip olabilir:

MQZAT_INITIAL_CONTEXT

Kimlik doğrulama çağrısı, kullanıcı bağlamının kullanıma hazırlanmasından kaynaklanıyor. Bu değer bir MQCONN ya da MQCONNX çağrısı sırasında kullanılır.

MQZAT_CHANGE_CONTEXT

Kimlik doğrulama çağrısı, kullanıcı bağlamının değiştirilmesinden kaynaklanıyor. Bu değer, MCA kullanıcı bağlamını değiştirdiğinde kullanılır.

v

BindType (MQLONG)

Bağ Tanımı Tipi.

Bu alan, kullanılmakta olan bağ tanımının tipini belirtir.

Bu, aşağıdaki değerlerden birine sahip olabilir:

MQCNO_FASTPATH_BINDING

Hızlı yol bağlama.

MQCNO_SHARED_BINDING

Paylaşılan bağ tanımı.

MQCNO_ISOLATED_BINDING

Yalıtılmış bağ.

C bildirimi

```
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 **IBM i üzerinde MQZAD (Yetki verileri)**

MQZAD yapısı, iki deęiřtirge için MQZ_ENUMERATE_AUTHORITY_DATA çağrısında kullanılır.

Filter ve **AuthorityBuffer** parametreleriyle ilgili daha fazla bilgi için bkz. [“IBM i üzerinde MQZ_ENUMERATE_AUTHORITY_DATA \(Yetki verilerini numaralandır\)” sayfa 1669](#) :

- MQZAD, çağrıya giriş olan **Filter** deęiřtirgesi için kullanılır. Bu parametre, çağrı tarafından döndürülen yetki verilerini seçmek için kullanılacak seçim ölçütlerini belirtir.
- MQZAD, çağrıdan çıkış olan **AuthorityBuffer** deęiřtirgesi için de kullanılır. Bu parametre, profil adı, nesne tipi ve varlığın bir birleşimine ilişkin yetkileri belirtir.

Alanlar

StrucId (MQCHAR4)

Yapı tanıtıcısı.

Deęer:

MQZAD_STRUC_ID

Yetki veri yapısına ilişkin tanıtıcı.

C programlama dili için, MQZAD_STRUC_ID_ARRAY deęiřmezi de tanımlanır; bu deęer MQZAD_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu, hizmete ilişkin bir giriş alanıdır.

Sürüm (MQLONG)

Yapı sürümü numarası.

Deęer:

MQZAD_VERSION_1

Version-1 yetki veri yapısı.

Ařağıdaki deęiřmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZAD_CURRENT_VERSION

Yetki veri yapısının yürürlükteki sürümü.

Bu, hizmete ilişkin bir giriş alanıdır.

ProfileName (MQCHAR48)

Profil adı.

Filter parametresi için bu alan, yetki verilerinin gerekli olduęu profil adıdır. Ad, alanın sonuna ya da ilk boş karakterine kadar tamamen boşsa, tüm tanıtım adlarına ilişkin yetki verileri döndürülür.

AuthorityBuffer parametresi için bu alan, belirtilen seçim ölçütüyle eşleşen bir profilin adıdır.

ObjectType (MQLONG)

Nesne tipi.

Filter parametresi için bu alan, yetki verilerinin gerekli olduęu nesne türüdür. Deęer MQOT_ALL ise, tüm nesne tipleri için yetki verileri döndürülür.

AuthorityBuffer deęiřtirgesi için bu alan, **ProfileName** ile tanıtılan tanıtımın geçerli olduęu nesne tipidir.

Değer aşağıdakilerden biridir; **Filter** değıştirgesi için MQOT_ALL değeri de geçerlidir:

MQOT_AUTH_INFO

Kimlik dođrulama bilgileri.

MQOT_KANAL

Kanal.

MQOT_CLNTCONN_CHANNEL

İstemci bağlantı kanalı.

MQOT_LISTENER

Dinleyici.

MQOT_NAMELIST

-Namelist.

MQOT_PROCESS

Süreç tanımlaması.

MQOT_Q

Sraya girin.

MQOT_Q_MGR

Kuyruk yöneticisi.

MQOT_SERVICE

Servis.

Yetki (MQLONG)

Otorite.

Filter parametresi için bu alan yoksayılr.

AuthorityBuffer parametresi için bu alan, varlığın **ProfileName** ve **ObjectType** ile tanımlanan nesnelere için sahip olduđu yetkileri temsil eder. Varlığın tek bir yetkisi varsa, alan uygun yetki değeri (MQZAO_* değışmezi) eşittir. Varlığın birden çok yetkisi varsa, alan ilgili MQZAO_* sabitlerinin bit bakımından OR işlecidir.

EntityDataPtr (PMQZED)

Bir varlığı tanımlayan MQZED yapısının adresi.

Filter değıştirgesi için bu alan, yetki verilerinin gerekli olduđu varlığı tanımlayan bir MQZED yapısını gösterir. **EntityDataPtr** boş değeri işaretçiyse, tüm varlıklar için yetki verileri döndürölür.

AuthorityBuffer değıştirgesi için bu alan, döndürölen yetki verilerinin geldiđi varlığı tanıyan bir MQZED yapısını gösterir.

EntityType (MQLONG)

Varlık tipi.

Filter parametresi için bu alan, yetki verilerinin gerekli olduđu varlık tipini belirtir. Deđer MQZAET_NONE ise, tüm varlık tipleri için yetki verileri döndürölür.

AuthorityBuffer değıştirgesi için bu alan, **EntityDataPtr** ile gösterilen MQZED yapısıyla tanıyan varlığın tipini belirtir.

Deđer aşağıdakilerden biridir; **Filter** parametresi için MQZAET_NONE değeri de geçerlidir:

MQZAET_BIRINCIL KULLANICISI

Müdürebey.

MQZAET_GRUP

Grup.

Seçenekler (MQAUTHOPT)

Seçenekler.

Bu alan, görüntölenen tanımlar üzerinde denetim sağlayan seçenekleri belirler.

Aşğıdakilerden biri belirtilmelidir:

MQAUTHOPT_NAME_ALL_EŞLEŞME

Tüm tanımları görüntüler

MQAUTHOPT_NAME_EXPLICIT

ProfileName alanında belirtilenle tam olarak aynı ada sahip tanımları görüntüler.

Ayrıca, aşağıdakilerden biri de belirtilmelidir:

MQAUTHOPT_ENTITY_SET

Varlığın, **ProfileName** tarafından belirtilen nesne üzerinde sahip olduğu kümülatif yetkiyi hesaplamak için kullanılan tüm profilleri görüntüler. **ProfileName** alanı herhangi bir genel arama karakteri içermemelidir.

- Belirtilen varlık bir birincil kullanıcıysa, kümenin her üyesi için {entity, groups} nesne için geçerli olan en uygulanabilir profil görüntülenir.
- Belirtilen varlık bir grupsa, nesneye uygulanan gruptaki en uygulanabilir profil görüntülenir.
- Bu değer belirtilirse, **ProfileName**, **ObjectType**, **EntityType** değerlerinin ve **EntityDataPtr** MQZED yapısında belirtilen varlık adının tümü boş olmamalıdır.

MQAUTHOPT_NAME_ALL_EŞLEŞME belirttiyseniz, aşağıdakileri de belirtebilirsiniz:

MQAUTHOPT_ENTITY_EXPLICIT

EntityDataPtr MQZED yapısında belirtilen varlık adıyla tam olarak aynı varlık adına sahip profilleri görüntüler.

C bildirimi

```
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 üzerinde MQZED (Varlık tanımlayıcı)

MQZED yapısı, yetkilendirmenin denetleneceği varlığı belirtmek için bir dizi yetkilendirme hizmeti çağrısında kullanılır.

Alanlar

StrucId (MQCHAR4)

Yapı tanıtıcısı.

Değer:

MQZED_STRUC_ID

Varlık tanımlayıcı yapısına ilişkin tanıtıcı.

C programlama dili için, MQZED_STRUC_ID_ARRAY değişmezi de tanımlanır; bu değer MQZED_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu, hizmete ilişkin bir giriş alanıdır.

Sürüm (MQLONG)

Yapı sürümü numarası.

Değer:

MQZED_VERSION_1

Version-1 varlık tanımlayıcı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZED_CURRENT_VERSION

Varlık tanımlayıcı yapısının yürürlükteki sürümü.

Bu, hizmete ilişkin bir giriş alanıdır.

EntityNamePtr (PMQCHAR)

Varlık adının adresi.

Bu, yetkilendirmesi denetlenecek varlığın adını gösteren bir işaretçidir.

EntityDomainPtr (PMQCHAR)

Varlık etki alanı adının adresi.

Bu, yetkisi denetlenecek varlığın tanımını içeren etki alanının adını gösteren bir işaretçidir.

SecurityId (MQBYTE40)

Güvenlik tanıtıcısı.

Bu, yetkisi denetlenecek olan güvenlik tanıtıcısıdır.

CorrelationPtr (MQPTR)

İlinti göstergesi.

Bu, kimlik doğrulama kullanıcı işlevi ile diğer uygun OAM işlevleri arasında korelasyon verilerinin aktarılmasını kolaylaştırır.

C bildirim

```
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 */  
};
```

IBM i üzerinde MQZFP (Serbest değişirgeler)

Bu değişirge, serbest bırakılacak kaynakla ilgili verileri belirtir.

MQZFP yapısı, **FreeParms** değişirgesine ilişkin MQZ_FREE_USER çağrısında kullanılır.

Alanlar

StrucId (MQCHAR4)

Yapı tanıtıcısı.

Değer:

MQZFP_STRUC_ID

Serbest parametre yapısına ilişkin tanıtıcı.

C programlama dili için, değişmez MQZFP_STRUC_ID_ARRAY da tanımlanır; bu değer MQZFP_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu, hizmete ilişkin bir giriş alanıdır.

Sürüm (MQLONG)

Yapı sürümü numarası.

Değer:

MQZFP_VERSION_1

Version-1 serbest deęiřtirge yapısı.

Ařaęıdaki deęiřmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZFP_CURRENT_VERSION

Serbest parametre yapısının geçerli sürümü.

Bu, hizmete iliřkin bir giriř alanıdır.

Ayrılmıř (MQBYTE8)

Ayrılmıř alan.

İlk deęer boř.

CorrelationPtr (MQPTR)

İlinti göstergesi.

Serbest bırakılacak kaynakla ilgili ilinti verilerinin adresi.

C bildirimi

```
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

IBM i üzerinde MQZIC (Kimlik baęlamı)

MQZIC yapısı, **IdentityContext** deęiřtirgesi için MQZ_AUTHENTICATE_USER çağrısında kullanılır.

MQZIC yapısı, iletiyi bir kuyruęa ilk koyan uygulamanın kullanıcıını tanıtan kimlik baęlamı bilgilerini içerir:

- Kuyruk yöneticisi, UserIdentifier alanını kullanıcıyı tanıtan bir adla doldurur; kuyruk yöneticisinin bunu yapabilme řekli, uygulamanın çalıştıęı ortama baęlıdır.
- Kuyruk yöneticisi, AccountingToken alanını, iletiyi koyan uygulamadan belirledięi bir simge ya da numarayla doldurur.
- Uygulamalar, kullanıcı hakkında (örneęin, řifrelenmiř bir parola) eklemek istedikleri ek bilgiler için ApplIdentityVeri alanını kullanabilir.

Uygun yetkili uygulamalar, MQZ_AUTHENTICATE_USER iřlevini kullanarak kimlik baęlamını ayarlayabilir.

IBM MQ for Windowsaltında bir ileti oluřturulduęunda, bir Windows sistem güvenlik tanıtıcısı (SID) AccountingToken alanında saklanır. SID, UserIdentifier alanını tamamlamak ve bir kullanıcının kimlik bilgilerini oluřturmak için kullanılabilir.

Alanlar

StrucId (MQCHAR4)

Yapı tanıtıcısı.

Deęer:

MQZIC_STRUC_ID

Kimlik baęlamı yapısına iliřkin tanıtıcı.

C programlama dili için, MQZIC_STRUC_ID_ARRAY deęiřmezi de tanımlanır; bu deęer MQZIC_STRUC_ID ile aynıdır, ancak dizgi yerine bir karakter dizisidir.

Bu, hizmete iliřkin bir giriř alanıdır.

Sürüm (MQLONG)

Yapı sürümü numarası.

Değer:

MQZIC_VERSION_1

Version-1 kimlik bağlamı yapısı.

Aşağıdaki değişmez, yürürlükteki sürümün sürüm numarasını belirtir:

MQZIC_CURRENT_VERSION

Kimlik bağlamı yapısının yürürlükteki sürümü.

Bu, hizmete ilişkin bir giriş alanıdır.

UserIdentifier (MQCHAR12)

Kullanıcı kimliği.

Bu, iletinin **kimlik bağlamının** bir parçasıdır.

UserIdentifier, iletiyi oluşturan uygulamanın kullanıcı kimliğini belirtir. Kuyruk yöneticisi bu bilgileri karakter verileri olarak kabul eder, ancak biçimini tanımlamaz. *UserIdentifier* alanıyla ilgili daha fazla bilgi için bkz. [“MQMD için UserIdentifier \(MQCHAR12\)” sayfa 455.](#)

AccountingToken (MQBYTE32)

Hesap simgesi.

Bu, iletinin **kimlik bağlamının** bir parçasıdır.

AccountingToken, bir uygulamanın, iletinin sonucu olarak yapılan işlerin uygun şekilde ücretlendirilmesine neden olmasını sağlar. Kuyruk yöneticisi bu bilgileri bir bit dizgisi olarak işler ve içeriğini denetlemez. *AccountingToken* alanıyla ilgili daha fazla bilgi için bkz. [“MQMD için AccountingToken \(MQBYTE32\)” sayfa 456.](#)

ApplIdentityVerileri (MQCHAR32)

Kimlikle ilgili uygulama verileri.

Bu, iletinin **kimlik bağlamının** bir parçasıdır.

ApplIdentityData, uygulama grubu tarafından tanımlanan ve iletinin kökenine ilişkin ek bilgi sağlamak için kullanılacak bilgilerdir. Örneğin, kimlik verilerinin güvenilir olup olmadığını belirtmek için uygun kullanıcı yetkisiyle çalışan uygulamalar tarafından ayarlanabilir. *ApplIdentityData* alanıyla ilgili daha fazla bilgi için bkz. [“MQMD için ApplIdentityVerileri \(MQCHAR32\)” sayfa 458.](#)

C bildiri

```
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 */
};
```

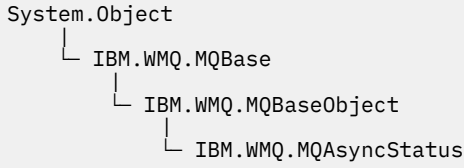
IBM MQ .NET sınıfları ve arabirimleri

IBM MQ .NET sınıfları ve arabirimleri alfabetik olarak listelenir. Özellikler, yöntemler ve oluşturucular açıklanır.

MQAsyncStatus.NET sınıfı

Önceki MQI etkinliğinin durumunu sorgulamak için *MQAsyncStatus* kullanın; örneğin, önceki zamanuyumsuz koyma işlemlerinin başarısını sorgulamak için. *MQAsyncStatus*, MQSTS veri yapısının özelliklerini içerir.

Sınıf



```
public class IBM.WMQ.MQAsyncStatus extends IBM.WMQ.MQBaseObject;
```

- “Özellikler” sayfa 1695
- “Oluşturucular” sayfa 1696

Özellikler

Özellikler alınırken MQException testi yapılıyor.

```
public static int CompCode {get;}
```

İlk hata ya da uyarının tamamlanma kodu.

```
public static int Reason {get;}
```

İlk hata ya da uyarının neden kodu.

```
public static int PutSuccessCount {get;}
```

Başarılı zamanuyumsuz MQI koyma çağrılarının sayısı.

```
public static int PutWarningCount {get;}
```

Bir uyarıyla başarılı olan zamanuyumsuz MQI koyma çağrılarının sayısı.

```
public static int PutFailureCount {get;}
```

Başarısız olan zamanuyumsuz MQI koyma çağrılarının sayısı.

```
public static int ObjectType {get;}
```

İlk hataya ilişkin nesne tipi. Aşağıdaki değerler kullanılabilir:

- MQC.MQOT_ALIAS_Q
- MQC.MQOT_LOCAL_Q
- MQC.MQOT_MODEL_Q
- MQC.MQOT_Q
- MQC.MQOT_REMOTE_Q
- MQC.MQOT_TOPIC
- 0, herhangi bir nesnenin döndürülmediği anlamına gelir

```
public static string ObjectName {get;}
```

Nesne adı.

```
public static string ObjectQMgrName {get;}
```

Nesne kuyruğu yöneticisi adı.

```
public static string ResolvedObjectName {get;}
```

Çözümlenen nesne adı.

```
public static string ResolvedObjectQMgrName {get;}
```

Çözülen nesne kuyruğu yöneticisi adı.

Oluşturucular

public MQAsyncStatus() throws MQException;

Oluşturucu yöntemi, alanları uygun şekilde sıfır ya da boş olarak kullanıma hazırlanmış bir nesne oluşturur.

MQAuthenticationInformationRecord.NET sınıfı

IBM MQ TLS istemci bağlantısında kullanılacak bir kimlik doğrulayıcıya ilişkin bilgileri belirtmek için MQAuthenticationInformationRecord kullanın. MQAuthenticationInformationRecord, bir kimlik doğrulama bilgileri kaydını (MQAIR) içerir.

Sınıf

```
System.Object
└─ IBM.WMQ.MQAuthenticationInformationRecord
```

```
public class IBM.WMQ.MQAuthenticationInformationRecord extends System.Object;
```

- [“Özellikler” sayfa 1696](#)
- [“Oluşturucular” sayfa 1697](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

public long Version {get; set;}

Yapı sürümü numarası.

public long AuthInfoType {get; set;}

Kimlik doğrulama bilgilerinin tipi. Bu öznitelik aşağıdaki değerlerden birine ayarlanmalıdır:

- OCSP -Sertifika iptal durumu denetimi OCSP kullanılarak yapılır.
- CRLLDAP -Sertifika iptal durumu denetimi, LDAP sunucularında Sertifika İptal Listeleri kullanılarak yapılır.

public string AuthInfoConnName {get; set;}

İsteğe bağlı bir kapı numarasıyla, LDAP sunucusunun çalıştığı anasistemin DNS adı ya da IP adresi. Bu anahtar sözcük gereklidir.

public string LDAPPassword {get; set;}

LDAP sunucusuna erişen kullanıcının ayırt edici adıyla ilişkili parola. Bu özellik yalnızca **AuthInfoType**, CRLLDAPolarak ayarlandığında geçerlidir.

public string LDAPUserName {get; set;}

LDAP sunucusuna erişen kullanıcının ayırt edici adı. Bu özelliği ayarladığınızda, LDAPUserNameLength ve LDAPUserNamePtr otomatik olarak doğru ayarlanır. Bu özellik yalnızca AuthInfoType CRLLDAPolarak ayarlandığında geçerlidir.

public string OCSPResponderURL {get; set;}

OCSP yanıtlayıcısıyla iletişim kurulabilecek URL . Bu özellik yalnızca AuthInfoTip OCSP olarak ayarlandığında geçerlidir.

Bu alan büyük ve küçük harfe duyarlıdır. Küçük harfli http:// dizjisiyle başlamalıdır. URL ' nin geri kalanı, OCSP sunucusu uygulamasına bağlı olarak büyük ve küçük harfe duyarlı olabilir.

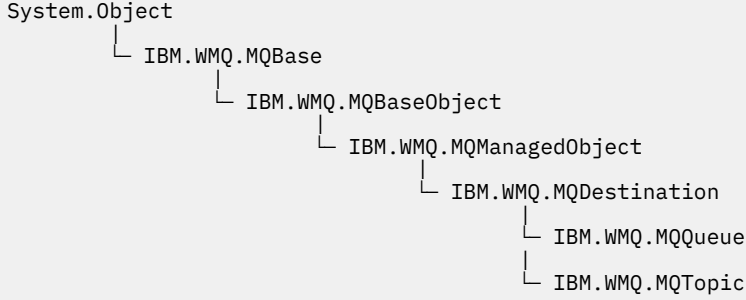
Olusturucular

MQAuthenticationInformationRecord();

MQDestination.NET sınıfı

MQQueue ve MQTopic için ortak olan yöntemlere erişmek üzere MQDestination komutunu kullanın. MQDestination soyut bir temel sınıftır ve somutlaştırılmaz.

Sınıf



```
public class IBM.WMQ.MQDestination extends IBM.WMQ.MQManagedObject;
```

- [“Özellikler” sayfa 1697](#)
- [“yöntemler” sayfa 1697](#)
- [“Olusturucular” sayfa 1699](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

```
public DateTime CreationDateTime {get;}
```

Kuyruğun ya da konunun yaratıldığı tarih ve saat. Başlangıçta MQQueue içinde bulunan bu özellik, temel MQDestination sınıfına taşınmıştır.

Varsayılan değer yoktur.

```
public int DestinationType {get;}
```

Kullanılmakta olan hedefin tipini tanımlayan tamsayı değeri. MQQueue ya da MQTopic alt sınıf oluşturucusundan kullanıma hazırlanan bu değer aşağıdaki değerlerden birini alabilir:

- MQOT_Q
- MQOT_TOPIC

Varsayılan değer yoktur.

yöntemler

```
public void Get(MQMessage message);  
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions);  
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions, int MaxMsgSize);
```

MQException ögesini atar.

Hedef bir MQQueue nesnesiyse ya da hedef bir MQTopic nesnesiyse, alma işlemini gerçekleştirmek için varsayılan bir MQGetMessageOptions eşgörünümü kullanarak kuyruktan bir ileti alır.

Alma işlemi başarısız olursa, MQMessage nesnesi değişmez. Başarılı olursa, MQMessage ileti tanımlayıcısı ve ileti verileri bölümleri, gelen iletideki ileti tanımlayıcısı ve ileti verileriyle değiştirilir.

Belirli bir MQQueueManager içinden IBM MQ ' e yapılan tüm çağrılar zamanuyumdur. Bu nedenle, beklemeyle alma işlemi gerçekleştirirseniz, aynı MQQueueManager ögesini kullanan diğer tüm iş parçacıklarının, Get çağrısı tamamlanmaya kadar başka IBM MQ çağrısı yapması engellenir. IBM MQ ' e aynı anda erişmek için birden çok iş parçacığına gereksinim duyarsanız, her iş parçacığı kendi MQQueueManager nesnesini oluşturmalıdır.

ileti

İleti tanımlayıcıyı ve döndürülen ileti verilerini içerir. İleti tanımlayıcıdaki alanlardan bazıları giriş değiştirgeleridir. MessageId ve CorrelationId giriş parametrelerinin gerektiği gibi ayarlandığından emin olmak önemlidir.

Yeniden bağlanabilir bir istemci, MQGM_SYNCPOINT altında alınan iletiler için başarılı bir yeniden bağlantıdan sonra MQRC_BACKED_OUT neden kodunu döndürür.

getMessageSeçenekleri

Alma işlemi denetleyen seçenekler.

MQC.MQGMO_CONVERT seçeneğinin kullanılması, tek baytlık karakter kodlarından çift baytlık kodlara dönüştürülürken MQC.MQRC_CONVERTED_STRING_TOO_BIG neden koduyla bir kural dışı durumla sonuçlanabilir. Bu durumda, ileti dönüştürme yapılmadan arabelleğe kopyalanır.

getMessageOptions belirtilmezse, kullanılan ileti seçeneği MQGMO_NOWAIT olur.

Yeniden bağlanabilir bir istemcide MQGMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

MaxMsgBoyutu

Bu ileti nesnesinin alacağı en büyük ileti. Kuyruktaki ileti bu boyuttan büyükse, iki şeyden biri oluşur:

- MQGetMessageOptions nesnesinde MQGMO_ACCEPT_TRUNCATED_MSG işareti ayarlanırsa, ileti mümkün olduğunca çok ileti verileriyle doldurulur. MQCC_WARNING tamamlanma kodu ve MQRC_TRUNCATED_MSG_ACCEPTED neden koduyla bir kural dışı durum yayınlandı.
- MQGMO_ACCEPT_TRUNCATED_MSG işareti ayarlanmazsa, ileti kuyrukta kalır. MQCC_WARNING tamamlanma kodu ve MQRC_TRUNCATED_MSG_FAILED neden koduyla bir kural dışı durum yayınlandı.

MaxMsgSize belirtilmezse, iletinin tamamı alınır.

```
public void Put(MQMessage message);  
public void Put(MQMessage message, MQPutMessageOptions putMessageOptions);
```

MQException ögesini atar.

Hedef bir MQQueue nesnesiyse bir iletiyi kuyruğa koyar ya da hedef bir MQTopic nesnesiyse bir iletiyi bir konuya yayınlar.

Koyma çağrısı tamamlandıktan sonra MQMessage nesnesinde yapılan değişiklikler, IBM MQ kuyruğundaki ya da yayın konusundaki gerçek iletiyi etkilemez.

Put, MQMessage nesnesinin MessageId ve CorrelationId özelliklerini günceller ve ileti verilerini temizlemez. Diğer Put ya da Get çağrıları, MQMessage nesnesindeki güncellenen bilgilere başurmaktadır. Örneğin, aşağıdaki kod parçacığına ilk ileti a ve ikinci abıçerir.

```
msg.WriteString("a");  
q.Put(msg, pmo);  
msg.WriteString("b");  
q.Put(msg, pmo);
```

ileti

Gönderilecek ileti tanımlayıcı verilerini ve iletiyi içeren bir MQMessage nesnesi. İleti tanımlayıcısı, bu yöntemin sonucu olarak değiştirilebilir. Bu yöntem tamamlandıktan hemen sonra ileti tanımlayıcısındaki değerler, kuyruğa konan ya da konuya yayınlanan değerlerdir.

Yeniden bağlanabilir bir istemciye aşağıdaki neden kodları döndürülür:

- MQRC_CALL_INTERRUPTED kalıcı bir iletide Put çağrısı çalıştırılırken bağlantı kesilirse ve yeniden bağlantı başarılı olursa.
- MQRC_NONE kalıcı olmayan bir iletide Put çağrısı çalıştırılırken bağlantı başarılı olursa (bkz. [Application Recovery](#)).

putMessageSeçenekleri

Koyma işleminin eylemini denetleyen seçenekler.

putMessageOptions belirtilmezse, varsayılan MQPutMessageOptions eşgörünümü kullanılır.

Yeniden bağlanabilir bir istemcide MQPMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

Not: Kolaylık ve başarı için, kuyruğa tek bir ileti koymak istiyorsanız MQQueueManager.Put nesnesini kullanın. Bunun için bir MQQueue nesnesine sahip olun.

Oluşturucular

MQDestination soyut bir temel sınıftır ve somutlaştırılmaz. MQQueue ve MQTopic oluşturucuları kullanarak ya da MQQueueManager.AccessQueue ve MQQueueManager.AccessTopic methodskullanarak hedeflere erişin.

MQEnvironment.NET sınıfı

MQQueueManager oluşturucusunun nasıl çağrılacağını denetlemek ve bir IBM MQ MQI client bağlantısı seçmek için MQEnvironment komutunu kullanın. MQEnvironment sınıfı, IBM MQ' nin davranışını denetleyen özellikleri içerir.

Sınıf

```
System.Object
├── IBM.WMQ.MQEnvironment
```

```
public class IBM.WMQ.MQEnvironment extends System.Object;
```

- [“Özellikler-yalnızca istemci” sayfa 1699](#)
- [“Özellikler” sayfa 1700](#)
- [“Oluşturucular” sayfa 1701](#)

Özellikler-yalnızca istemci

Özellikler alınırken MQException testi yapılıyor.

```
public static int CertificateValPolicy {get; set;}
```

Uzak iş ortağı sistemlerinden alınan dijital sertifikaları doğrulamak için hangi TLS sertifika doğrulama ilkesinin kullanılacağını ayarlayın. Geçerli değerler şunlardır:

- MQC.CERTIFICATE_VALIDATION_POLICY_ANY
- MQC.CERTIFICATE_VALIDATION_POLICY_RFC5280

```
public static ArrayList EncryptionPolicySuiteB {get; set;}
```

Suite B uyumlu şifreleme düzeyini ayarlayın. Geçerli değerler şunlardır:

- MQC.MQ_SUITE_B_NONE -Varsayılan değer budur.
- MQC.MQ_SUITE_B_128_BIT
- MQC.MQ_SUITE_B_192_BIT

public static string Channel {get; set;}

Hedef kuyruk yöneticisine bağlanacak kanalın adı. İstemci kipinde bir MQQueueManager yönetim ortamını somutlaştırmadan önce kanal özelliğini ayarlamanız gerekir.

public static int FipsRequired {get; set;}

Şifreleme IBM MQ içinde gerçekleştiriliyorsa yalnızca FIPS onaylı algoritmaları kullanmak için MQC.MQSSL_FIPS_YES değerini belirleyin. Varsayılan değer MQC.MQSSL_FIPS_NO' dir.

Şifreleme donanımı yapılandırıldıysa, kullanılan şifreleme modülleri donanım ürünü tarafından sağlanan modüllerdir. Kullanılmakta olan donanıma bağlı olarak, bunlar belirli bir düzeyde FIPS onaylı olmayabilir.

public static string Hostname {get; set;}

IBM MQ sunucusunun bulunduğu bilgisayarın TCP/IP anasistem adı. Anasistem adı ayarlanmazsa ve geçersiz kılan özellikler ayarlanmazsa, yerel kuyruk yöneticisine bağlanmak için sunucu bağ tanımları kipi kullanılır.

public static int Port {get; set;}

Bağlantı kurulacak kapı. Bu, IBM MQ sunucusunun gelen bağlantı isteklerini dinlediği kapıdır. Varsayılan değer 1414 'tür.

public static string SSLCipherSpec {get; set;}

Bağlantı için TLS ' yi etkinleştirmek üzere SVRCONN kanalında SSLCipherSpec ayarını CipherSpec değerine ayarlayın. Varsayılan değer Boş değerdir ve TLS bağlantı için etkinleştirilmez.

public static string sslPeerName {get; set;}

Ayırt edici ad kalıbı. sslCipherSpec ayarlandıysa, bu değişken doğru kuyruk yöneticisinin kullanıldığından emin olmak için kullanılabilir. Boş değere (varsayılan) ayarlanırsa, kuyruk yöneticisinin ayırt edici adı (DN) gerçekleştirilmez. sslPeerName , sslCipherSpec boş değerliyse yoksayılır.

V 9.3.0 > V 9.3.0 public static string SSLKeyRepository {get; set;}

Anahtar havuzuna erişmek için kullanılan düz metin ya da şifreli parola. Anahtar havuzu geçiş tümceleri, **runmqicred** yardımcı programı kullanılarak istemci uygulamaları tarafından kullanılmak üzere şifrelenir.

SSLKeyRepositoryPassword boş (varsayılan) olarak ayarlanırsa, *MQKEYRPWD* ortam değişkeninin ya da istemci yapılandırma dosyasındaki **SSLKeyRepositoryPassword** özniteliğinin değeri kullanılır.

V 9.3.0 > V 9.3.0 public static string InitialKey {get; set;}

SSLKeyRepositoryPassword içinde belirtilen anahtar havuzu parolasını şifrelemek için kullanılan ilk anahtar.

Anahtar havuzu parolası **runmqicred** yardımcı programı kullanılarak şifrelendiğinde bir ilk anahtar dosyası belirtilirse, ilk anahtar belirtilmelidir.

Özellikler

Özellikler alınırken MQException testi yapılıyor.

public static ArrayList HdrCompList {get; set;}

Üstbilgi Veri Sıkıştırma Listesi

public static int KeyResetCount {get; set;}

Gizli anahtar yeniden anlaşılmadan önce bir TLS etkileşimi içinde gönderilen ve alınan şifrelenmemiş bayt sayısını belirtir.

public static ArrayList MQAIRArray {get; set;}

Bir MQAuthenticationInformationRecord nesnelere dizisi.

public static ArrayList MsgCompList {get; set;}

İleti Verileri Sıkıştırma Listesi

public static string Password {get; set;}

Kimliği doğrulanacak parola. MQCSP yapısından gönderme yapılan parolaya bu Parola özelliği ayarlanarak veri yerleştirilir.

public static string ReceiveExit {get; set;}

Alma çıkışı, bir kuyruk yöneticisinden alınan verileri incelemenizi ve değiştirmenizi sağlar. Olağan durumda, kuyruk yöneticisinde karşılık gelen bir gönderme çıkışıyla birlikte kullanılır. ReceiveExit boş değere ayarlanırsa, alma çıkışı çağrılmaz.

public static string ReceiveUserData {get; set;}

Bir alma çıkışıyla ilişkili kullanıcı verileri. 32 karakterle sınırlıdır.

public static string SecurityExit {get; set;}

Güvenlik çıkışı, bir kuyruk yöneticisine bağlanma girişiminde bulunulduğunda oluşan güvenlik akışlarını uyarlamenizi sağlar. SecurityExit boş değere ayarlanırsa, güvenlik çıkışı çağrılmaz.

public static string SecurityUserData {get; set;}

Bir güvenlik çıkışıyla ilişkili kullanıcı verileri. 32 karakterle sınırlıdır.

public static string SendExit {get; set;}

Gönderme çıkışı, bir kuyruk yöneticisine gönderilen verileri incelemenizi ya da değiştirmenizi sağlar. Olağan durumda, kuyruk yöneticisinde karşılık gelen bir alma çıkışı ile birlikte kullanılır. SendExit boş değere ayarlanırsa, gönderme çıkışı çağrılmaz.

public static string SendUserData {get; set;}

Bir gönderme çıkışıyla ilişkili kullanıcı verileri. 32 karakterle sınırlıdır.

public static string SharingConversations {get; set;}

SharingConversations alanı, bu uygulamalar bir istemci kanal tanımlama çizelgesi (CCDT) kullanmadığında .NET uygulamalarından gelen bağlantılarda kullanılır.

SharingConversations , bu bağlantıyla ilişkilendirilmiş bir yuvada paylaşılacak etkileşim sayısı üst sınırını belirler.

0 değeri, iletişim paylaşımı, önden okuma ve sağlıklı işletim ile ilgili olarak kanalın IBM WebSphere MQ 7.0' den önceki gibi çalıştığı anlamına gelir.

Alan, IBM MQ kuyruk yöneticisi somutlaştırılırken özelliklerin HASH çizelgesine SHARING_CONVERSATIONS_PROPERTY olarak geçirilir.


SharingConversations belirtmezseniz, varsayılan değer olan 10 kullanılır.

public static string SSLCryptoHardware {get; set;}

Sistemde bulunan şifreleme donanımının konfigürasyonunu tanımlamak için gereken parametre diziliminin adını belirler. SSLCryptoHardware , sslCipherSpec boş değerliyse yoksayılır.

public static string SSLKeyRepository {get; set;}

Anahtar havuzunun tam olarak nitelenmiş dosya adını ayarlayın.

 Dosya uzantısı sağlanmazsa, bunun .kdb olduğu varsayılır.

SSLKeyRepository boş değere (varsayılan) ayarlanırsa, anahtar havuzunu bulmak için MQSSLKEYR ortam değişkeni kullanılır.

public static string UserId {get; set;}

Kimliği doğrulanacak kullanıcı kimliği. MQCSP yapısında gönderme yapılan kullanıcı kimliğine UserId ayarlanarak veri yerleştirilir. Bir API ya da Güvenlik çıkışı kullanarak UserId kimliğini doğrulayın.

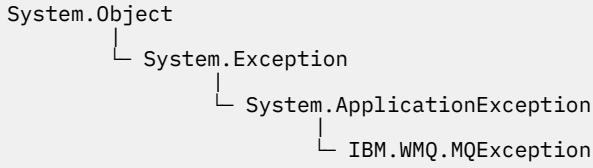
Oluşturuçular

public MQEnvironment()

MQException.NET sınıfı

Başarısız olan bir IBM MQ işlevinin tamamlanma ve neden kodlarını öğrenmek için MQException komutunu kullanın. Bir IBM MQ hatası oluştuğunda MQException yayınlanır.

Sınıf



```
public class IBM.WMQ.MQException extends System.ApplicationException;
```

- “[Özellikler](#)” sayfa 1702
- “[Oluşturucular](#)” sayfa 1702

Özellikler

```
public int CompletionCode {get; set;}
```

Hatayla ilişkili IBM MQ tamamlama kodu. Olası değerler şunlardır:

- MQException.MQCC_OK
- MQException.MQCC_WARNING
- MQException.MQCC_FAILED

```
public int ReasonCode {get; set;}
```

Hatayı açıklayan IBM MQ neden kodu.

Oluşturucular

```
public MQException(int completionCode, int reasonCode)
```

completionCode

IBM MQ tamamlama kodu.

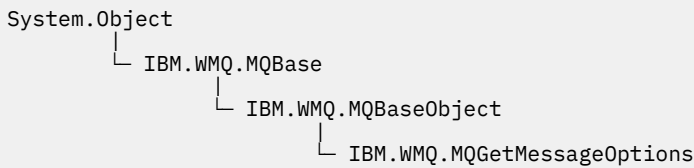
reasonCode

IBM MQ tamamlama kodu.

MQGetMessageOptions.NET sınıfı

İletilerin nasıl alınacağını belirtmek için MQGetMessageOptions kullanın. MQDestination.Getdavranışını değiştirir.

Sınıf



```
public class IBM.WMQ.MQGetMessageOptions extends IBM.WMQ.MQBaseObject;
```

- “[Özellikler](#)” sayfa 1702
- “[Oluşturucular](#)” sayfa 1705

Özellikler

Not: Bu sınıftaki bazı seçeneklerin davranışı, bu seçeneklerin kullanıldığı ortama bağlıdır. Bu öğeler yıldız işaretiyle * işaretlenir.

Özellikler alınırken MQException testi yapılıyor.

public int GroupStatus {get;}*

GroupStatus , alınan iletinin bir grupta olup olmadığını ve bu iletinin gruptaki son ileti olup olmadığını gösterir. Olası değerler şunlardır:

MQC.MQGS_LAST_MSG_IN_GROUP

İleti, gruptaki son ya da tek iletidir.

MQC.MQGS_MSG_IN_GROUP

İleti bir grupta, ancak gruptaki son ileti değil.

MQC.MQGS_NOT_IN_GROUP

İleti bir grupta değil.

public int MatchOptions {get; set;}*

MatchOptions iletinin nasıl seçileceğini belirler. Aşağıdaki eşleşme seçenekleri ayarlanabilir:

MQC.MQMO_MATCH_CORREL_ID

Eşleştirilecek ilinti tanıtıcısı.

MQC.MQMO_MATCH_GROUP_ID

Eşleştirilecek grup tanıtıcısı.

MQC.MQMO_MATCH_MSG_ID

Eşleştirilecek ileti tanıtıcısı.

MQC.MQMO_MATCH_MSG_SEQ_NUMBER

İleti sıra numarasını eşleştir.

MQC.MQMO_NONE

Eşleştirme gerekmez.

public int Options {get; set;}

Seçenekler , MQQueue . getylemini denetler. Aşağıdaki değerlerden herhangi biri belirtilebilir. Birden çok seçenek gerekiyorsa, değerler bit açısından OR işleci kullanılarak eklenebilir ya da birleştirilebilir.

MQC.MQGMO_ACCEPT_TRUNCATED_MSG

İleti verilerinin kesilmesine izin ver.

MQC.MQGMO_ALL_MSGS_AVAILABLE*

Bir gruptaki iletileri yalnızca gruptaki tüm iletiler kullanılabilir olduğunda alın.

MQC.MQGMO_ALL_SEGMENTS_AVAILABLE*

Mantıksal bir iletinin bölümlerini yalnızca gruptaki tüm bölümler kullanılabilir olduğunda alın.

MQC.MQGMO_BROWSE_FIRST

Kuyruğun başlangıcından itibaren göz atın.

MQC.MQGMO_BROWSE_MSG_UNDER_CURSOR*

Göz atma imlecinin altındaki iletiye göz atın.

MQC.MQGMO_BROWSE_NEXT

Kuyruktaki yürürlükteki konumdan göz atın.

MQC.MQGMO_COMPLETE_MSG*

Yalnızca tam mantıksal iletileri alın.

MQC.MQGMO_CONVERT

Veriler ileti arabelleğine kopyalanmadan önce, MQMessage' un CharSet ve Encoding özniteliklerine uymak için uygulama verilerinin dönüştürülmesini isteyin. Veriler ileti arabelleğinden alındığında veri dönüştürme de uygulandığından, uygulamalar bu seçeneği ayarlamaz.

Bu seçeneğin kullanılması, tek baytlık karakter takımlarından çift baytlık karakter takımlarına dönüştürme sırasında sorunlara neden olabilir. Bunun yerine, ileti teslim edildikten sonra readString, readLineve writeString yöntemlerini kullanarak dönüştürmeyi gerçekleştirin.

MQC.MQGMO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturulursa başarısız olur.

MQC.MQGMO_LOCK*

Göz atılan iletiyi kilitleyin.

MQC.MQGMO_LOGICAL_ORDER*

İletileri gruplar halinde ve mantıksal iletilerin bölümlerini mantıksal sırayla döndürür.

Yeniden bağlanabilir bir istemcide MQGMO_LOGICAL_ORDER seçeneğini kullanırsanız, uygulamaya MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

MQC.MQGMO_MARK_SKIP_BACKOUT*

Kuyruktaki iletiyi yeniden belirtmeden bir iş biriminin geriletilmesine izin verin.

MQC.MQGMO_MSG_UNDER_CURSOR

İletiyi göz atma imlecinin altına alın.

MQC.MQGMO_NONE

Başka seçenek belirtilmedi; tüm seçenekler varsayılan değerlerini varsayar.

MQC.MQGMO_NO_PROPERTIES

İleti tanımlayıcısında (ya da uzantıda) bulunan özellikler dışında, iletinin özellikleri alınmaz.

MQC.MQGMO_NO_SYNCPOINT

Eşitleme noktası denetimi olmadan ileti alın.

MQC.MQGMO_NO_WAIT

Uygun bir ileti yoksa hemen döndür.

MQC.MQGMO_PROPERTIES_AS_Q_DEF

MQQueueöğesinin PropertyControl özneliğiyle tanımlandığı şekilde ileti özelliklerini alın. İleti tanımlayıcısındaki ya da uzantıdaki ileti özelliklerine erişim, PropertyControl özneliğinden etkilenmez.

MQC.MQGMO_PROPERTIES_COMPATIBILITY

MQRFH2 üstbilgilerinde mcd, jms, usrya da mqextönekine sahip ileti özelliklerini alın. İleti tanımlayıcısında ya da uzantıda bulunan özellikler dışında, iletinin diğer özellikleri atılır.

MQC.MQGMO_PROPERTIES_FORCE_MQRFH2

MQRFH2 üstbilgilerinde, ileti tanımlayıcısında ya da uzantıda bulunan özellikler dışında ileti özelliklerini alın. Özellikleri almayı bekleyen, ancak ileti tanıtıcılarını kullanacak şekilde değiştirilemeyen uygulamalarda MQC.MQGMO_PROPERTIES_FORCE_MQRFH2 kullanın.

MQC.MQGMO_PROPERTIES_IN_HANDLE

MsgHandlekullanarak ileti özelliklerini alın.

MQC.MQGMO_SYNCPOINT

İletiyi eşitleme noktası denetimi altında alın. İleti, diğer uygulamalar tarafından kullanılamıyor olarak işaretlenir, ancak yalnızca iş birimi kesinleştirildiğinde kuyruktan silinir. İş birimi geriletildiyse, ileti yeniden kullanılabilir kılır.

MQC.MQGMO_SYNCPOINT_IF_PERSISTENT*

İleti kalıcıysa, eşitleme noktası denetimine sahip iletiyi alın.

MQC.MQGMO_UNLOCK*

Önceden kilitlemiş bir iletinin kilidini açın.

MQC.MQGMO_WAIT

Bir mesajın gelmesini bekleyin.

public string ResolvedQueueName {get;}

Kuyruk yöneticisi ResolvedQueueName (Kuyruk adını çözümle) ögesini, iletinin alındığı kuyruğun yerel adına ayarlar. ResolvedQueueName (Çözüm Kuyruğu Adı), bir diğer ad kuyruğu ya da model kuyruğu açıldığında kuyruğu açmak için kullanılan addan farklıdır.

public char Segmentation {get;}*

Bölümleme , alınan ileti için bölümlemeye izin verilip verilemeyeceğini gösterir. Olası değerler şunlardır:

MQC.MQSEG_INHIBITED

Bölümlemeye izin verme.

MQC.MQSEG_ALLOWED

Bölümlemeye izin ver

public byte SegmentStatus {get;}*

SegmentStatus , alınan iletinin bir mantıksal iletinin parçası olup olmadığını gösteren bir çıkış alanıdır. İleti bir kesim ise, işaret bunun son kesim olup olmadığını gösterir. Olası değerler şunlardır:

MQC.MQSS_LAST_SEGMENT

İleti, mantıksal iletinin son ya da tek bölümüdür.

MQC.MQSS_NOT_A_SEGMENT

İleti bir bölüm değil.

MQC.MQSS_SEGMENT

İleti bir bölüm, ancak mantıksal iletinin son bölümü değil.

public int WaitInterval {get; set;}

WaitInterval , bir MQQueue.get çağrısına ilişkin milisaniye cinsinden uygun bir iletinin gelmesini bekleme süresi üst sınırıdır. MQC.MQGMO_WAIT ile WaitInterval komutunu kullanın. Bir ileti için sınırsız süre beklemek üzere MQC.MQWI_UNLIMITED değerini ayarlayın.

Oluşturucular

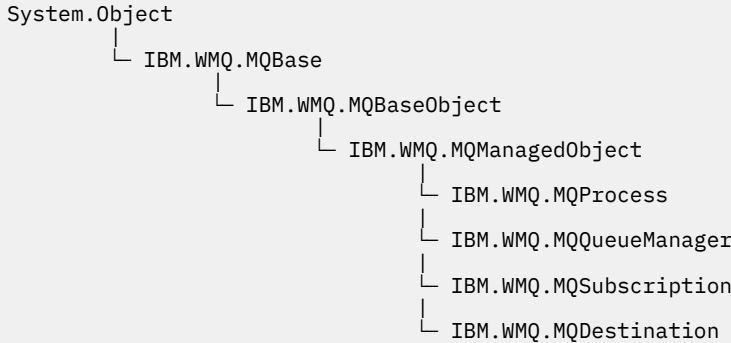
public MQGetMessageOptions()

Seçenekler MQC.MQGMO_NO_WAIT, WaitInterval sıfıra ayarlanmış şekilde yeni bir MQGetMessageOptions nesnesi oluşturun ve ResolvedQueueName boş olarak ayarlandı.

MQManagedObject.NET sınıfı

MQDestination, MQProcess, MQQueueManager ve MQSubscription özelliklerini sorgulamak ve ayarlamak için MQManagedObject kullanın. MQManagedObject , bu sınıfların üst sınıfıdır.

Sınıflar



```
public class IBM.WMQ.MQManagedObject extends IBM.WMQ.MQBaseObject;
```

- [“Özellikler” sayfa 1705](#)
- [“yöntemler” sayfa 1706](#)
- [“Oluşturucular” sayfa 1707](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

public string AlternateUserId {get; set;}

Diğer kullanıcı kimliği (varsa), kaynağın açıldığı zamanı belirler. AlternateUserID.set , açılan bir nesne için yayınlandığında yoksayılar. AlternateUserId abonelikler için geçerli değil.

public int CloseOptions {get; set;}

Kaynağın nasıl kapatıldığını denetlemek için bu özniteliği ayarlayın. Varsayılan değer MQC.MQCO_NONE'dedir. MQC.MQCO_NONE , kalıcı dinamik kuyruklar, geçici dinamik kuyruklar, abonelikler ve bunları yaratan nesnelere tarafından erişilmekte olan konular dışındaki tüm kaynaklar için izin verilen tek değerdir.

Kuyruklar ve konular için aşağıdaki ek değerlere izin verilebilir:

MQC.MQCO_DELETE

İleti yoksa kuyruğu silin.

MQC.MQCO_DELETE_PURGE

Kuyruktaki iletileri temizleyip kuyruğu silin.

MQC.MQCO_QUIESCE

Herhangi bir ileti kalırsa (son kapanıştan önce alınmalarına izin veren), kuyruğun kapatılmasını isteyin.

Abonelikler için aşağıdaki ek değerlere izin verilebilir:

MQC.MQCO_KEEP_SUB

Abonelik silinmez. Bu seçenek, özgün abonelik sürekli ise geçerlidir. MQC.MQCO_KEEP_SUB , kalıcı bir konuya ilişkin varsayılan değerdir.

MQC.MQCO_REMOVE_SUB

Abonelik silindi. MQC.MQCO_REMOVE_SUB , dayanıklı olmayan, yönetilmeyen bir konu için varsayılan değerdir.

MQC.MQCO_PURGE_SUB

Abonelik silindi. MQC.MQCO_PURGE_SUB , sürekli olmayan, yönetilen bir konunun varsayılan değeridir.

public MQQueueManager ConnectionReference {get;}

Bu kaynağın ait olduğu kuyruk yöneticisi.

public string MQDescription {get;}

Kuyruk yöneticisi tarafından tutulan kaynağın tanımlaması. MQDescription , abonelikler ve konular için boş bir dizgi döndürür.

public boolean IsOpen {get;}

Kaynağın açık olup olmadığını belirtir.

public string Name {get;}

Kaynağın adı. Ad, erişim yönteminde sağlanan ya da kuyruk yöneticisi tarafından dinamik bir kuyruk için ayrılan addır.

public int OpenOptions {get; set;}

OpenOptions , bir IBM MQ nesnesi açıldığında ayarlanır. OpenOptions.set yöntemi yoksayılr ve bir hataya neden olmaz. Aboneliklerin OpenOptionsseçeneği yoktur.

yöntemler

public virtual void Close();

MQExceptionögesini atar.

Nesneyi kapatır. Closeçağırısından sonra bu kaynakla ilgili başka işlem yapılmasına izin verilmez. Close yönteminin davranışını değiştirmek için closeOptions özniteliğini ayarlayın.

public string GetAttributeString(int selector, int length);

MQExceptionögesini atar.

Bir öznitelik dizgisi alır.

seçici

Sorgulanmakta olan özniteliği gösteren tamsayı.

uzunluk

Gerekli dizginin uzunluğunu gösteren tamsayı.

public void Inquire(int[] selectors, int[] intAttrs, byte[] charAttrs);

MQExceptionögesini atar.

Bir kuyruk, işlem ya da kuyruk yöneticisinin özniteliklerini içeren bir dizi tamsayı ve karakter dizgisi döndürür. Sorgulanacak öznitelikler seçiciler dizisinde belirtilir.

Not: Daha sık kullanılan özniteliklerin çoğu, MQManagedObject, MQQueue ve MQQueueManager içinde tanımlanan Get yöntemleri kullanılarak sorgulanabilir.

seçiciler

Sorulacak değerleri olan öznitelikleri tanımlayan tamsayı dizisi.

intAttrs

Tamsayı öznitelik değerlerinin döndürüldüğü dizi. Tamsayı öznitelik değerleri, seçiciler dizisindeki tamsayı öznitelik seçicileri ile aynı sırada döndürülür.

charAttrs

Karakter özniteliklerinin döndürüldüğü, bitleştirilmiş arabellek. Karakter öznitelikleri, seçiciler dizisindeki karakter öznitelik seçicileri ile aynı sırada döndürülür. Her öznitelik dizgisinin uzunluğu, her öznitelik için sabittir.

public void Set(int[] selectors, int[] intAttrs, byte[] charAttrs);

MQExceptionögesini atar.

Seçicilerin vektöründe tanımlanan öznitelikleri ayarlar. Ayarlanacak öznitelikler seçiciler dizisinde belirtilir.

seçiciler

Ayarlanacak değerlerle öznitelikleri tanımlayan tamsayı dizisi.

intAttrs

Ayarlanacak tamsayı öznitelik değerleri dizisi. Bu değerler, seçiciler dizisindeki tamsayı öznitelik seçicileri ile aynı sırada olmalıdır.

charAttrs

Karakter özniteliklerinin ayarlanacağı arabellek bitleştirilir. Bu değerler, seçiciler dizisindeki karakter öznitelik seçicileri ile aynı sırada olmalıdır. Her karakter özniteliğinin uzunluğu düzeltilmiştir.

public void SetAttributeString(int selector, string value, int length);

MQExceptionögesini atar.

Bir öznitelik dizgisi ayarlar.

seçici

Hangi özniteliğin ayarlandığını gösteren tamsayı.

değer

Öznitelik değeri olarak ayarlanacak dizgi.

uzunluk

Gerekli dizginin uzunluğunu gösteren tamsayı.

Oluşturucular

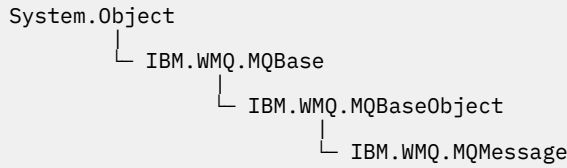
protected MQManagedObject()

Oluşturucu yöntemi. Bu nesne, kendisi tarafından somutlaştırılmayan soyut bir temel sınıftır.

MQMessage.NET sınıfı

Bir IBM MQ iletisine ilişkin ileti tanımlayıcısına ve verilere erişmek için MQMessage komutunu kullanın. MQMessage bir IBM MQ iletisini kapsüller.

Sınıf



```
public class IBM.WMQ.MQMessage extends IBM.WMQ.MQBaseObject;
```

Bir MQMessage nesnesi oluşturun ve daha sonra, ileti ile uygulamanızdaki diğer nesnelere arasında veri aktarmak için Read ve Write yöntemlerini kullanın. MQMessage nesnelerini MQDestination, MQQueue ve MQTopic sınıflarının Put ve Get yöntemlerini kullanarak gönderin ve alın.

MQMessage özelliklerini kullanarak ileti tanımlayıcısının özelliklerini alın ve ayarlayın. SetProperty ve GetProperty yöntemlerini kullanarak genişletilmiş ileti özelliklerini ayarlayın ve alın.

- [“Özellikler” sayfa 1708](#)
- [“Read ve Write ileti yöntemleri” sayfa 1713](#)
- [“Arabellek yöntemleri” sayfa 1716](#)
- [“Özellik yöntemleri” sayfa 1717](#)
- [“Oluşturucular” sayfa 1719](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

```
public string AccountingToken {get; set;}
```

İletinin kimlik bağlamının bir parçası; bir uygulamanın ileti sonucu yapılan iş için ücret almasına yardımcı olur. Varsayılan değer MQC.MQACT_NONE'dir.

```
public string ApplicationIdData {get; set;}
```

İletinin kimlik bağlamının bir parçası. ApplicationIdData, uygulama grubu tarafından tanımlanan bilgilerdir ve ileti ya da oluşturucusu hakkında ek bilgi sağlamak için kullanılabilir. Varsayılan değer ""'dır.

```
public string ApplicationOriginData {get; set;}
```

İletinin kökeni hakkında ek bilgi sağlamak için kullanılabilir, uygulama tarafından tanımlanan bilgiler. Varsayılan değer ""'dır.

```
public int BackoutCount {get;}
```

Bir iş biriminin parçası olarak bir MQQueue.Get çağrısı tarafından iletinin kaç kez geri döndürüldüğünü ve kaç kez geri çekildiğini belirten sayı. Varsayılan değer sıfırdır.

```
public int CharacterSet {get; set;}
```

İletideki karakter verilerinin kodlanmış karakter takımı tanıtcısı.

İletideki karakter verilerinin karakter kümesini tanımlamak için CharSet ayarını tanımlayın. İletideki karakter verilerini kodlamak için hangi karakter kümesinin kullanıldığını öğrenmek için CharSet öğesini alın.

.NET uygulamaları her zaman Unicode 'da çalışırken, diğer ortamlarda uygulamalar kuyruk yöneticisiyle aynı karakter kümesinde çalışır.

ReadString ve ReadLine yöntemleri, iletideki karakter verilerini sizin için Unicode 'a dönüştürür.

WriteString yöntemi, CharSet içinde kodlanmış karakter kümesine dönüştürür. CharSet varsayılan değerine (MQC.MQCCSI_Q_MGR) ayarlanırsa, dönüştürme gerçekleşmez ve CharSet 1200 olarak ayarlanır. CharSet ayarını başka bir değere ayarlarsanız, WriteString Unicode 'dan diğer değere dönüştürür.

Not: Diğer okuma ve yazma yöntemleri CharSet özelliğini kullanmaz.

- ReadChar ve WriteChar , dönüştürmeden ileti arabelleğine/arabelleğine bir Unicode karakteri okuma ve yazma.
- ReadUTF ve WriteUTF , uygulamadaki bir Unicode dizgisi ile öneki 2 baytlık bir alan olan UTF-8 dizgisi arasında ileti arabelleğinde dönüştürme.
- Bayt yöntemleri, uygulama ile ileti arabelleği arasında değişiklik yapmadan bayt aktarımı sağlar.

public byte[] CorrelationId {get; set;}

- Bir MQQueue . Get çağrısı için, alınacak iletinin ilinti tanıtıcısı. Kuyruk yöneticisi, ileti tanımlayıcı alanlarıyla eşleşen bir ileti tanıtıcısı ve ilinti tanıtıcısı içeren ilk iletiyi döndürür. Varsayılan değer olan MQC.MQCI_NONE, herhangi bir ilinti tanıtıcısının eşleşmesine yardımcı olur.
- Bir MQQueue . Put çağrısı için ayarlanacak ilinti tanıtıcısı.

public int DataLength {get;}

Okunacak ileti verilerinin kalan bayt sayısı.

public int DataOffset {get; set;}

İleti verileri içindeki geçerli imleç konumu. Okuma ve yazma işlemleri geçerli konumda etkili olur.

public int Encoding {get; set;}

Uygulama ileti verilerinde sayısal değerler için kullanılan gösterim. Kodlama , ikili, paketlenmiş onlu ve kayan noktalı veriler için geçerlidir. Bu sayısal biçimler için okuma ve yazma yöntemlerinin davranışı buna göre değiştirilir. Bu üç bölümün her birinden bir değer ekleyerek kodlama alanı için bir değer oluşturun. Alternatif olarak, bit bakımından OR işlecini kullanarak üç bölümün her birinden değerleri birleştiren değeri oluşturun.

1. İkili tamsayı

MQC.MQENC_INTEGER_NORMAL

Big-endian tamsayıları.

MQC.MQENC_INTEGER_REVERSED

Intel mimarisinde kullanılan Little-endian tamsayıları.

2. Paketli onlu

MQC.MQENC_DECIMAL_NORMAL

z/OS tarafından kullanılan Big-endian paketli onlu.

MQC.MQENC_DECIMAL_REVERSED

Little-endian paketli onlu.

3. Kayan nokta

MQC.MQENC_FLOAT_IEEE_NORMAL

Big-endian IEEE yüzüyor.

MQC.MQENC_FLOAT_IEEE_REVERSED

Kullanılan Intel mimarisi gibi Little-endian IEEE yüzer.

MQC.MQENC_FLOAT_S390

z/OS biçim kayan noktaları.

Varsayılan değer:

```
MQC.MQENC_INTEGER_REVERSED |
MQC.MQENC_DECIMAL_REVERSED |
MQC.MQENC_FLOAT_IEEE_REVERSED
```

Varsayılan ayar, WriteInt 'in little-endian tamsayı yazmasına ve ReadInt ' un little-endian tamsayı okumasına neden olur. Bunun yerine MQC.MQENC_INTEGER_NORMAL işaretini ayarlarsanız, WriteInt büyük sıralı bir tamsayı yazar ve ReadInt büyük sıralı değer (big-endian) tamsayısını okur.

Not: IEEE biçim kayan noktalarından zSeries biçim kayan noktalarına dönüştürülürken duyarlık kaybı oluşabilir.

public int Expiry {get; set;}

İletiyi koyan uygulama tarafından ayarlanan, saniyenin onda biri cinsinden ifade edilen süre bitimi. Bir iletinin süre bitim tarihi geçtikten sonra, kuyruk yöneticisi tarafından atılabilir. İleti MQC.MQRO_EXPIRATION işaretlerinden birini belirtmişse, ileti atıldığında bir rapor oluşturulur. Varsayılan değer, iletinin süresinin hiçbir zaman sona ermeyeceğini belirten MQC.MQEI_UNLIMITEDdeğeridir.

public int Feedback {get; set;}

Raporun niteliğini belirtmek için MQC.MQMT_REPORT tipinde bir iletiyle Geribildirim 'i kullanın. Aşağıdaki geribildirim kodları sistem tarafından tanımlanır:

- 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

MQC.MQFB_APPL_FIRST - MQC.MQFB_APPL_LAST aralığındaki uygulama tanımlı geribildirim değerleri de kullanılabilir. Bu alanın varsayılan değeri, geribildirim sağlanmadığını gösteren MQC.MQFB_NONEdeğeridir.

public string Format {get; set;}

İletideki verilerin türünü alıcıya göstermek için iletinin göndereni tarafından kullanılan bir biçim adı. Kendi biçim adlarınızı kullanabilirsiniz, ancak MQ harfleriyle başlayan adların kuyruk yöneticisi tarafından tanımlanan anlamları vardır. Kuyruk yöneticisi yerleşik biçimleri şunlardır:

MQC.MQFMT_ADMIN

Komut sunucusu isteği/yanıt iletisi.

MQC.MQFMT_COMMAND_1

1 komut yanıt iletisi yazın.

MQC.MQFMT_COMMAND_2

2 komut yanıt iletisi yazın.

MQC.MQFMT_DEAD_LETTER_HEADER

Ölmeyen harf üstbilgisi.

MQC.MQFMT_EVENT

Olay iletisi.

MQC.MQFMT_NONE

Biçim adı yok.

MQC.MQFMT_PCF

Programlanabilir komut biçiminde kullanıcı tanımlı ileti.

MQC.MQFMT_STRING

Tamamen karakterlerden oluşan ileti.

MQC.MQFMT_TRIGGER

Tetikleyici ileti

MQC.MQFMT_XMIT_Q_HEADER

İletim kuyruğu üstbilgisi.

Varsayılan değer MQC . MQFMT_NONEdeğeridir.

public byte[] GroupId {get; set;}

Fiziksel iletinin ait olduğu ileti grubunu tanıtan bayt dizilimi. Varsayılan değer MQC . MQGI_NONEdeğeridir.

public int MessageFlags {get; set;}

Bir iletinin bölümlenmesini ve durumunu denetleyen işaretler.

public byte[] MessageId {get; set;}

MQQueue . Get çağrısı için bu alan, alınacak iletinin ileti tanıtıcısını belirtir. Olağan durumda, kuyruk yöneticisi ileti tanımlayıcı alanlarıyla eşleşen bir ileti tanıtıcısı ve ilinti tanıtıcısı içeren ilk iletiyi döndürür. Herhangi bir ileti tanıtıcısının MQC . MQMI_NONEözel değerini kullanarak eşleşmesine izin verin.

MQQueue . Put çağrısı için bu alan, kullanılacak ileti tanıtıcısını belirtir. MQC . MQMI_NONE belirtilirse, kuyruk yöneticisi ileti konduğunda benzersiz bir ileti tanıtıcısı oluşturur. Bu üye değişkeninin değeri, kullanılan ileti tanıtıcısını belirtmek için, koyma işleminden sonra güncellenir. Varsayılan değer MQC . MQMI_NONEdeğeridir.

public int MessageLength {get;}

MQMessage nesnesindeki ileti verilerinin bayt sayısı.

public int MessageSequenceNumber {get; set;}

Bir grup içindeki mantıksal iletinin sıra numarası.

public int MessageType {get; set;}

İletinin tipini gösterir. Aşağıdaki değerler sistem tarafından tanımlanmıştır:

- MQC . MQMT_DATAGRAM
- MQC . MQMT_REPLY
- MQC . MQMT_REPORT
- MQC . MQMT_REQUEST

MQC . MQMT_APPL_FIRST - MQC . MQMT_APPL_LASTaralığında, uygulama tanımlı değerler de kullanılabilir. Bu alanın varsayılan değeri MQC . MQMT_DATAGRAMdeğeridir.

public int Offset {get; set;}

Bölümlenmiş bir iletide, mantıksal iletinin başlangıcından itibaren fiziksel bir iletideki verilerin görelî konumu.

public int OriginalLength {get; set;}

Bölümlenmiş iletinin özgün uzunluğu.

public int Persistence {get; set;}

İleti kalıcılığı. Aşağıdaki değerler tanımlanır:

- MQC . MQPER_NOT_PERSISTENT

Bu seçeneği yeniden bağlanabilir bir istemcide ayarlarsanız, bağlantı başarılı olduğunda MQRC_NONE neden kodu uygulamaya döndürülür.

- MQC . MQPER_PERSISTENT

Bu seçeneği yeniden bağlanabilir bir istemcide ayarlarsanız, bağlantı başarılı olduktan sonra uygulamaya MQRC_CALL_INTERRUPTED neden kodu döndürülür.

- MQC . MQPER_PERSISTENCE_AS_Q_DEF

Varsayılan değer, hedef kuyruğun varsayılan kalıcılık özneliğinden iletiye ilişkin kalıcılığı alan MQC . MQPER_PERSISTENCE_AS_Q_DEFdeğeridir.

public int Priority {get; set;}

İleti önceliği. MQC . MQPRI_PRIORITY_AS_Q_DEF özel değeri giden iletide de ayarlanabilir. İletiyeye ilişkin öncelik, hedef kuyruğun varsayılan öncelik özneliğinden alınır. Varsayılan değer MQC . MQPRI_PRIORITY_AS_Q_DEFdeğeridir.

public int PropertyValidation {get; set;}

İletinin bir özelliği ayarlandığında özelliklerin doğrulanmasının gerçekleşip gerçekleşmeyeceğini belirtir. Olası değerler şunlardır:

- MQCMHO_DEFAULT_VALIDATION
- MQCMHO_VALIDATE
- MQCMHO_NO_VALIDATION

Varsayılan değer MQCMHO_DEFAULT_VALIDATION'dur.

public string PutApplicationName {get; set;}

İletiyi koyan uygulamanın adı. Varsayılan değer ""'dır.

public int PutApplicationType {get; set;}

İletiyi koyan uygulamanın tipi. PutApplicationTipi, sistem tanımlı ya da kullanıcı tanımlı bir değer olabilir. Aşağıdaki değerler sistem tarafından tanımlanır:

- 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

Varsayılan değer, iletide bağlam bilgisi olmadığını gösteren MQC.MQAT_NO_CONTEXT'dur.

public DateTime PutDateTime {get; set;}

İletinin konma saati ve tarihi.

public string ReplyToQueueManagerName {get; set;}

Yanıt ya da rapor iletileri gönderilecek kuyruk yöneticisinin adı. Varsayılan değer ""'dır ve kuyruk yöneticisi ReplyToQueueManagerAdı'nı sağlar.

public string ReplyToQueueName {get; set;}

İletiyeye ilişkin alma isteğini yayınlayan uygulamanın MQC.MQMT_REPLY ve MQC.MQMT_REPORT iletilerini gönderdiği ileti kuyruğunun adı. Varsayılan ReplyToQueueName ""'dır.

public int Report {get; set;}

Rapor ve yanıt iletilerine ilişkin seçenekleri belirlemek için Rapor'u kullanın:

- Raporların gerekli olup olmadığını belirler.
- Uygulama iletilerinin raporlara dahil edilip edilmeyeceğini belirler.
- Raporla ya da yanıtta ileti ve ilinti tanıtıcılarının ayarlanması.

Dört rapor tipinin herhangi bir birleşimi istenebilir:

- Dört rapor tipinin herhangi bir birleşimini belirtin. Uygulama iletilerinin rapor iletilerine dahil edilip edilmeyeceğine bağlı olarak, her rapor tipi için üç seçenektен herhangi birini seçme.

1. Variyasyonla

- MQC.MQRO_COA
- MQC.MQRO_COA_WITH_DATA
- MQC.MQRO_COA_WITH_FULL_DATA **

2. Teslimde onayla

- MQC.MQRO_COD
- MQC.MQRO_COD_WITH_DATA
- MQC.MQRO_COD_WITH_FULL_DATA **

3. Kural dışı durum

- MQC.MQRO_EXCEPTION
- MQC.MQRO_EXCEPTION_WITH_DATA
- MQC.MQRO_EXCEPTION_WITH_FULL_DATA **

4. Süre Bitim Tarihi

- MQC.MQRO_EXPIRATION
- MQC.MQRO_EXPIRATION_WITH_DATA
- MQC.MQRO_EXPIRATION_WITH_FULL_DATA **

Not: Listede ** ile işaretlenen değerler, z/OS kuyruk yöneticileri tarafından desteklenmez. Uygulamanız, uygulamanın çalıştığı platformdan bağımsız olarak bir z/OS kuyruk yöneticisine erişebilirse bunları kullanmayın.

- Rapor ya da yanıt iletisi için ileti tanıtıcısının nasıl oluşturulduğunu denetlemek için aşağıdakilerden birini belirtin:
 - MQC.MQRO_NEW_MSG_ID
 - MQC.MQRO_PASS_MSG_ID
- Rapor ya da yanıt iletisinin ilinti tanıtıcısının nasıl ayarlanacağını denetlemek için aşağıdakilerden birini belirtin:
 - MQC.MQRO_COPY_MSG_ID_TO_CORREL_ID
 - MQC.MQRO_PASS_CORREL_ID
- Hedef kuyruğa teslim edilemediğinde özgün iletinin yok edilmesini denetlemek için aşağıdakilerden birini belirtin:
 - MQC.MQRO_DEAD_LETTER_Q
 - MQC.MQRO_DISCARD_MSG **
- Rapor seçeneği belirtilmezse, varsayılan değer:

```
MQC.MQRO_NEW_MSG_ID |
MQC.MQRO_COPY_MSG_ID_TO_CORREL_ID |
MQC.MQRO_DEAD_LETTER_Q
```

- Alan uygulamanın pozitif bir eylem ya da negatif eylem raporu iletisi göndermesini istemek için aşağıdakilerden birini ya da her ikisini belirtebilirsiniz.
 - MQC.MQRO_PAN
 - MQC.MQRO_NAN

public int TotalMessageLength {get;}

Bu iletinin alındığı ileti kuyruğunda saklanan toplam bayt sayısı.

public string UserId {get; set;}

UserId, iletinin kimlik bağlamının bir parçasıdır. Kuyruk yöneticisi genellikle değeri sağlar. Kimlik bağlamını ayarlama yetkiniz varsa, değeri geçersiz kılabilirsiniz.

public int Version {get; set;}

Kullanılmakta olan MQMD yapısının sürümü.

Read ve Write ileti yöntemleri

Read ve Write yöntemleri, .NET System.IO ad alanındaki BinaryReader ve BinaryWriter sınıflarının üyeleriyle aynı işlevleri gerçekleştirir. Tam dil sözdizimi ve kullanım örnekleri için MSDN ' ye

bakın. İleti arabelleğindeki yürürlükteki konumdan okuma ya da yazma yöntemleri. Yürürlükteki konumu okunan ya da yazılan bayt sayısına göre ileri taşır.

Not: İleti verileri bir MQRFH ya da MQRFH2 üstbilgisi içeriyorsa, verileri okumak için ReadBytes yöntemini kullanmanız gerekir.

- Tüm yöntemler IOException' i atar.
- ReadFully yöntemleri, hedef byte ya da sbyte dizisini iletiye tam olarak uyacak şekilde otomatik olarak yeniden boyutlandırır. Boş bir dizi de yeniden boyutlandırılır.
- Read yöntemleri EndOfStreamException' dir.
- WriteDecimal yöntemleri MQException' dir.
- ReadString, ReadLine ve WriteString yöntemleri, Unicode ve iletinin karakter kümesi arasında dönüştürülür; bkz. [CharacterSet](#).
- Decimal yöntemleri, Kodlamadeğerine göre big-endian, MQC.MQENC_DECIMAL_NORMAL ya da little-endian MQC.MQENC_DECIMAL_REVERSE biçiminde kodlanmış paketlenmiş ondalık sayıları okur ve yazar. Ondalık aralıkları ve ilgili .NET tipleri aşağıdaki gibidir:

Decimal2/short

-999-999

Decimal4/int

-9999999-9999999

Decimal8/long

-999999999999999-99999999999

- Double ve Float yöntemleri, Kodlamadeğerine göre IEE big-endian ve little-endian biçimlerinde (MQC.MQENC_FLOAT_IEEE_NORMAL ve MQC.MQENC_FLOAT_IEEE_REVERSED) ya da S/390 biçiminde (MQC.MQENC_FLOAT_S390) kodlanmış kayan değerleri okur ve yazar.
- Int yöntemleri, Kodlamadeğerine göre big-endian, MQC.MQENC_INTEGER_NORMAL ya da little-endian, MQC.MQENC_INTEGER_REVERSED biçiminde kodlanmış tamsayı değerlerini okur ve yazar. İşaretsiz 2 baytlık tamsayı tipinin eklenmesi dışında, tamsayıların tümü işaretli olur. Tamsayı boyutları ve .NET ve IBM MQ tipleri aşağıdaki gibidir:

2 bayt

short, Int2, ushort, UInt2

4 bayt

int, Int4

8 bayt

long, Int8

- WriteObject , bir nesnenin sınıfını, geçici olmayan ve durağan olmayan alanlarının değerlerini ve üst tiplerindeki alanlarını ileti arabelleğine aktarır.
- ReadObject , nesnenin sınıfından, sınıfın imzasından, geçici olmayan ve durağan olmayan alanlarının değerlerinden ve üst tiplerindeki alanlardan bir nesne yaratır.

Çizelge 843. İleti okuma ve yazma yöntemleri	
Hedef türü	Yöntem imzaları
Boolean	<pre>public bool ReadBoolean(); public void WriteBoolean(bool value);</pre>

Çizelge 843. İleti okuma ve yazma yöntemleri (devamı var)

Hedef türü	Yöntem imzaları
Byte	<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>
Bytes	<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>
Decimal2	<pre>public void WriteDecimal2(short value)</pre>
Decimal4	<pre>public void WriteDecimal4(short value)</pre>
Decimal8	<pre>public void WriteDecimal8(short value)</pre>
Double	<pre>public double ReadDouble() public void WriteDouble(double value)</pre>
Float	<pre>public float ReadFloat() public void WriteFloat(float value)</pre>
Int2	<pre>public void WriteInt2(int value)</pre>
Int4	<pre>public int readDecimal4() public int ReadInt() public int ReadInt4() public void WriteInt(int value) public void WriteInt4(int value)</pre>

Çizelge 843. İleti okuma ve yazma yöntemleri (devamı var)

Hedef türü	Yöntem imzaları
Int8	<pre>public void WriteInt8(long value)</pre>
Long	<pre>public long ReadDecimal8() public long ReadLong() public long ReadInt8() public void WriteLong(long value)</pre>
Object	<pre>public Object ReadObject() public void WriteObject(Object object)</pre>
Short	<pre>public short ReadShort() public short ReadDecimal2() public short ReadInt2() public void WriteShort(int value)</pre>
string	<pre>public string ReadString(int length) public void WriteString(string string)</pre>
Unsigned Short	<pre>public ushort ReadUnsignedShort() public ushort ReadUInt2()</pre>
Unicode	<pre>public string ReadLine() public char ReadChar() public void WriteChar(int value) public void WriteChars(string string)</pre>
UTF	<pre>public string ReadUTF() public void WriteUTF(string string)</pre>

Arabellek yöntemleri

```
public void ClearMessage();
```

IOExceptionögesini atar.

İleti arabelleğindeki verileri atar ve veri görelî konumunu sıfıra geri ayarlar.

public void ResizeBuffer(int size)

IOExceptionöğesini atar.

MQMessage nesnesine, sonraki alma işlemleri için gerekli olabilecek arabellek boyutuyla ilgili bir ipucu. İleti şu anda ileti verileri içeriyorsa ve yeni boyut geçerli boyuttan azsa, ileti verileri kesilir.

public void Seek(int pos)

IOException, ArgumentOutOfRangeException, ArgumentExceptionöğesini atar.

İmleci, postarafından verilen ileti arabelleğinde mutlak konuma taşır. Sonraki okuma ve yazma işlemleri, arabellekte bu konumda işlem yapar.

public int SkipBytes(int i)

IOException, EndOfStreamExceptionöğesini atar.

İleti arabelleğinde n byte 'ı leri taşır ve atlanan byte sayısı olan ndeğerini döndürür.

SkipBytes yöntemi, aşağıdaki olaylardan biri gerçekleşinceye kadar öbekler:

- Tüm baytlar atlandı
- İleti arabelleğinin sonu saptandı
- Bir kural dışı durum yayınlandı

Özellik yöntemleri

public void DeleteProperty(string name);

MQExceptionöğesini atar.

Belirtilen ada sahip bir özelliği iletiden siler.

Ad

Silinecek özelliğın adı.

public System.Collections.IEnumerator GetPropertyNames(string name)

MQExceptionöğesini atar.

Belirtilen adla eşleşen tüm özellik adlarından bir IEnumerator döndürür. Adın sonunda '%' yüzde işareti, iletinin özelliklerini süzmek için genel arama karakteri olarak kullanılabilir; nokta imi de içinde olmak üzere, sıfır ya da daha çok karakterle eşleşir.

Ad

Eşleştirilecek özelliğın adı.

SetProperty ve GetProperty yöntemleri

Tüm SetProperty ve GetProperty yöntemleri MQExceptionöğesini atar.

Bir özellik yoksa, MQMessage .NET sınıfının SetProperty yöntemi yeni bir özellik ekler. Ancak, özellik zaten varsa, sağlanan özellik değeri listenin sonuna eklenir. SetPropertykullanılarak birden çok değeri bir özellik adına ayarlandığında, bu ad için GetProperty çağrıldığında bu değeri, bu değeri ayarlandığı sırayla döndürülür.

Davranış, GetLongProperty, SetLongProperty, GetBooleanProperty, SetBooleanProperty, GetStringPropertyve SetStringPropertygibi Set*Property ve Get*Property tipinde tüm yöntemler için aynıdır.

Çizelge 844. SetProperty ve GetProperty yöntemleri

Tip	Yöntem imzaları
Boolean	<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>
Byte	<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>
Bytes	<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>
Double	<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>
Float	<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>
Int2	<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>
Int4	<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>

Çizelge 844. SetProperty ve GetProperty yöntemleri (devamı var)

Tip	Yöntem imzaları
Int8	<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>
Long	<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>
Object	<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>
Short	<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>
string	<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>

Oluşturuçular

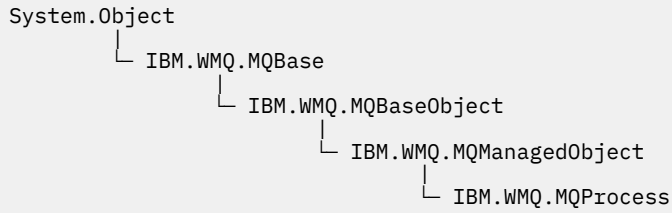
public MQMessage();

Varsayılan ileti tanımlayıcı bilgileri ve boş bir ileti arabelleğine sahip bir MQMessage nesnesi yaratır.

MQProcess.NET sınıfı

Bir IBM MQ sürecinin özniteliklerini sorgulamak için MQProcess kullanın. Bir oluşturuçuyu ya da MQQueueManager AccessProcess yöntemi kullanarak bir MQProcess nesnesi yaratın.

Sınıf



```
public class IBM.WMQ.MQProcess extends IBM.WMQ.MQManagedObject;
```

- [“Özellikler” sayfa 1720](#)
- [“Oluşturucular” sayfa 1721](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

public string ApplicationId {get;}

Başlatılacak uygulamayı tanıtan karakter dizilimini alır. ApplicationId , bir tetikleyici izleyicisi uygulaması tarafından kullanılır. ApplicationId , tetikleyici iletisinin bir parçası olarak başlatma kuyruğuna gönderilir.

Varsayılan değer boş değerdir.

public int ApplicationType {get;}

Tetikleyici izleme uygulaması tarafından başlatılacak işlemin tipini tanımlar. Standart tipler tanımlanır, ancak diğerleri kullanılabilir:

- MQAT_AIX
- MQAT_CICS
- MQAT_IMS
- MQAT_MVS
- MQAT_NATIVE
- MQAT_OS400
- MQAT_UNIX
- MQAT_WINDOWS
- MQAT_JAVA
- MQAT_USER_FIRST
- MQAT_USER_LAST

Varsayılan değer MQAT_NATIVEdeğeridir.

public string EnvironmentData {get;}

Başlatılacak uygulamanın ortamıyla ilgili bilgileri alır.

Varsayılan değer boş değerdir.

public string UserData {get;}

Kullanıcının başlatılacak uygulama hakkında sağladığı bilgileri alır.

Varsayılan değer boş değerdir.

Olusturucular

```
public MQProcess(MQQueueManager queueManager, string processName, int
openOptions);
public MQProcess(MQQueueManager qMgr, string processName, int openOptions,
string queueManagerName, string alternateUserId);
```

MQException ögesini atar.

Süreç özniteliklerini sorgulamak için kuyruk yöneticisindeki bir IBM MQ işleme *qMgr* erişin.

qMgr

Erişim için kuyruk yöneticisi.

processName

Açılacak sürecin adı.

openOptions

Sürecin açılmasını denetleyen seçenekler. Bitsel OR kullanılarak eklenebilen ya da birleştirilebilen geçerli seçenekler şunlardır:

- MQC.MQ00_FAIL_IF QUIESCING
- MQC.MQ00_INQUIRE
- MQC.MQ00_SET
- MQC.MQ00_ALTERNATE_USER_AUTHORITY

queueManagerAdı

İşlemin tanımlandığı kuyruk yöneticisinin adı. Kuyruk yöneticisi, işlemin eriştiği adla aynıysa, boş ya da boş bir kuyruk yöneticisi adı bırakabilirsiniz.

alternateUserKimliği

openOptions değiştirilmesinde MQC.MQ00_ALTERNATE_USER_AUTHORITY belirtilirse, *alternateUserId* işleme ilişkin yetkilendirmeyi denetlemek için kullanılan diğer kullanıcı kimliğini belirtir. MQ00_ALTERNATE_USER_AUTHORITY belirtilmezse, *alternateUserId* boş değerli ya da boş olabilir.

MQC.MQ00_ALTERNATE_USER_AUTHORITY belirtilmezse, kuyruk yöneticisine bağlanmak için varsayılan kullanıcı yetkisi kullanılır.

```
public MQProcess MQQueueManager.AccessProcess(string processName, int
openOptions);
public MQProcess MQQueueManager.AccessProcess(string processName, int
openOptions, string queueManagerName, string alternateUserId);
```

MQException ögesini atar.

Süreç özniteliklerini sorgulamak için bu kuyruk yöneticisindeki bir IBM MQ işleme erişin.

processName

Açılacak sürecin adı.

openOptions

Sürecin açılmasını denetleyen seçenekler. Bitsel OR kullanılarak eklenebilen ya da birleştirilebilen geçerli seçenekler şunlardır:

- MQC.MQ00_FAIL_IF QUIESCING
- MQC.MQ00_INQUIRE
- MQC.MQ00_SET
- MQC.MQ00_ALTERNATE_USER_AUTHORITY

queueManagerAdı

İşlemin tanımlandığı kuyruk yöneticisinin adı. Kuyruk yöneticisi, işlemin eriştiği adla aynıysa, boş ya da boş bir kuyruk yöneticisi adı bırakabilirsiniz.

alternateUserKimliđi

openOptions deđiřtirgesinde MQC.MQ00_ALTERNATE_USER_AUTHORITY belirtilirse, *alternateUserId* iřleme iliřkin yetkilendirmeyi denetlemek iin kullanılan diđer kullanıcı kimliđini belirtir. MQ00_ALTERNATE_USER_AUTHORITY belirtilmezse, *alternateUserId* boş deđerli ya da boş olabilir.

MQC.MQ00_ALTERNATE_USER_AUTHORITY belirtilmezse, kuyruk yneticisine bađlanmak iin varsayılan kullanıcı yetkisi kullanılır.

MQPropertyDescriptor.NET sınıfı

MQPropertyDescriptor parametresini MQMessage GetMessage ve SetProperty yntemleri iin bir parametre olarak kullanın. MQPropertyDescriptor , bir MQMessage zelliđini aıklar.

Sınıf

```
System.Object
└─ IBM.WMQ.MQPropertyDescriptor
```

```
public class IBM.WMQ.MQPropertyDescriptor extends System.Object;
```

- “zellikler” sayfa 1722
- “Oluřturucular” sayfa 1723

zellikler

zellikler alınırken MQException testi yapılıyor.

```
public int Context {get; set;}
```

zelliđin ait olduđu ileti bađlamı. Olası deđerler řunlardır:

MQC.MQPD_NO_CONTEXT

zellik bir ileti bađlamıyla iliřkilendirilmemiř.

MQC.MQPD_USER_CONTEXT

zellik, kullanıcı bađlamıyla iliřkilendirildi.

Kullanıcı yetkiliyse, bir ileti alındıđında kullanıcı bađlamıyla iliřkili bir zellik kaydedilir. Kaydedilen bađlama gnderme yapan sonraki bir Put yntemi, zelliđi yeni iletiye geirebilir.

```
public int CopyOptions {get; set;}
```

CopyOptions , zelliđin hangi ileti tipine kopyalanabileceđini aıklar.

Bir kuyruk yneticisi, kuyruk yneticisinin yanlıř olarak algıladıđı IBM MQ tanımlı bir zelliđi ieren bir ileti aldıđında, kuyruk yneticisi CopyOptions alanının deđerini dzeltir.

Ařađıdaki seeneklerin herhangi bir birleřimi belirtilebilir. Deđerleri ekleyerek ya da bit řeklinde ORkullanarak seenekleri birleřtirin.

MQC.MQCOPY_ALL

zellik, sonraki iletilerin tm tiplerine kopyalanır.

MQC.MQCOPY_FORWARD

zellik, iletilmekte olan bir iletiye kopyalanır.

MQC.MQCOPY_PUBLISH

zellik, bir ileti yayınlanırken bir abone tarafından alınan iletiye kopyalanır.

MQC.MQCOPY_REPLY

zellik bir yanıt iletisine kopyalanır.

MQC.MQCOPY_REPORT

Özellik bir rapor iletisine kopyalanır.

MQC.MQCOPY_DEFAULT

Değer, başka bir kopyalama seçeneği belirtilmediğini belirtti. Özellik ve sonraki iletiler arasında ilişki yok. İleti tanımlayıcı özellikleri için her zaman MQC.MQCOPY_DEFAULT döndürülür.

MQC.MQCOPY_NONE

MQC.MQCOPY_DEFAULT ile aynı

public int Options { set; }

Seçenekler varsayılan olarak CMQC.MQPD_NONE değerine ayarlanır. Başka bir değer ayarlayamazsınız.

public int Support { get; set; }

IBM MQ tanımlı ileti özellikleri için gereken destek düzeyini belirtmek için Destek seçeneğini belirleyin. Diğer tüm özellikler için destek isteğe bağlıdır. Aşağıdaki değerlerin herhangi biri ya da hiçbiri belirtilemez

MQC.MQPD_SUPPORT_OPTIONAL

Özellik desteklenmese de bir kuyruk yöneticisi tarafından kabul edilir. İletinin ileti özelliklerini desteklemeyen bir kuyruk yöneticisine akması için özellik atılabilir. Bu değer, IBM MQ tanımlı olmayan özelliklere de atanır.

MQC.MQPD_SUPPORT_REQUIRED

Özellik için destek gereklidir. İletiyi, IBM MQ tanımlı özelliği desteklemeyen bir kuyruk yöneticisine koyarsanız, yöntem başarısız olur. MQC.MQCC_FAILED tamamlanma kodunu ve MQC.MQRC_UNSUPPORTED_PROPERTY neden kodunu döndürür.

MQC.MQPD_SUPPORT_REQUIRED_IF_LOCAL

İleti yerel bir kuyruğa gönderildiyse, özellik için destek gereklidir. İletiyi, IBM MQ tanımlı özelliği desteklemeyen bir kuyruk yöneticisinde yerel bir kuyruğa koyarsanız, yöntem başarısız olur. MQC.MQCC_FAILED tamamlanma kodunu ve MQC.MQRC_UNSUPPORTED_PROPERTY neden kodunu döndürür.

İletinin uzak kuyruk yöneticisine konup konmadığı denetlenmez.

Oluşturucular

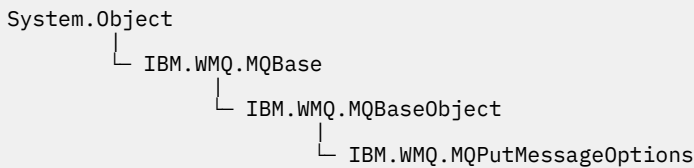
PropertyDescriptor();

Bir özellik tanımlayıcı oluşturun.

MQPutMessageOptions.NET sınıfı

İletilerin nasıl gönderileceğini belirtmek için MQPutMessageOptions seçeneğini kullanın. MQDestination.Putdavranışını değiştirir.

Sınıf



```
public class IBM.WMQ.MQPutMessageOptions extends IBM.WMQ.MQBaseObject;
```

- [“Özellikler” sayfa 1724](#) [“Oluşturucular” sayfa 1726](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

Not: Bu sınıftaki bazı seçeneklerin davranışı, bu seçeneklerin kullanıldığı ortama bağlıdır. Bu öğeler yıldız işaretiyle (*) işaretlenir.

public MQQueue ContextReference {get; set;}

options alanı MQC.MQPMO_PASS_IDENTITY_CONTEXT ya da MQC.MQPMO_PASS_ALL_CONTEXT içeriyorsa, bu alanı bağlam bilgilerinin hangi MQQueue alanından alacağına başvurmak için ayarlayın.

Bu alanın ilk değeri boş.

public int InvalidDestCount {get;} *

Genel olarak, dağıtım listeleri için kullanılan InvalidDestSayı , dağıtım listesindeki kuyruklara gönderilemeyen iletilerin sayısını gösterir. Sayı, açıklamayan kuyrukları ve başarıyla açılan, ancak koyma işleminin başarısız olduğu kuyrukları içerir.

.NET dağıtım listelerini desteklemez, ancak tek bir kuyruk açılırken InvalidDestCount ayarlanır.

public int KnownDestCount {get;} *

Genellikle dağıtım listeleri için kullanılır, KnownDestCount (Bilinen Dest Sayısı), yürürlükteki aramanın yerel kuyruklara çözülen kuyruklara başarıyla gönderdiği ileti sayısını gösterir.

.NET dağıtım listelerini desteklemez, ancak tek bir kuyruk açılırken InvalidDestCount ayarlanır.

public int Options {get; set;}

MQDestination.put ve MQQueueManager.puteylemine denetleyen seçenekler. Aşağıdaki değerlerin herhangi biri ya da hiçbiri belirtilemez. Birden çok seçenek gerekiyorsa, değerler bitisel OR işleci kullanılarak eklenebilir ya da birleştirilebilir.

MQC.MQPMO_ASYNC_RESPONSE

Bu seçenek, MQDestination.put çağrısına bazı yanıt verileriyle zamanuyumsuz olarak yapılmasına neden olur.

MQC.MQPMO_DEFAULT_CONTEXT

Varsayılan bağlamı iletiyle ilişkilendirin.

MQC.MQPMO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturulursa başarısız olur.

MQC.MQPMO_LOGICAL_ORDER *

Mantıksal iletileri ve bölümleri ileti gruplarına mantıksal sıralarına koyun.

Yeniden bağlanabilir bir istemcide MQPMO_LOGICAL_ORDER seçeneğini kullanırsanız, uygulamaya MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

MQC.MQPMO_NEW_CORREL_ID *

Gönderilen her ileti için yeni bir ilinti tanıtıcısı oluşturun.

MQC.MQPMO_NEW_MSG_ID *

Gönderilen her ileti için yeni bir ileti tanıtıcısı oluşturun.

MQC.MQPMO_NONE

Seçenek belirtilmedi. Diğer seçeneklerle birlikte kullanmayın.

MQC.MQPMO_NO_CONTEXT

İletiyle ilişkilendirilecek bir bağlam yok.

MQC.MQPMO_NO_SYNCPOINT

Eşitleme noktası denetimi olmayan bir ileti koyun. Eşitleme noktası denetimi seçeneği belirtilmezse, varsayılan bir eşitleme noktası olmadığı varsayılır.

MQC.MQPMO_PASS_ALL_CONTEXT

Giriş kuyruğu tanıtıcısından tüm bağlamı geçirin.

MQC.MQPMO_PASS_IDENTITY_CONTEXT

Bir giriş kuyruğu tanıtıcısından kimlik bağlamını geçirin.

MQC.MQPMO_RESPONSE_AS_Q_DEF

Bir MQDestination.put çağrısı için bu seçenek, kuyruğun DEFPRESP özneliğinden koyma yanıtı tipini alır.

MQQueueManager.put çağrısı için bu seçenek, aramanın zamanuyumlu olarak yapılmasına neden olur.

MQC.MQPMO_RESPONSE_AS_TOPIC_DEF

MQC.MQPMO_RESPONSE_AS_TOPIC_DEF , konu nesneleriyle kullanılmak üzere

MQC.MQPMO_RESPONSE_AS_Q_DEF eşanlamlıdır.

MQC.MQPMO_RETAIN

Gönderilmekte olan yayın, kuyruk yöneticisi tarafından alıkonacak. Bu seçenek kullanılırsa ve yayın alıkonamazsa, ileti yayınlanmaz ve çağrı MQC.MQRC_PUT_NOT_RETAINED ile başarısız olur.

MQSubscription.RequestPublicationUpdate yöntemini çağırarak, yayımlandıktan sonra bu yayının bir kopyasını isteyin. Kaydedilen yayın, MQC.MQSO_NEW_PUBLICATIONS_ONLY seçeneği ayarlanmadan abonelik oluşturan uygulamalara gönderilir. Alıkonan yayın olup olmadığını öğrenmek için, bir yayının MQIsRetained ileti özelliğini denetleyin.

Alıkonan yayınlar bir abone tarafından istendiğinde, kullanılan abonelik konu dizisinde genel arama karakteri içerebilir. Konu ağacında abonelik eşleşen birden çok alıkonan yayın varsa, bunların tümü gönderilir.

MQC.MQPMO_SET_ALL_CONTEXT

Uygulamadan tüm bağlamı ayarlayın.

MQC.MQPMO_SET_IDENTITY_CONTEXT

Uygulamadan kimlik bağlamını ayarlayın.

MQC.MQPMO_SYNC_RESPONSE

Bu seçenek, MQDestination.put ya da MQQueueManager.put aramasının tam yanıt verileriyle zamanuyumlu olarak yapılmasına neden olur.

MQC.MQPMO_SUPPRESS_REPLYTO

Yayının ReplyToQueueName ve ReplyToQueueManagerName alanlarına doldurulan bilgiler abonelere iletilmez. Bu seçenek, ReplyToQueueName gerektiren bir rapor seçeneğiyle birlikte kullanılırsa, çağrı MQC.MQRC_MISSING_REPLY_TO_Q ile başarısız olur.

MQC.MQPMO_SYNCPOINT

Eşitleme noktası denetimini içeren bir ileti koyun. İleti, iş birimi kesinleştirilinceye kadar iş biriminin dışında görünmez. İş birimi geriletildiyse, ileti silinir.

public int RecordFields {get; set;} *

Dağıtım listeleriyle ilgili bilgi. Dağıtım listeleri .NET içinde desteklenmiyordur.

public string ResolvedQueueManagerName {get;}

Kuyruk yöneticisi tarafından, uzak kuyruk adı tarafından belirlenen kuyruğun iyesi olan kuyruk yöneticisinin adına ayarlanan bir çıkış alanı. ResolvedQueueManagerName , kuyruk uzak bir kuyruksa, kuyruğa erişilen kuyruk yöneticisinin adından farklı olabilir.

Boş olmayan bir değer, yalnızca nesne tek bir kuyruksa döndürülür. Nesne bir dağıtım listesi ya da konuyla, döndürülen değer tanımsız olur.

public string ResolvedQueueName {get;}

Kuyruk yöneticisi tarafından, iletinin yerleştirildiği kuyruğun adına ayarlanan bir çıkış alanı. ResolvedQueueName (Çözüm Kuyruğu Adı), açılan kuyruk bir diğer ad ya da model kuyruğuysa, kuyruğu açmak için kullanılan addan farklı olabilir.

Boş olmayan bir değer, nesne tek bir kuyruksa döndürülür. Nesne bir dağıtım listesi ya da konuyla, döndürülen değer tanımsız olur.

```
public int UnknownDestCount {get;} *
```

Genellikle dağıtım listeleri için kullanılan UnknownDestCount , kuyruk yöneticisi tarafından ayarlanan bir çıkış alanıdır. Yürürlükteki çağrının uzak kuyruklara çözülecek kuyruklara başarıyla gönderdiği ileti sayısını bildirir.

.NET dağıtım listelerini desteklemez, ancak tek bir kuyruk açılırken InvalidDestCount ayarlanır.

Oluşturucular

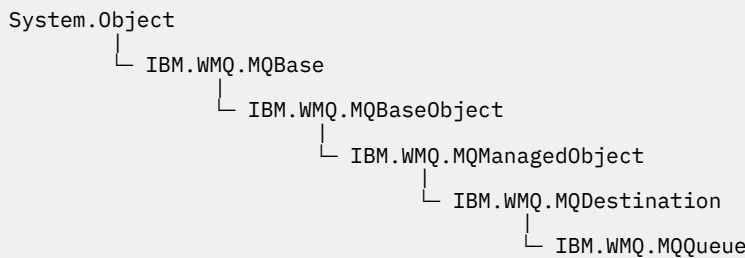
```
public MQPutMessageOptions();
```

Seçenek ayarlanmamış yeni bir MQPutMessageOptions nesnesi ve boş bir ResolvedQueueName ve ResolvedQueueManagerName oluşturun.

MQQueue.NET sınıfı

Bir IBM MQ kuyruğuna ilişkin ileti ve sorgu özniteliklerini göndermek ve almak için MQQueue komutunu kullanın. Bir oluşturucu ya da MQQueueManager .AccessProcess yöntemini kullanarak bir MQQueue nesnesi yaratın.

Sınıf



```
public class IBM.WMQ.MQQueue extends IBM.WMQ.MQDestination;
```

- [“Özellikler” sayfa 1726](#)
- [“yöntemler” sayfa 1728](#)
- [“Oluşturucular” sayfa 1731](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

```
public int ClusterWorkLoadPriority {get;}
```

Kuyruğun önceliğini belirler. Bu parametre yalnızca yerel, uzak ve diğer ad kuyrukları için geçerlidir.

```
public int ClusterWorkLoadRank {get;}
```

Kuyruğun sırasını belirtir. Bu parametre yalnızca yerel, uzak ve diğer ad kuyrukları için geçerlidir.

```
public int ClusterWorkLoadUseQ {get;}
```

Hedef kuyruksa yerel bir yönetim ortamı ve en az bir uzak küme yönetim ortamı olduğunda MQPUT işleminin davranışını belirtir. MQPUT bir küme kanalından kaynaklandıysa bu parametre geçerli değildir. Bu parametre yalnızca yerel kuyruklar için geçerlidir.

```
public DateTime CreationDateTime {get;}
```

Bu kuyruğun yaratıldığı tarih ve saat.

```
public int CurrentDepth {get;}
```

Kuyruksa bulunan iletilerin sayısını alır. Bu değer, bir koyma çağrısı sırasında ve alma çağrısı geri çekilirken artırılır. Bu, göz atmayan bir alma sırasında ve bir koyma çağrısının geriletme sırasında azalır.

public int DefinitionType {get;}

Kuyruğun nasıl tanımlandığını. Olası değerler şunlardır:

- MQC.MQQDT_PREDEFINED
- MQC.MQQDT_PERMANENT_DYNAMIC
- MQC.MQQDT_TEMPORARY_DYNAMIC

public int InhibitGet {get; set;}

Bu kuyrukta ya da bu konuda ileti alıp almayacağınızı denetler. Olası değerler şunlardır:

- MQC.MQQA_GET_INHIBITED
- MQC.MQQA_GET_ALLOWED

public int InhibitPut {get; set;}

İletileri bu kuyruğa mı, yoksa bu konuya mı koyabileceğinizi denetler. Olası değerler şunlardır:

- MQQA_PUT_INHIBITED
- MQQA_PUT_ALLOWED

public int MaximumDepth {get;}

Herhangi bir anda kuyrukta bulunabilecek ileti sayısı üst sınırı. Bu kadar çok ileti içeren bir iletiyi kuyruğa koyma girişimi MQC.MQRC_Q_FULLneden koduyla başarısız olur.

public int MaximumMessageLength {get;}

Bu kuyruktaki her iletide bulunabilecek uygulama verilerinin uzunluk üst sınırı. Bu değerden büyük bir iletiyi koyma girişimi MQC.MQRC_MSG_TOO_BIG_FOR_Qneden koduyla başarısız oldu.

public int NonPersistentMessageClass {get;}

Bu kuyruğa konan kalıcı olmayan iletiler için güvenilirlik düzeyi.

public int OpenInputCount {get;}

Kuyruktan ileti kaldırmak için geçerli olan tanıtıcı sayısı. OpenInputCount , yalnızca uygulama tarafından yaratılan tanıtıcıları değil, yerel kuyruk yöneticisi tarafından bilinen geçerli giriş tanıtıcılarının toplam sayısıdır.

public int OpenOutputCount {get;}

Kuyruğa ileti eklemek için geçerli olan tanıtıcı sayısı. OpenOutputCount , yalnızca uygulama tarafından yaratılan tanıtıcıları değil, yerel kuyruk yöneticisi tarafından bilinen geçerli çıkış tanıtıcılarının toplam sayısıdır.

public int QueueAccounting {get;}

Kuyruğa ilişkin hesap bilgilerinin toplanmasını etkinleştirip etkinleştiremeyeceğinizi belirler.

public int QueueMonitoring {get;}

Kuyruk için izlemeyi etkinleştirip etkinleştiremeyeceğinizi belirler.

public int QueueStatistics {get;}

Kuyruğa ilişkin istatistiklerin toplanmasını etkinleştirip etkinleştiremeyeceğinizi belirler.

public int QueueType {get;}

Aşağıdaki değerlerden birini içeren bu kuyruğun tipi:

- MQC.MQQT_ALIAS
- MQC.MQQT_LOCAL
- MQC.MQQT_REMOTE
- MQC.MQQT_CLUSTER

public int Shareability {get;}

Kuyruğun bir kereden fazla giriş için açılıp açılmayacağını belirler. Olası değerler şunlardır:

- MQC.MQQA_SHAREABLE
- MQC.MQQA_NOT_SHAREABLE

public string TPIPE {get;}

IBM MQ IMS köprüsü kullanılarak OTMA ile iletişim için kullanılan TPIPE adı.

public int TriggerControl {get; set;}

Tetikleyici iletilerinin bir başlatma kuyruğuna yazılıp yazılmayacağı, kuyruğa hizmet vermek üzere bir uygulama başlatılıp başlatılmayacağı. Olası değerler şunlardır:

- MQC.MQTC_OFF
- MQC.MQTC_ON

public string TriggerData {get; set;}

Kuyruk yöneticisinin tetikleyici iletilerine eklediği serbest biçimli veriler. Bu kuyruğa gelen bir ileti, başlatma kuyruğuna bir tetikleyici iletilerinin yazılmasına neden olduğunda TriggerData ekler. Dizginin izin verilen uzunluk üst sınırı MQC.MQ_TRIGGER_DATA_LENGTH tarafından verilir.

public int TriggerDepth {get; set;}

Tetikleyici tipi MQC.MQTT_DEPTH olarak ayarlandığında, bir tetikleyici iletilerinin yazılmadan önce kuyruktaki olması gereken ileti sayısı.

public int TriggerMessagePriority {get; set;}

İletiler, tetikleyici iletilerinin oluşturulmasına katkıda bulunmadığı ileti önceliği. Yani, kuyruk yöneticisi tetikleyici oluşturulup oluşturulmayacağına karar verirken bu iletileri yoksayar. Sıfır değeri, tüm iletilerinin tetikleyici iletilerinin oluşturulmasına katkıda bulunmasına neden olur.

public int TriggerType {get; set;}

Bu kuyruğa gelen iletilerinin sonucu olarak tetikleyici iletilerinin yazıldığı koşullar. Olası değerler şunlardır:

- MQC.MQTT_NONE
- MQC.MQTT_FIRST
- MQC.MQTT_EVERY
- MQC.MQTT_DEPTH

yöntemler

public void Get(MQMessage message);

public void Get(MQMessage message, MQGetMessageOptions getMessageOptions);

public void Get(MQMessage message, MQGetMessageOptions getMessageOptions, int MaxMsgSize);

MQException ögesini atar.

Kuyruktan bir ileti alır.

Alma işlemi başarısız olursa, MQMessage nesnesi değişmez. Başarılı olursa, MQMessage ileti tanımlayıcısı ve ileti verileri bölümleri, gelen iletideki ileti tanımlayıcısı ve ileti verileriyle değiştirilir.

Belirli bir MQQueueManager içinden IBM MQ 'e yapılan tüm çağrılar zamanuyumdur. Bu nedenle, beklemeyle alma işlemi gerçekleştirirseniz, aynı MQQueueManager ögesini kullanan diğer tüm iş parçacıklarının, Get çağrısı tamamlanıncaya kadar başka IBM MQ çağrısı yapması engellenir. IBM MQ 'e aynı anda erişmek için birden çok iş parçacığına gereksinim duyarsanız, her iş parçacığı kendi MQQueueManager nesnesini oluşturmalıdır.

ileti

İleti tanımlayıcıyı ve döndürülen ileti verilerini içerir. İleti tanımlayıcıdaki alanlardan bazıları giriş değiştirgeleridir. MessageId ve CorrelationId giriş parametrelerinin gerektiği gibi ayarlandığından emin olmak önemlidir.

Yeniden bağlanabilir bir istemci, MQGM_SYNCPOINT altında alınan iletiler için başarılı bir yeniden bağlantıdan sonra MQRC_BACKED_OUT neden kodunu döndürür.

getMessageSeçenekleri

Alma işlemi denetleyen seçenekler.

MQC.MQGMO_CONVERT seçeneğinin kullanılması, tek baytlık karakter kodlarından çift baytlık kodlara dönüştürülürken MQC.MQRC_CONVERTED_STRING_TOO_BIG neden koduyla bir kural dışı durumla sonuçlanabilir. Bu durumda, ileti dönüştürme yapılmadan arabelleğe kopyalanır.

getMessageOptions belirtilmezse, kullanılan ileti seçeneği MQGMO_NOWAIT olur.

Yeniden bağlanabilir bir istemcide MQGMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

MaxMsgBoyutu

Bu ileti nesnesinin alacağı en büyük ileti. Kuyruktaki ileti bu boyuttan büyükse, iki şeyden biri oluşur:

- MQGetMessageOptions nesnesinde MQGMO_ACCEPT_TRUNCATED_MSG işareti ayarlanırsa, ileti mümkün olduğunca çok ileti verileriyle doldurulur. MQCC_WARNING tamamlanma kodu ve MQRC_TRUNCATED_MSG_ACCEPTED neden koduyla bir kural dışı durum yayınlandı.
- MQGMO_ACCEPT_TRUNCATED_MSG işareti ayarlanmazsa, ileti kuyruğa kalır. MQCC_WARNING tamamlanma kodu ve MQRC_TRUNCATED_MSG_FAILED neden koduyla bir kural dışı durum yayınlandı.

MaxMsgSize belirtilmezse, iletinin tamamı alınır.

```
public void Put(MQMessage message);  
public void Put(MQMessage message, MQPutMessageOptions putMessageOptions);
```

MQException ögesini atar.

Bir iletiyi kuyruğa koyar.

Koyma çağrısı tamamlandıktan sonra MQMessage nesnesinde yapılan değişiklikler, IBM MQ kuyruğundaki ya da yayın konusundaki gerçek iletiyi etkilemez.

Put , MQMessage nesnesinin MessageId ve CorrelationId özelliklerini günceller ve ileti verilerini temizlemez. Diğer Put ya da Get çağrıları, MQMessage nesnesindeki güncellenen bilgilere başvurmaktadır. Örneğin, aşağıdaki kod parçasığına ilk ileti a ve ikinci abıçerir.

```
msg.WriteString("a");  
q.Put(msg, pmo);  
msg.WriteString("b");  
q.Put(msg, pmo);
```

ileti

Gönderilecek ileti tanımlayıcı verilerini ve iletiyi içeren bir MQMessage nesnesi. İleti tanımlayıcısı, bu yöntemin sonucu olarak değiştirilebilir. Bu yöntem tamamlandıktan hemen sonra ileti tanımlayıcısındaki değerler, kuyruğa konan ya da konuya yayınlanan değerlerdir.

Yeniden bağlanabilir bir istemciye aşağıdaki neden kodları döndürülür:

- MQRC_CALL_INTERRUPTED kalıcı bir iletide Put çağrısı çalıştırılırken bağlantı kesilirse ve yeniden bağlantı başarılı olursa.
- MQRC_NONE kalıcı olmayan bir iletide Put çağrısı çalıştırılırken bağlantı başarılı olursa (bkz. [Application Recovery](#)).

putMessageSeçenekleri

Koyma işleminin eylemini denetleyen seçenekler.

putMessageOptions belirtilmezse, varsayılan MQPutMessageOptions eşgörünümü kullanılır.

Yeniden bağlanabilir bir istemcide MQGMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

Not: Kolaylık ve başarı için, kuyruğa tek bir ileti koymak istiyorsanız MQQueueManager . Put nesnesini kullanın. Bunun için bir MQQueue nesnesine sahip olun.

```
public void PutForwardMessage(MQMessage message);  
public void PutForwardMessage(MQMessage message, MQPutMessageOptions  
putMessageOptions);
```

MQException ögesini atar

Kuyruğa iletilen bir ileti koyun; burada *message* özgün iletidir.

ileti

Gönderilecek ileti tanımlayıcı verilerini ve iletiyi içeren bir MQMessage nesnesi. İleti tanımlayıcısı, bu yöntemin sonucu olarak değiştirilebilir. Bu yöntem tamamlandıktan hemen sonra ileti tanımlayıcısındaki değerler, kuyruğa konan ya da konuya yayınlanan değerlerdir.

Yeniden bağlanabilir bir istemciye aşağıdaki neden kodları döndürülür:

- MQRC_CALL_INTERRUPTED kalıcı bir iletide Put çağrısı çalıştırılırken bağlantı kesilirse ve yeniden bağlantı başarılı olursa.
- MQRC_NONE kalıcı olmayan bir iletide Put çağrısı çalıştırılırken bağlantı başarılı olursa (bkz. [Application Recovery](#)).

putMessageSeçenekleri

Koyma işleminin eylemini denetleyen seçenekler.

putMessageOptions belirtilmezse, varsayılan MQPutMessageOptions eşgörünümü kullanılır.

Yeniden bağlanabilir bir istemcide MQPMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

```
public void PutReplyMessage(MQMessage message)  
public void PutReplyMessage(MQMessage message, MQPutMessageOptions  
putMessageOptions)
```

MQExceptionöğesini atar.

Kuyruğa bir yanıt iletisi koyun; burada *message* özgün iletidir.

ileti

İleti tanımlayıcıyı ve döndürülen ileti verilerini içerir. İleti tanımlayıcıdaki alanlardan bazıları giriş deęiřtirgeleridir. MessageId ve CorrelationId giriş parametrelerinin gerektięi gibi ayarlandıęından emin olmak önemlidir.

Yeniden bağlanabilir bir istemci, MQGM_SYNCPOINTaltında alınan iletiler için başarılı bir yeniden bağlantıdan sonra MQRC_BACKED_OUT neden kodunu döndürür.

putMessageSeçenekleri

Koyma işleminin eylemini denetleyen seçenekler.

putMessageOptions belirtilmezse, varsayılan MQPutMessageOptions eşgörünümü kullanılır.

Yeniden bağlanabilir bir istemcide MQPMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

```
public void PutReportMessage(MQMessage message)  
public void PutReportMessage(MQMessage message, MQPutMessageOptions  
putMessageOptions)
```

MQExceptionöğesini atar.

Kuyruğa bir rapor iletisi koyun; burada *message* özgün iletidir.

ileti

İleti tanımlayıcıyı ve döndürülen ileti verilerini içerir. İleti tanımlayıcıdaki alanlardan bazıları giriş deęiřtirgeleridir. MessageId ve CorrelationId giriş parametrelerinin gerektięi gibi ayarlandıęından emin olmak önemlidir.

Yeniden bağlanabilir bir istemci, MQGM_SYNCPOINTaltında alınan iletiler için başarılı bir yeniden bağlantıdan sonra MQRC_BACKED_OUT neden kodunu döndürür.

putMessageSeçenekleri

Koyma işleminin eylemini denetleyen seçenekler.

putMessageOptions belirtilmezse, varsayılan MQPutMessageOptions eşgörünümü kullanılır.

Yeniden bağlanabilir bir istemcide MQPMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

Olusturucular

```
public MQQueue MQQueueManager.AccessQueue(string queueName, int openOptions);  
public MQQueue MQQueueManager.AccessQueue(string queueName, int openOptions,  
string queueManagerName, string dynamicQueueName, string alternateUserId);
```

MQException ögesini atar.

Bu kuyruk yöneticisindeki bir kuyruğa erişir.

İletileri alabilir ya da göz atabilir, iletileri koyabilir, kuyruğun özniteliklerini sorgulayabilir ya da kuyruğun özniteliklerini ayarlayabilirsiniz. Adı belirtilen kuyruk bir model kuyruğuysa, dinamik bir yerel kuyruk yaratılır. Dinamik kuyruğun adını öğrenmek için name sonuçtaki MQQueue nesnesinin özneliğini sorgulayın.

queueName

Açılacak kuyruğun adı.

openOptions

Kuyruğun açılmasını denetleyen seçenekler.

MQC.MQOO_ALTERNATE_USER_AUTHORITY

Belirtilen kullanıcı kimliğiyle doğrulayın.

MQC.MQOO_BIND_AS_QDEF

Kuyruk için varsayılan bağ tanımını kullan.

MQC.MQOO_BIND_NOT_FIXED

Belirli bir hedefe bağlanmayın.

MQC.MQOO_BIND_ON_OPEN

Kuyruk açıldığında hedef için bağ tanımlama tanıtıcısı.

MQC.MQOO_BROWSE

İletiyeye göz atmak için açın.

MQC.MQOO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturulursa başarısız olur.

MQC.MQOO_INPUT_AS_Q_DEF

Kuyruk tanımlı varsayılan değeri kullanarak iletileri almak için açın.

MQC.MQOO_INPUT_SHARED

Paylaşılan erişime sahip iletileri almak için açın.

MQC.MQOO_INPUT_EXCLUSIVE

Özel erişime sahip iletileri almak için açın.

MQC.MQOO_INQUIRE

Sorgu için aç-özellikleri sorgulamak istiyorsanız gereklidir.

MQC.MQOO_OUTPUT

İletileri koymak için açık.

MQC.MQOO_PASS_ALL_CONTEXT

Tüm bağlamın iletilmesine izin ver

MQC.MQOO_PASS_IDENTITY_CONTEXT

Kimlik bağlamının iletilmesine izin ver.

MQC.MQOO_SAVE_ALL_CONTEXT

İleti alındığında bağlamı kaydet.

MQC.MQOO_SET

Öznitelikleri ayarlamak için açın-özellikleri ayarlamak istiyorsanız gereklidir.

MQC.MQOO_SET_ALL_CONTEXT

Tüm bağlamın ayarlanmasına izin verir.

MQC.MQOO_SET_IDENTITY_CONTEXT

Kimlik bağlamının ayarlanmasına izin verir.

queueManagerAdı

Kuyruğun tanımlandığı kuyruk yöneticisinin adı. Tamamen boş ya da boş olan bir ad, MQQueueManager nesnesinin bağlı olduğu kuyruk yöneticisini belirtir.

dynamicQueueAdı

dynamicQueueName , *queueName* bir model kuyruğunun adını belirtmedikçe yoksayılr. Varsa, *dynamicQueueName* yaratılacak dinamik kuyruğun adını belirtir. *queueName* bir model kuyruğunun adını belirtiyorsa, boş ya da boş bir ad geçerli değildir. Addaki son boşluk olmayan karakter bir yıldız işaretiyse (*), kuyruk yöneticisi yıldız işaretini bir karakter dizisiyle değiştirir. Kuyruk için oluşturulan adın bu kuyruk yöneticisinde benzersiz olduğunu garanti eden karakterler.

alternateUserKimliği

openOptions değiştirgesinde MQC.MQOO_ALTERNATE_USER_AUTHORITY belirtilirse, *alternateUserId* açma yetkisini denetlemek için kullanılan diğer kullanıcı kimliğini belirtir. MQC.MQOO_ALTERNATE_USER_AUTHORITY belirtilmezse, *alternateUserId* boş bırakılabilir ya da boş bırakılabilir.

```
public MQQueue(MQQueueManager queueManager, string queueName, int openOptions,  
string queueManagerName, string dynamicQueueName, string alternateUserId);
```

MQExceptionögesini atar.

queueManagerüzerinde bir kuyruğa erişir.

İletileri alabilir ya da göz atabilir, iletileri koyabilir, kuyruğun özniteliklerini sorgulayabilir ya da kuyruğun özniteliklerini ayarlayabilirsiniz. Adı belirtilen kuyruk bir model kuyruğuysa, dinamik bir yerel kuyruk yaratılır. Dinamik kuyruğun adını öğrenmek için name sonuçtaki MQQueue nesnesinin özniteliğini sorgulayın.

queueManager

Kuyruğa erişmek için kuyruk yöneticisi.

queueName

Açılacak kuyruğun adı.

openOptions

Kuyruğun açılmasını denetleyen seçenekler.

MQC.MQOO_ALTERNATE_USER_AUTHORITY

Belirtilen kullanıcı kimliğiyle doğrulayın.

MQC.MQOO_BIND_AS_QDEF

Kuyruk için varsayılan bağ tanımını kullan.

MQC.MQOO_BIND_NOT_FIXED

Belirli bir hedefe bağlanmayın.

MQC.MQOO_BIND_ON_OPEN

Kuyruk açıldığında hedef için bağ tanımlama tanıtıcısı.

MQC.MQOO_BROWSE

İletiyeye göz atmak için açın.

MQC.MQOO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturulursa başarısız olur.

MQC.MQOO_INPUT_AS_Q_DEF

Kuyruk tanımlı varsayılan değeri kullanarak iletileri almak için açın.

MQC.MQOO_INPUT_SHARED

Paylaşılan erişime sahip iletileri almak için açın.

MQC.MQOO_INPUT_EXCLUSIVE

Özel erişime sahip iletileri almak için açın.

MQC.MQOO_INQUIRE

Sorgu için aç-özellikleri sorgulamak istiyorsanız gereklidir.

MQC.MQOO_OUTPUT

İletileri koymak için açık.

MQC.MQOO_PASS_ALL_CONTEXT

Tüm bağlamın iletilmesine izin ver

MQC.MQOO_PASS_IDENTITY_CONTEXT

Kimlik bağlamının iletilmesine izin ver.

MQC.MQOO_SAVE_ALL_CONTEXT

İleti alındığında bağlamı kaydet.

MQC.MQOO_SET

Öznitelikleri ayarlamak için açın-özellikleri ayarlamak istiyorsanız gereklidir.

MQC.MQOO_SET_ALL_CONTEXT

Tüm bağlamın ayarlanmasına izin verir.

MQC.MQOO_SET_IDENTITY_CONTEXT

Kimlik bağlamının ayarlanmasına izin verir.

queueManagerAdı

Kuyruğun tanımlandığı kuyruk yöneticisinin adı. Tamamen boş ya da boş olan bir ad, MQQueueManager nesnesinin bağlı olduğu kuyruk yöneticisini belirtir.

dynamicQueueAdı

dynamicQueueName , *queueName* bir model kuyruğunun adını belirtmedikçe yoksayıdır. Varsa, *dynamicQueueName* yaratılacak dinamik kuyruğun adını belirtir. *queueName* bir model kuyruğunun adını belirtiyorsa, boş ya da boş bir ad geçerli değildir. Addaki son boşluk olmayan karakter bir yıldız işaretiyse (*), kuyruk yöneticisi yıldız işaretini bir karakter dizisiyle değiştirir. Kuyruk için oluşturulan adın bu kuyruk yöneticisinde benzersiz olduğunu garanti eden karakterler.

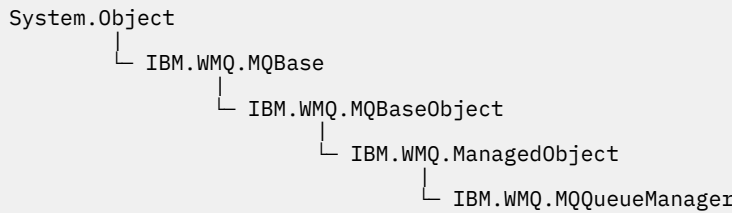
alternateUserKimliği

openOptions değiştirilmesinde MQC.MQOO_ALTERNATE_USER_AUTHORITY belirtilirse, *alternateUserId* açma yetkisini denetlemek için kullanılan diğer kullanıcı kimliğini belirtir. MQC.MQOO_ALTERNATE_USER_AUTHORITY belirtilmezse, *alternateUserId* boş bırakılabilir ya da boş bırakılabilir.

MQQueueManager.NET sınıfı

Bir kuyruk yöneticisine bağlanmak ve kuyruk yöneticisi nesnelere erişmek için MQQueueManager ögesini kullanın. Ayrıca işlemleri de kontrol eder. MQQueueManager oluşturucusu bir istemci ya da sunucu bağlantısı yaratır.

Sınıf



```
public class IBM.WMQ.MQQueueManager extends IBM.WMQ.MQManagedObject;
```

- [“Özellikler” sayfa 1733](#)
- [“yöntemler” sayfa 1737](#)
- [“Oluşturucular” sayfa 1742](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

public int AccountingConnOverride {get;}

Uygulamaların MQI muhasebe ve kuyruk muhasebesi değerlerinin ayarını geçersiz kılıp kılamayacağını belirler.

public int AccountingInterval {get;}

Ara muhasebe kayıtlarının yazılması için geçen süre (saniye).

public int ActivityRecording {get;}

Etkinlik raporlarının oluşturulmasını denetler.

public int AdoptNewMCACheck {get;}

Yeni bir gelen kanal algılandığında MCA 'nın benimsenip benimsenmediğini belirlemek için hangi öğelerin denetleneceğini belirtir. Benimsenmesi için MCA adı, etkin bir MCA adıyla eşleşmelidir.

public int AdoptNewMCAInterval {get;}

Yeni kanalın artık kanalın sona ermesini bekleyeceği süre (saniye).

public int AdoptNewMCAType {get;}

AdoptNewMCACheck değeriyle eşleşen yeni bir gelen kanal isteği saptandığında, artık kalmış bir MCA eşgörünümünün kullanılıp kullanılmayacağını (yeniden başlatılıp başlatılmayacağını) belirler.

public int BridgeEvent {get;}

IMS köprü olaylarının oluşturulup oluşturulmadığını belirler.

public int ChannelEvent {get;}

Kanal olaylarının oluşturulup oluşturulmayacağını belirler.

public int ChannelInitiatorControl {get;}

Kuyruk yöneticisi başlatıldığında kanal başlatıcısının otomatik olarak başlatılıp başlatılmayacağını belirler.

public int ChannelInitiatorAdapters {get;}

IBM MQ çağrılarının işleneceği bağdaştırıcı alt görevlerinin sayısı.

public int ChannelInitiatorDispatchers {get;}

Kanal başlatıcısı için kullanılacak dağıtıcıların sayısı.

public int ChannelInitiatorTraceAutoStart {get;}

Kanal başlatıcı izlemesinin otomatik olarak başlatılıp başlatılmayacağını belirler.

public int ChannelInitiatorTraceTableSize {get;}

Kanal başlatıcısının izleme verileri alanının megabayt cinsinden büyüklüğü.

public int ChannelMonitoring {get;}

Kanal izlemenin kullanılıp kullanılmadığını belirler.

public int ChannelStatistics {get;}

Kanallar için istatistik verilerinin toplanmasını denetler.

public int CharacterSet {get;}

Kuyruk yöneticisinin kodlanmış karakter takımı tanıtıcısını (CCSID) döndürür. CharacterSet , kuyruk yöneticisi tarafından uygulama programlama arabirimindeki tüm karakter dizgisi alanları için kullanılır.

public int ClusterSenderMonitoring {get;}

Otomatik olarak tanımlanan küme gönderen kanalları için çevrimiçi izleme verilerinin toplanmasını denetler.

public int ClusterSenderStatistics {get;}

Otomatik olarak tanımlanan küme gönderen kanalları için istatistik verilerinin toplanmasını denetler.

public int ClusterWorkLoadMRU {get;}

Giden küme kanalı sayısı üst sınırı.

public int ClusterWorkLoadUseQ {get;}

QMGRdeğerini belirtiyorsa, MQQueue özelliğinin varsayılan değeri olan ClusterWorkLoadUseQ.

public int CommandEvent {get;}

Komut olaylarının oluşturulup oluşturulmayacağını belirtir.

public string CommandInputQueueName {get;}

Kuyruk yöneticisinde tanımlanan komut giriş kuyruğunun adını döndürür. Uygulama yetkisi varsa, bu kuyruğa komut gönderebilir.

public int CommandLevel {get;}

Kuyruk yöneticisinin işlev düzeyini gösterir. Belirli bir işlev düzeyine karşılık gelen işlevler kümesi, platforma bağlıdır. Belirli bir altyapıda, tüm kuyruk yöneticilerinin ortak olduğu en düşük işlevsel düzeyde işlevleri destekleyen her kuyruk yöneticisine güvenebilirsiniz.

public int CommandLevel {get;}

Kuyruk yöneticisi başlatıldığında komut sunucusunun otomatik olarak başlatılıp başlatılmayacağını belirler.

public string DNSGroup {get;}

Artık kullanılmıyor.

public int DNSWLM {get;}

Artık kullanılmıyor.

public int IPAddressVersion {get;}

Kanal bağlantısı için kullanılacak IP iletişim kuralı (IPv4 ya da IPv6).

public boolean IsConnected {get;}

isConnecteddeğerini döndürür.

Değeri true ise, kuyruk yöneticisiyle bağlantı kurulur ve kesildiği bilinmez. IsConnected çağrıları, kuyruk yöneticisine etkin olarak ulaşmaya çalışmaz, bu nedenle fiziksel bağlantı bozulabilir, ancak IsConnected yine de true değerini döndürebilir. IsConnected durumu yalnızca, kuyruk yöneticisinde bir ileti koyma, ileti alma gibi bir etkinlik gerçekleştirildiğinde güncellenir.

Değeri false ise, kuyruk yöneticisiyle bağlantı kurulmadı ya da koptu ya da bağlantısı kesildi.

public int KeepAlive {get;}

Bağlantının diğer ucunun hala kullanılabilir olup olmadığını denetlemek için TCP KEEPALIVE olanağının kullanılıp kullanılmayacağını belirtir. Kullanılmıyorsa, kanal kapalıdır.

public int ListenerTimer {get;}

IBM MQ ' in bir APPC ya da TCP/IP hatasından sonra dinleyiciyi yeniden başlatma girişimleri arasındaki saniye cinsinden zaman aralığı.

public int LoggerEvent {get;}

Günlüğe kaydedici olaylarının oluşturulup oluşturulmayacağını belirler.

public string LU62ARMSuffix {get;}

SYS1.PARMLIB. Bu sonek, bu kanal başlatıcısı için LUADD ' yi verir. Otomatik yeniden başlatma yöneticisi (ARM) kanal başlatıcısı yeniden başlattığında, z/OS komutu SET APPC=xx yayınlanır.

public string LUGroupName {get; z/os}

Kuyruk paylaşım grubuna ilişkin gelen iletimleri işleyen LU 6.2 dinleyicisi tarafından kullanılacak soysal LU adı.

public string LUName {get;}

Giden LU 6.2 iletimleri için kullanılacak LU ' nun adı.

public int MaximumActiveChannels {get;}

Herhangi bir zamanda etkin olabilecek kanal sayısı üst sınırı.

public int MaximumCurrentChannels {get;}

Herhangi bir zamanda geçerli olabilecek kanal sayısı üst sınırı (bağlı istemcilere sahip sunucu bağlantısı kanalları da içinde olmak üzere).

public int MaximumLU62Channels {get;}

LU 6.2 iletim protokolünü kullanan, yürürlükteki kanal ya da bağlanabilen istemci sayısı üst sınırı.

public int MaximumMessageLength {get;}

Kuyruk yöneticisi tarafından işlenebilecek bir iletinin (bayt cinsinden) uzunluk üst sınırını döndürür. İleti uzunluğu üst sınırı MaximumMessageLengthdeğerinden büyük olan bir kuyruk tanımlanamaz.

public int MaximumPriority {get;}

Kuyruk yöneticisi tarafından desteklenen ileti önceliği üst sınırını döndürür. Öncelikler sıfır (en düşük) ile bu değer arasında değişir. Kuyruk yöneticisiyle bağlantıyı kestikten sonra bu yöntemi çağırırsanız MQException yayınlanır.

public int MaximumTCPChannels {get;}

TCP/IP iletim protokolünü kullanan, yürürlükteki kanal ya da bağlanabilen istemci sayısı üst sınırı.

public int MQIAccounting {get;}

MQI verileri için muhasebe bilgileri toplamasını denetler.

public int MQIStatistics {get;}

Kuyruk yöneticisine ilişkin istatistik izleme bilgilerinin toplanmasını denetler.

public int OutboundPortMax {get;}

Giden kanallar için bağ tanımlanırken kullanılacak kapı numaraları aralığındaki değer üst sınırı.

public int OutboundPortMin {get;}

Giden kanallar için bağ tanımlanırken kullanılacak kapı numaraları aralığındaki değer alt sınırı.

public int QueueAccounting {get;}

Tüm kuyruklar için sınıf 3 muhasebe (iş parçacığı düzeyi ve kuyruk düzeyi muhasebe) verilerinin kullanılıp kullanılmayacağını belirler.

public int QueueMonitoring {get;}

Kuyruklara ilişkin çevrimiçi izleme verilerinin toplanmasını denetler.

public int QueueStatistics {get;}

Kuyruklara ilişkin istatistik verilerinin toplanmasını denetler.

public int ReceiveTimeout {get;}

TCP/IP kanalının, etkin değil durumuna dönmeden önce ortağından veri (sağlıklı işletim bildirimleri de içinde olmak üzere) almayı beklediği süre.

public int ReceiveTimeoutMin {get;}

TCP/IP kanalının, etkin olmayan duruma dönmeden önce ortağından veri almayı (sağlıklı işletim bildirimleri de içinde olmak üzere) bekleyeceği süre alt sınırı.

public int ReceiveTimeoutType {get;}

ReceiveTimeout'teki değere uygulanacak niteleyici.

public int SharedQueueQueueManagerName {get;}

İletilerin paylaşılan bir kuyruğa nasıl teslim edileceğini belirtir. Koyma değeri, hedef kuyruk yöneticisiyle aynı kuyruk paylaşım grubundan farklı bir kuyruk yöneticisi belirtirse, ileti iki şekilde teslim edilir:

MQC.MQSQQM_USE

İletiler, paylaşılan kuyruğa konmadan önce nesne kuyruğu yöneticisine teslim edilir.

MQCMQSQQM_IGNORE

İletiler doğrudan paylaşılan kuyruğa yerleştirilir.

public int SSLEvent {get;}

TLS olaylarının oluşturulup oluşturulmadığını belirler.

public int SSLFips {get;}

Şifreleme, şifreleme donanımı yerine IBM MQ içinde gerçekleştirildiğinde yalnızca FIPS sertifikalı algoritmaların kullanılıp kullanılmayacağını belirler.

public int SSLKeyResetCount {get;}

Gizli anahtar yeniden anlaşılmadan önce bir TLS etkileşimi içinde gönderilen ve alınan şifrelenmemiş bayt sayısını belirtir.

public int ClusterSenderStatistics {get;}

İstatistiklerin ardışık toplanmaları arasındaki aralığı dakika cinsinden belirtir.

public int SyncpointAvailability {get;}

Kuyruk yöneticisinin MQQueue.get ve MQQueue.put yöntemleriyle iş birimlerini ve eşitleme noktalarını destekleyip desteklemediğini belirtir.

public string TCPName {get;}

TCPStackType değerine bağlı olarak, kullanılacak tek ya da varsayılan TCP/IP sisteminin adı.

public int TCPStackType {get;}

Kanal başlatıcının yalnızca TCPName(TCPName) içinde belirtilen TCP/IP adres alanını kullanıp kullanmayacağını belirler. Diğer bir seçenek olarak, kanal başlatıcı herhangi bir TCP/IP adresine bağlanabilir.

public int TraceRouteRecording {get;}

Rota izleme bilgilerinin kaydedilmesini denetler.

yöntemler

public MQProcess AccessProcess(string processName, int openOptions);

public MQProcess AccessProcess(string processName, int openOptions, string queueManagerName, string alternateUserId);

MQExceptionöğesini atar.

Süreç özniteliklerini sorgulamak için bu kuyruk yöneticisindeki bir IBM MQ işlemine erişin.

processName

Açılacak sürecin adı.

openOptions

Sürecin açılmasını denetleyen seçenekler. Bitset OR kullanılarak eklenebilen ya da birleştirilebilen geçerli seçenekler şunlardır:

- MQC.MQ00_FAIL_IF QUIESCING
- MQC.MQ00_INQUIRE
- MQC.MQ00_SET
- MQC.MQ00_ALTERNATE_USER_AUTHORITY

queueManagerAdı

İşlemin tanımlandığı kuyruk yöneticisinin adı. Kuyruk yöneticisi, işlemin eriştiği adla aynıysa, boş ya da boş bir kuyruk yöneticisi adı bırakabilirsiniz.

alternateUserKimliği

openOptions değiştirilmesinde MQC.MQ00_ALTERNATE_USER_AUTHORITY belirtilirse, *alternateUserId* işleme ilişkin yetkilendirmeyi denetlemek için kullanılan diğer kullanıcı kimliğini belirtir. MQ00_ALTERNATE_USER_AUTHORITY belirtilmezse, *alternateUserId* boş değerli ya da boş olabilir.

MQC.MQ00_ALTERNATE_USER_AUTHORITY belirtilmezse, kuyruk yöneticisine bağlanmak için varsayılan kullanıcı yetkisi kullanılır.

public MQQueue AccessQueue(string queueName, int openOptions);

public MQQueue AccessQueue(string queueName, int openOptions, string queueManagerName, string dynamicQueueName, string alternateUserId);

MQExceptionöğesini atar.

Bu kuyruk yöneticisindeki bir kuyruğa erişir.

İletileri alabilir ya da göz atabilir, iletileri koyabilir, kuyruğun özniteliklerini sorgulayabilir ya da kuyruğun özniteliklerini ayarlayabilirsiniz. Adı belirtilen kuyruk bir model kuyruğuysa, dinamik bir yerel kuyruk yaratılır. Dinamik kuyruğun adını öğrenmek için name sonuçtaki MQQueue nesnesinin özniteliğini sorgulayın.

queueName

Açılacak kuyruğun adı.

openOptions

Kuyruğun açılmasını denetleyen seçenekler.

MQC.MQ00_ALTERNATE_USER_AUTHORITY

Belirtilen kullanıcı kimliğiyle doğrulayın.

MQC.MQOO_BIND_AS_QDEF

Kuyruk için varsayılan bağ tanımını kullan.

MQC.MQOO_BIND_NOT_FIXED

Belirli bir hedefe bağlanmayın.

MQC.MQOO_BIND_ON_OPEN

Kuyruk açıldığında hedef için bağ tanımlama tanıtıcısı.

MQC.MQOO_BROWSE

İletiyeye göz atmak için açın.

MQC.MQOO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturulursa başarısız olur.

MQC.MQOO_INPUT_AS_Q_DEF

Kuyruk tanımlı varsayılan değeri kullanarak iletileri almak için açın.

MQC.MQOO_INPUT_SHARED

Paylaşılan erişime sahip iletileri almak için açın.

MQC.MQOO_INPUT_EXCLUSIVE

Özel erişime sahip iletileri almak için açın.

MQC.MQOO_INQUIRE

Sorgu için aç-özellikleri sorgulamak istiyorsanız gereklidir.

MQC.MQOO_OUTPUT

İletileri koymak için açık.

MQC.MQOO_PASS_ALL_CONTEXT

Tüm bağlamın iletilmesine izin ver

MQC.MQOO_PASS_IDENTITY_CONTEXT

Kimlik bağlamının iletilmesine izin ver.

MQC.MQOO_SAVE_ALL_CONTEXT

İleti alındığında bağlamı kaydet.

MQC.MQOO_SET

Öznitelikleri ayarlamak için açın-özellikleri ayarlamak istiyorsanız gereklidir.

MQC.MQOO_SET_ALL_CONTEXT

Tüm bağlamın ayarlanmasına izin verir.

MQC.MQOO_SET_IDENTITY_CONTEXT

Kimlik bağlamının ayarlanmasına izin verir.

queueManagerAdı

Kuyruğun tanımlandığı kuyruk yöneticisinin adı. Tamamen boş ya da boş olan bir ad, MQQueueManager nesnesinin bağlı olduğu kuyruk yöneticisini belirtir.

dynamicQueueAdı

dynamicQueueName , *queueName* bir model kuyruğunun adını belirtmedikçe yoksayılr. Varsa, *dynamicQueueName* yaratılacak dinamik kuyruğun adını belirtir. *queueName* bir model kuyruğunun adını belirtiyorsa, boş ya da boş bir ad geçerli değildir. Addaki son boşluk olmayan karakter bir yıldız işaretiyse (*), kuyruk yöneticisi yıldız işaretini bir karakter dizisiyle değiştirir. Kuyruk için oluşturulan adın bu kuyruk yöneticisinde benzersiz olduğunu garanti eden karakterler.

alternateUserKimliği

openOptions değiştirilmesinde MQC.MQOO_ALTERNATE_USER_AUTHORITY belirtilirse, *alternateUserId* açma yetkisini denetlemek için kullanılan diğer kullanıcı kimliğini belirtir. MQC.MQOO_ALTERNATE_USER_AUTHORITY belirtilmezse, *alternateUserId* boş bırakılabilir ya da boş bırakılabilir.

```

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);

```

Bu kuyruk yöneticisindeki bir konuya erişin.

MQTopic nesnelere, bazen konu nesnelere olarak adlandırılan yönetim konusu nesnelereyle yakından ilişkilidir. Girişte, *topicObject* bir yönetim konusu nesnesini gösterir. MQTopic oluşturucusu konu nesnesinden bir konu dizgisi alır ve konu adı oluşturmak için bunu *topicName* ile birleştirir. *topicObject* ya da *topicName* boş değerli olabilir. Konu adı, konu ağacına eşleştirilir ve *topicObject* içinde en yakın eşleşen yönetim konusu nesnesinin adı döndürülür.

MQTopic nesnesiyle ilişkili konular, iki konu dizgisini birleştirmenin sonucudur. İlk konu dizgisi, *topicObject* ile tanımlanan yönetim konusu nesnesi tarafından tanımlanır. İkinci konu dizgesi: *topicString*. MQTopic nesnesiyle ilişkilendirilen sonuçtaki konu dizgisi, genel arama karakterleri ekleyerek birden çok konuyu tanımlayabilir.

Konunun yayınlama ya da abone olma için açılıp açılmamasına bağlı olarak, konular üzerinde yayınlamak için MQTopic .Put yöntemlerini ya da konulara ilişkin yayınları almak için MQTopic .Get yöntemlerini kullanabilirsiniz. Aynı konuyu yayınlamak ve aynı konuya abone olmak istiyorsanız, konuya bir kez yayınlama, bir kez de abone olma olmak üzere iki kez erişmeniz gerekir.

Bir MQDestination nesnesi sağlamadan abonelik için bir MQTopic nesnesi oluşturursanız, yönetilen abonelik kabul edilir. Bir kuyruğu MQDestination nesnesi olarak geçirirseniz, yönetilmeyen abonelik kabul edilir. Ayarladığınız abonelik seçeneklerinin yönetilen ya da yönetilmeyen abonelik tutarlı olduğundan emin olmanız gerekir.

Hedef

destination bir MQQueue örneğidir. *destination* sağlanarak MQTopic , yönetilmeyen bir abonelik olarak açılır. Konuyla ilgili yayınlar, *destination* . olarak erişilen kuyruğa gönderilir.

topicName

Konu adının ikinci kısmı olan bir konu dizgisi. *topicName* , *topicObject* yönetim konusu nesnesinde tanımlanan konu dizgisiyle birleştirilmiştir. *topicName* değerini boş değere ayarlayabilirsiniz; bu durumda konu adı, *topicObject* içindeki konu dizgesiyle tanımlanır.

topicObject

Girişte *topicObject* , konu adının ilk kısmını oluşturan konu dizgisini içeren konu nesnesinin adıdır. *topicObject* içindeki konu dizgisi *topicName* ile birleştirilmiştir. Konu dizgileri oluşturma kuralları, [Konu dizgilerini birleştirme](#) içinde tanımlanır.

Çıkışta, *topicObject* konu ağacında konu dizgisiyle tanımlanan konuya en yakın eşleşen yönetim konusu nesnesinin adını içerir.

openAs

Yayınlamak ya da abone olmak için konuya erişin. Parametre aşağıdaki seçeneklerden yalnızca birini içerebilir:

- MQC.MQTOPIC_OPEN_AS_SUBSCRIPTION

- MQC.MQTOPIC_OPEN_AS_PUBLICATION

seçenekler

Yayın ya da abonelik için konunun açılmasını denetleyen seçenekleri birleştirin. Aboneliğe ilişkin bir konuya erişmek için MQC.MQSO_* değişmezlerini ve yayına ilişkin bir konuya erişmek için MQC.MQOO_* değişmezlerini kullanın.

Birden çok seçenek gerekiyorsa, değerleri birlikte ekleyin ya da bit şeklinde OR işlecini kullanarak seçenek değerlerini birleştirin.

alternateUserKimliği

İşlemi tamamlamak için gereken yetkiyi denetlemek için kullanılan diğer kullanıcı kimliğini belirtin. Seçenekler parametresinde MQC.MQOO_ALTERNATE_USER_AUTHORITY ya da MQC.MQSO_ALTERNATE_USER_AUTHORITY belirlendiyse, *alternateUserId* belirtmeniz gerekir.

subscriptionName

MQC.MQSO_DURABLE ya da MQC.MQSO_ALTER seçenekleri sağlanırsa *subscriptionName* gereklidir. Her iki durumda da MQTopic, abonelik için örtük olarak açılır. MQC.MQSO_DURABLE ayarlandıysa, abonelik varsa ya da MQC.MQSO_ALTER ayarlandıysa ve abonelik yoksa bir özel durum oluşur.

Özellikler

Bir HASH çizelgesini kullanarak listelenen özel abonelik özelliklerinden herhangi birini ayarlayın. HASH çizelgesinde belirlenen girişler çıkış değerleriyle güncellenir. Girişler, çıkış değerlerini raporlamak için HASH çizelgesine eklenmez.

- 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();

MQException ögesini atar

Kuyruk yöneticisi bağlantısına ilişkin zamanuyumsuz etkinliği gösteren bir MQAsyncStatus nesnesi döndürür.

public void Backout();

MQException ögesini atar.

Son eşitleme noktasından bu yana eşitleme noktası içinde okunan ya da yazılan iletileri geri çevirin.

MQC.MQPMO_SYNCPOINT işaret kümesiyle yazılan iletiler kuyruklardan kaldırılır.

MQC.MQGMO_SYNCPOINT işaretiyle okunan iletiler, geldikleri kuyruklarda yeniden yürürlüğe girmiştir. İletiler kalıcıysa, değişiklikler günlüğe kaydedilir.

Yeniden bağlanabilir istemciler için, yeniden bağlanma başarılı olduktan sonra istemciye MQRC_NONE neden kodu döndürülür.

public void Begin();

MQException ögesini atar.

Begin yalnızca sunucu bağ tanımları kipinde desteklenir. Küresel bir iş birimi başlatır.

public void Commit();

MQException ögesini atar.

Son eşitleme noktasından bu yana eşitleme noktası içinde okunan ya da yazılan iletileri kesinleştirin.

MQC.MQPMO_SYNCPOINT işaret kümesiyle yazılan iletiler diğer uygulamaların kullanımına sunulur. MQC.MQGMO_SYNCPOINT işaret kümesiyle alınan iletiler silinir. İletiler kalıcıysa, değişiklikler günlüğe kaydedilir.

Yeniden bağlanabilir bir istemciye aşağıdaki neden kodları döndürülür:

- Kesinleştirme çağrısı sırasında bağlantı kesilirse MQRC_CALL_INTERRUPTED .
- Yeniden bağlantıdan sonra kesinleştirme çağrısı verilirse MQRC_BACKED_OUT .

Disconnect();

MQExceptionöğesini atar.

Kuyruk yöneticisiyle bağlantıyı kapatın. Bu kuyruk yöneticisinde erişilen tüm nesnelere artık bu uygulama erişemiyor. Nesnelere yeniden erişmek için bir MQQueueManager nesnesi oluşturun.

Genel olarak, bir iş biriminin parçası olarak gerçekleştirilen her iş kesinleştirilmiştir. Ancak, iş birimi .NETtarafından yönetiliyorsa, iş birimi geriye işlenebilir.

```
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);
```

MQExceptionöğesini atar.

Önce bir MQQueue ya da MQTopic nesnesi yaratmadan kuyruğa ya da konuya tek bir ileti yerleştirir.

queueName

İletinin yerleştirileceği kuyruğun adı.

destinationName

Hedef nesnenin adı. Bu, *typedegerine* bağlı olarak bir kuyruk ya da bir konudur.

tip

Hedef nesnenin tipi. Seçenekleri birleştirmemelisiniz.

MQC.MQOT_Q

Kuyruk

MQC.MQOT_TOPIC

Konu

queueManagerAdı

Kuyruğun tanımlandığı kuyruk yöneticisinin ya da kuyruk yöneticisi diğer adının adı.

MQC.MQOT_TOPIC tipi belirtilirse bu parametre yoksayılr.

Kuyruk bir model kuyruğuyrsa ve çözülen kuyruk yöneticisi adı bu kuyruk yöneticisi değilse, bir MQException yayınlanır.

topicString

topicString, *destinationName* konu nesnesindeki konu adıyla birleştirilir.

destinationName bir kuyruksa *topicString* yoksayılr.

ileti

Gönderilecek mesaj. İleti bir giriş/çıkış nesnesidir.

Yeniden bağlanabilir bir istemciye aşağıdaki neden kodları döndürülür:

- MQRC_CALL_INTERRUPTED kalıcı bir ileti üzerinde Put çağrısı gerçekleştirilirken bağlantı kesildiyse.
- MQRC_NONE kalıcı olmayan bir iletide Put çağrısı gerçekleştirilirken bağlantı başarılı olursa (bkz. [Application Recovery](#)).

putMessageSeçenekleri

Koymanın eylemlerini denetleyen seçenekler.

putMessageOptions ögesini atlarsanız, varsayılan bir *putMessageOptions* eşgörünümü yaratılır. *putMessageOptions* bir giriş/çıkış nesnesidir.

Yeniden bağlanabilir bir istemcide MQPMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

alternateUserKimliği

İletiyi bir kuyruğa yerleştirirken yetkilendirmeyi denetlemek için kullanılan diğer kullanıcı kimliğini belirtir.

putMessageOptions içinde MQC.MQOO_ALTERNATE_USER_AUTHORITY ayarını tanımlamazsanız *alternateUserId* değerini atlayabilirsiniz. MQC.MQOO_ALTERNATE_USER_AUTHORITY ayarını tanımlarsanız, *alternateUserId* da ayarlanmalıdır. *alternateUserId*, siz de MQC.MQOO_ALTERNATE_USER_AUTHORITY ayarını tanımlamadığınız sürece etkili olmaz.

Oluşturucular

```
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);
```

MQException ögesini atar.

Bir kuyruk yöneticisine bağlantı yaratır. İstemci bağlantısı yaratmak ya da sunucu bağlantısı yaratmak arasında seçim yapın.

Kuyruk yöneticisine bağlanmayı denerken, kuyruk yöneticisi üzerinde sorma (inq) yetkiniz olmalıdır. Sorma yetkisi olmadan, bağlantı girişimi başarısız olur.

Aşağıdaki koşullardan biri doğruysa bir istemci bağlantısı yaratılır:

1. *channel* ya da *connName*, oluşturucuda belirtilir.
2. *HostName*, *Port* ya da *Channel*, *properties* içinde belirtilir.
3. *MQEnvironment.HostName*, *MQEnvironment.Port* ya da *MQEnvironment.Channel* belirtilir.

Bağlantı özelliklerinin değerleri, gösterilen sırayla varsayılan olarak ayarlanır. Oluşturucunun *channel* ve *connName*, oluşturucunun özellik değerlerine göre önceliklidir. Oluşturucu özelliği değerleri, *MQEnvironment* özelliklerinden önceliklidir.

Anasistem adı, kanal adı ve kapı *MQEnvironment* sınıfında tanımlanır.

queueManagerAdı

Bağlantı kurulacak kuyruk yöneticisinin ya da kuyruk yöneticisi grubunun adı.

Parametreyi atlayın ya da boş bırakın ya da varsayılan bir kuyruk yöneticisi seçimi yapmak için boş bırakın. Bir sunucudaki varsayılan kuyruk yöneticisi bağlantısı, sunucudaki varsayılan kuyruk

yöneticisiyle bağlantıdır. İstemci bağlantısında varsayılan kuyruk yöneticisi bağlantısı, dinleyicinin bağlı olduğu kuyruk yöneticisine bağlanır.

seçenekler

MQCNO bağlantı seçeneklerini belirleyin. Değerler, yapılmakta olan bağlantı tipi için geçerli olmalıdır. Örneğin, bir istemci bağlantısı için aşağıdaki sunucu bağlantısı özelliklerini belirtirseniz bir MQException yayınlanır.

- MQC.MQCNO_FASTPATH_BINDING
- MQC.MQCNO_STANDARD_BINDING

Özellikler

Özellikler parametresi, MQEnvironment ; tarafından ayarlanan özellikleri geçersiz kılan bir dizi anahtar/değer çiftini alır. [“MQEnvironment özelliklerini geçersiz kıl” sayfa 1745](#)örneğine bakın. Aşağıdaki özellikler geçersiz kılınabilir:

- 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_CRYPTOHARDWARE_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

kanal

Sunucu bağlantı kanalının adı

connName

HostName (Kapı) biçiminde bağlantı adı.

CONNECTION_NAME_PROPERTY ' i kullanarak, MQQueueManager (String queueManagerName, Hashtable properties) oluşturucusuna bağımsız değişken olarak *anasistem adları* ve *kapı* listesini verebilirsiniz.

Örneğin:

```
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 MQQueue Manager("qmgrname",properties);
```

Bir bağlantı girişiminde bulunulduğunda, bağlantı adı listesi sırayla işlenir. İlk anasistem adı ve kapıya bağlanma girişimi başarısız olursa, ikinci öznitelik çiftine bağlanma girişiminde bulunilir. İstemci, başarılı bir bağlantı kuruluncaya ya da liste tükeninceye kadar bu işlemi yineler. Liste tükenirse, istemci uygulamasına uygun bir neden kodu ve tamamlanma kodu döndürülür.

Bağlantı adı için bir kapı numarası sağlanmadığında, varsayılan kapı (mqclient.ini içinde yapılandırılır) kullanılır.

Bağlantı Listesini Ayarla

Otomatik istemci yeniden bağlanma seçenekleri belirlendiğinde aşağıdaki yöntemleri kullanarak bağlantı listesini ayarlayabilirsiniz:

MQSERVER ile bağlantı listesini ayarla

Bağlantı listesini komut isteminden ayarlayabilirsiniz.

Komut isteminde aşağıdaki komutu belirleyin:

```
MQSERVER=SYSTEM.DEF.SVRCONN/TCP/Hostname1(Port1),Hostname2(Por2),Hostname3(Port3)
```

Örneğin:

```
MQSERVER=SYSTEM.DEF.SVRCONN/TCP/fred.mq.com(5266),nick.mq.com(6566),jack.mq.com(8413)
```

MQSERVER içinde bağlantıyı ayarlarsanız, uygulamada ayarlamayın.

Uygulamada bağlantı listesini ayarlarsanız, uygulama MQSERVER ortam değişkeninde ayarlanan her şeyin üzerine yazar.

Uygulama aracılığıyla bağlantı listesini ayarla

Anasistem adı ve kapı özelliklerini belirterek uygulamada bağlantı listesini ayarlayabilirsiniz.

```
String connName = "fred.mq.com(2344), nick.mq.com(3746), chris.mq.com(4288)";
MQQueueManager qm = new MQQueueManager("QM1", "TestChannel", connName);
```

Bağlantı listesini app.config ile ayarlayın

App.config, anahtar-değer çiftlerini belirttiğiniz bir XML dosyasıdır.

Bağlantı listesinde,

```
<app.Settings>
<add key="Connection1" value="Hostname1(Port1)"/>
<add key="Connection2" value="Hostname2(Port2)"/>
</app.Settings>
```

Örneğin:

```
<app.Settings>
<add key="Connection1" value="fred.mq.com(2966)"/>
<add key="Connection2" value="alex.mq.com(6533)"/>
</app.Settings>
```

app.config dosyasındaki bağlantı listesini doğrudan değiştirebilirsiniz.

Bağlantı listesini MQEnvironment ile ayarlayın

MQEnvironment aracılığıyla Bağlantı listesini ayarlamak için *ConnectionName* özelliğini kullanın.

```
MQEnvironment.ConnectionName = "fred.mq.com(4288),alex.mq.com(5211);
```

ConnectionName özelliği, MQEnvironment içinde ayarlanan anasistem adı ve kapı özelliklerinin üzerine yazar.

İstemci bağlantısı yarat

Aşağıdaki örnek, bir kuyruk yöneticisine istemci bağlantısının nasıl yaratılacağını göstermektedir. Yeni bir MQQueueManager Nesnesi yaratmadan önce MQEnvironment değişkenlerini ayarlayarak bir istemci bağlantısı oluşturabilirsiniz.

```
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");
```

Şekil 11. İstemci bağlantısı

MQEnvironment özelliklerini geçersiz kıl

Aşağıdaki örnekte, HASH çizelgesinde tanımlı kullanıcı kimliği ve parolasıyla bir kuyruk yöneticisinin nasıl yaratılacağı gösterilmektedir.

```
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);
}
```

Şekil 12. MQEnvironment özelliklerini geçersiz kılma

Yeniden bağlanabilir bir bağlantı oluştur

Aşağıdaki örnek, bir istemcinin kuyruk yöneticisine otomatik olarak nasıl yeniden bağlanacağını gösterir.

```
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);
```

Şekil 13. Bir istemcinin kuyruk yöneticisine otomatik olarak yeniden bağlanması

MQSubscription.NET sınıfı

Alıkonan yayınların aboneye gönderilmesini istemek için MQSubscription kullanın. MQSubscription, abonelik için açılan bir MQTopic nesnesinin özelliğidir.

Sınıf

```
System.Object
├── IBM.WMQ.MQBase
│   └── IBM.WMQ.MQBaseObject
│       └── IBM.WMQ.MQManagedObject
│           └── IBM.WMQ.MQSubscription
```

```
public class IBM.WMQ.MQSubscription extends IBM.WMQ.MQManagedObject;
```

- [“Özellikler” sayfa 1746](#)
- [“yöntemler” sayfa 1746](#)
- [“Oluşturucular” sayfa 1746](#)

Özellikler

MQManagedObject sınıfını kullanarak abonelik özelliklerine erişin; bkz. [“Özellikler” sayfa 1705](#).

yöntemler

MQManagedObject sınıfını kullanarak aboneliğe Inquire, Set ve Get yöntemlerine erişin; bkz. [“yöntemler” sayfa 1706](#).

```
public int RequestPublicationUpdate(int options);
```

MQExceptionöğesini atar.

Geçerli konu için güncellenmiş bir yayın isteyin. Kuyruk yöneticisinin konu için alıkonan yayınları varsa, bunlar aboneye gönderilir.

RequestPublicationUpdate' i aramadan önce, bir MQSubscription nesnesi elde etmek için abonelik için bir konu açın.

Genellikle, aboneliği MQC.MQSO_PUBLICATIONS_ON_REQUEST seçeneğiyle açın. Konu dizisinde joker karakter yoksa, bu çağrı sonucunda yalnızca bir yayın gönderilir. Konu dizesi joker karakter içeriyorsa, birçok yayın gönderilebilir. Bu yöntem, abonelik kuyruğuna gönderilen alıkonan yayınların sayısını döndürür. Özellikle kalıcı olmayan iletiler olduğunda, bu kadar çok yayın alınacağına garanti yoktur.

seçenekler

MQC.MQSRO_FAIL_IF QUIESCING

Kuyruk yöneticisi susturulmuş durumdaysa yöntem başarısız olur. z/OSsistemlerinde, bir CICS ya da IMS uygulaması için MQC.MQSRO_FAIL_IF QUIESCING , bağlantı susturulmuş durumdaysa yöntemin başarısız olmasına da zorlar.

MQC.MQSRO_NONE

Seçenek belirtilmedi.

Oluşturucular

Genel oluşturucu yok.

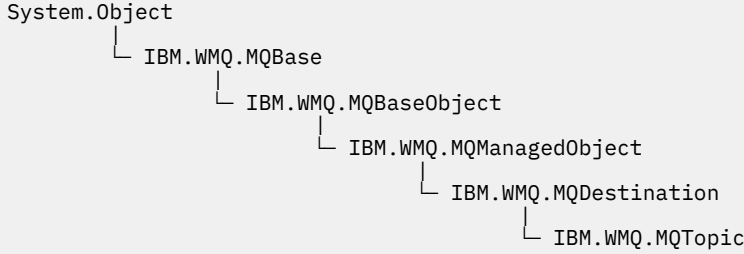
Abonelik için açılan bir MQTopic nesnesinin SubscriptionReference özelliğinde bir MQSubscription nesnesi döndürülür.

RequestPublicationUpdate yöntemini çağırın. MQSubscription , MQManagedObjectalt sınıfıdır. MQManagedObject' in özelliklerine ve yöntemlerine erişmek için başvuruyu kullanın.

MQTopic.NET sınıfı

Bir konuya ilişkin iletileri yayınlamak ya da abone olmak ya da bir konunun özniteliklerini sorgulamak ya da ayarlamak için MQTopic seçeneğini kullanın. Bir oluşturucu ya da MQQueueManager.AccessTopic yöntemini kullanarak yayınlamak ya da abone olmak için bir MQTopic nesnesi yaratın.

Sınıf



```
public class IBM.WMQ.MQTopic extends IBM.WMQ.MQDestination;
```

- [“Özellikler” sayfa 1747](#)
- [“yöntemler” sayfa 1747](#)
- [“Oluşturucular” sayfa 1749](#)

Özellikler

Özellikler alınırken MQException testi yapılıyor.

public Boolean IsDurable {get;}

Abonelik dayanıklıysa True , yoksa False döndüren salt okunur özellik. Konu yayınlanmak üzere açıldıysa, özellik yoksayılr ve her zaman Falsedöndürülür.

public Boolean IsManaged {get;};

Abonelik kuyruk yöneticisi tarafından yönetiliyorsa True döndüren salt okunur özellik ya da tersi durumda False . Konu yayınlanmak üzere açıldıysa, özellik yoksayılr ve her zaman False değerini döndürür.

public Boolean IsSubscribed {get;};

Konu abonelik için açıldıysa True ve konu yayınlanmak üzere açıldıysa False dizgisini döndüren salt okunur özellik.

public MQSubscription SubscriptionReference {get;};

Abonelik için açılan bir konu nesnesiyle ilişkili MQSubscription nesnesini döndüren salt okunur özellik. Kapatma seçeneklerini değiştirmek ya da nesne yöntemlerinden herhangi birini başlatmak istiyorsanız başvuru kullanılabilir.

public MQDestination UnmanagedDestinationReference {get;};

Yönetilmeyen bir abonelikle ilişkili MQQueue ögesini döndüren salt okunur özellik. Konu nesnesi yaratıldığında belirtilen hedeftir. Özellik, yayın için açılan ya da yönetilen bir abonelikle açılan konu nesnelere için boş değer döndürür.

yöntemler

public void Put(MQMessage message);

public void Put(MQMessage message, MQPutMessageOptions putMessageOptions);

MQException verir.

Konuya bir ileti yayınlar.

Koyma çağrısı tamamlandıktan sonra MQMessage nesnesinde yapılan değişiklikler, IBM MQ kuyruğundaki ya da yayın konusundaki gerçek iletiyi etkilemez.

Put , MQMessage nesnesinin MessageId ve CorrelationId özelliklerini günceller ve ileti verilerini temizlemez. Diğer Put ya da Get çağrıları, MQMessage nesnesindeki güncellenen bilgilere başurmaktadır. Örneğin, aşağıdaki kod parçacığına ilk ileti a ve ikinci abıçerir.

```
msg.WriteString("a");
q.Put(msg, pmo);
msg.WriteString("b");
q.Put(msg, pmo);
```

ileti

Gönderilecek ileti tanımlayıcı verilerini ve iletiyi içeren bir MQMessage nesnesi. İleti tanımlayıcısı, bu yöntemin sonucu olarak değiştirilebilir. Bu yöntem tamamlandıktan hemen sonra ileti tanımlayıcısındaki değerler, kuyruğa konan ya da konuya yayınlanan değerlerdir.

Yeniden bağlanabilir bir istemciye aşağıdaki neden kodları döndürülür:

- MQRC_CALL_INTERRUPTED kalıcı bir iletide Put çağrısı çalıştırılırken bağlantı kesilirse ve yeniden bağlantı başarılı olursa.
- MQRC_NONE kalıcı olmayan bir iletide Put çağrısı çalıştırılırken bağlantı başarılı olursa (bkz. [Application Recovery](#)).

putMessageSeçenekleri

Koyma işleminin eylemini denetleyen seçenekler.

putMessageOptions belirtilmezse, varsayılan MQPutMessageOptions eşgörünümü kullanılır.

Yeniden bağlanabilir bir istemcide MQPMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

Not: Kolaylık ve başarımlar için, kuyruğa tek bir ileti koymak istiyorsanız MQQueueManager .Put nesnesini kullanın. Bunun için bir MQQueue nesnesine sahip olun.

```
public void Get(MQMessage message);  
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions);  
public void Get(MQMessage message, MQGetMessageOptions getMessageOptions, int  
MaxMsgSize);
```

MQException verir.

Konudan bir ileti alır.

Bu yöntem, alma (get) işleminin varsayılan bir MQGetMessageOptions somut örneğini kullanır. Kullanılan ileti seçeneği: MQGMO_NOWAIT.

Alma işlemi başarısız olursa, MQMessage nesnesi değişmez. Başarılı olursa, MQMessage ileti tanımlayıcısı ve ileti verileri bölümleri, gelen iletideki ileti tanımlayıcısı ve ileti verileriyle değiştirilir.

Belirli bir MQQueueManager içinden IBM MQ ' e yapılan tüm çağrılar zamanuyumdur. Bu nedenle, beklemeyle alma işlemi gerçekleştirirseniz, aynı MQQueueManager ögesini kullanan diğer tüm iş parçacıklarının, Get çağrısı tamamlanincaya kadar başka IBM MQ çağrısı yapması engellenir. IBM MQ ' e aynı anda erişmek için birden çok iş parçacığına gereksinim duyarsanız, her iş parçacığı kendi MQQueueManager nesnesini oluşturmalıdır.

ileti

İleti tanımlayıcısı ve döndürülen ileti verilerini içerir. İleti tanımlayıcıdaki alanlardan bazıları giriş deęiştirgeleridir. MessageId ve CorrelationId giriş parametrelerinin gerektięi gibi ayarlandığından emin olmak önemlidir.

Yeniden bağlanabilir bir istemci, MQGM_SYNCPOINT altında alınan iletiler için başarılı bir yeniden bağlantıdan sonra MQRC_BACKED_OUT neden kodunu döndürür.

getMessageSeçenekleri

Alma işlemini denetleyen seçenekler.

MQC.MQGMO_CONVERT seçeneğinin kullanılması, tek baytlık karakter kodlarından çift baytlık kodlara dönüştürülürken MQC.MQRC_CONVERTED_STRING_TOO_BIG neden koduyla bir kural dışı durumla sonuçlanabilir. Bu durumda, ileti dönüştürme yapılmadan arabelleğe kopyalanır.

getMessageOptions belirtilmezse, kullanılan ileti seçeneği MQGMO_NOWAIT olur.

Yeniden bağlanabilir bir istemcide MQGMO_LOGICAL_ORDER seçeneğini kullanırsanız, MQRC_RECONNECT_INCOMPATIBLE neden kodu döndürülür.

MaxMsgBoyutu

Bu ileti nesnesinin alacağı en büyük ileti. Kuyruktaki ileti bu boyuttan büyükse, iki şeyden biri oluşur:

- MQGetMessageOptions nesnesinde MQGMO_ACCEPT_TRUNCATED_MSG işareti ayarlanırsa, ileti mümkün olduğunca çok ileti verileriyle doldurulur. MQCC_WARNING tamamlanma kodu ve MQRC_TRUNCATED_MSG_ACCEPTED neden koduyla bir kural dışı durum yayınlandı.
- MQGMO_ACCEPT_TRUNCATED_MSG işareti ayarlanmazsa, ileti kuyrukta kalır. MQCC_WARNING tamamlanma kodu ve MQRC_TRUNCATED_MSG_FAILED neden koduyla bir kural dışı durum yayınlandı.

MaxMsgSize belirtilmezse, iletinin tamamı alınır.

Oluşturucular

```
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);
```

queueManager ile ilgili bir konuya erişin.

MQTopic nesnelere, bazen konu nesnelere olarak adlandırılan yönetim konusu nesnelere yakından ilişkilidir. Girişte, *topicObject* bir yönetim konusu nesnesini gösterir. MQTopic oluşturucusu konu nesnesinden bir konu dizgisi alır ve konu adı oluşturmak için bunu *topicName* ile birleştirir. *topicObject* ya da *topicName* boş değerli olabilir. Konu adı, konu ağacına eşleştirilir ve *topicObject* içinde en yakın eşleşen yönetim konusu nesnesinin adı döndürülür.

MQTopic nesnesiyle ilişkili konular, iki konu dizgisini birleştirmenin sonucudur. İlk konu dizgisi, *topicObject* ile tanımlanan yönetim konusu nesnesi tarafından tanımlanır. İkinci konu dizgisi: *topicString*. MQTopic nesnesiyle ilişkilendirilen sonuçtaki konu dizgisi, genel arama karakterleri ekleyerek birden çok konuyu tanımlayabilir.

Konunun yayınlama ya da abone olma için açılıp açılmamasına bağlı olarak, konular üzerinde yayınlamak için MQTopic.Put yöntemlerini ya da konulara ilişkin yayınları almak için MQTopic.Get yöntemlerini kullanabilirsiniz. Aynı konuyu yayınlamak ve aynı konuya abone olmak istiyorsanız, konuya bir kez yayınlama, bir kez de abone olma üzere iki kez erişmeniz gerekir.

Bir MQDestination nesnesi sağlamadan abonelik için bir MQTopic nesnesi oluşturursanız, yönetilen abonelik kabul edilir. Bir kuyruğu MQDestination nesnesi olarak geçirirseniz, yönetilmeyen abonelik kabul edilir. Ayarladığınız abonelik seçeneklerinin yönetilen ya da yönetilmeyen abonelik tutarlı olduğundan emin olmanız gerekir.

queueManager

Bir konuya erişmek için kuyruk yöneticisi.

Hedef

destination bir MQQueue örneğidir. *destination* sağlanarak MQTopic , yönetilmeyen bir abonelik olarak açılır. Konuyla ilgili yayınlar, *destination*. olarak erişilen kuyruğa gönderilir.

topicName

Konu adının ikinci kısmı olan bir konu dizgisi. *topicName* , *topicObject* yönetim konusu nesnesinde tanımlanan konu dizgisiyle birleştirilmiştir. *topicName* değerini boş değere ayarlayabilirsiniz; bu durumda konu adı, *topicObject* içindeki konu dizisiyle tanımlanır.

topicObject

Girişte *topicObject* , konu adının ilk kısmını oluşturan konu dizgisini içeren konu nesnesinin adıdır. *topicObject* içindeki konu dizgisi *topicName* ile birleştirilmiştir. Konu dizgileri oluşturma kuralları, [Konu dizgilerini birleştirme](#) içinde tanımlanır.

Çıkışta, *topicObject* konu ağacında konu dizgisiyle tanımlanan konuya en yakın eşleşen yönetim konusu nesnesinin adını içerir.

openAs

Yayınlamak ya da abone olmak için konuya erişin. Parametre aşağıdaki seçeneklerden yalnızca birini içerebilir:

- MQC.MQTOPIC_OPEN_AS_SUBSCRIPTION
- MQC.MQTOPIC_OPEN_AS_PUBLICATION

seçenekler

Yayın ya da abonelik için konunun açılmasını denetleyen seçenekleri birleştirin. Aboneliğe ilişkin bir konuya erişmek için MQC.MQSO_* değişmezlerini ve yayına ilişkin bir konuya erişmek için MQC.MQOO_* değişmezlerini kullanın.

Birden çok seçenek gerekiyorsa, değerleri birlikte ekleyin ya da bit şeklinde OR işlecini kullanarak seçenek değerlerini birleştirin.

alternateUserKimliği

İşlemi tamamlamak için gereken yetkiyi denetlemek için kullanılan diğer kullanıcı kimliğini belirtin. Seçenekler parametresinde MQC.MQOO_ALTERNATE_USER_AUTHORITY ya da MQC.MQSO_ALTERNATE_USER_AUTHORITY belirlendiyse, *alternateUserId* belirtmeniz gerekir.

subscriptionName

MQC.MQSO_DURABLE ya da MQC.MQSO_ALTER seçenekleri sağlanırsa *subscriptionName* gereklidir. Her iki durumda da MQTopic , abonelik için örtük olarak açılır. MQC.MQSO_DURABLE ayarlandıysa, abonelik varsa ya da MQC.MQSO_ALTER ayarlandıysa ve abonelik yoksa bir özel durum oluşur.

Özellikler

Bir HASH çizelgesini kullanarak listelenen özel abonelik özelliklerinden herhangi birini ayarlayın. HASH çizelgesinde belirlenen girişler çıkış değerleriyle güncellenir. Girişler, çıkış değerlerini raporlamak için HASH çizelgesine eklenmez.

- 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);
```

Bu kuyruk yöneticisindeki bir konuya erişin.

MQTopic nesnelere, bazen konu nesnelere olarak adlandırılan yönetim konusu nesnelere yakından ilişkilidir. Girişte, topicObject bir yönetim konusu nesnesini gösterir. MQTopic oluşturucusu konu nesnesinden bir konu dizisi alır ve konu adı oluşturmak için bunu topicName ile birleştirir. topicObject ya da topicName boş değerli olabilir. Konu adı, konu ağacına eşleştirilir ve topicObject içinde en yakın eşleşen yönetim konusu nesnesinin adı döndürülür.

MQTopic nesnesiyle ilişkili konular, iki konu dizisini birleştirmenin sonucudur. İlk konu dizisi, topicObject ile tanımlanan yönetim konusu nesnesi tarafından tanımlanır. İkinci konu dizisi: topicString. MQTopic nesnesiyle ilişkilendirilen sonuçtaki konu dizisi, genel arama karakterleri ekleyerek birden çok konuyu tanımlayabilir.

Konunun yayınlama ya da abone olma için açılıp açılmamasına bağlı olarak, konular üzerinde yayınlamak için MQTopic .Put yöntemlerini ya da konulara ilişkin yayınları almak için MQTopic .Get yöntemlerini kullanabilirsiniz. Aynı konuyu yayınlamak ve aynı konuya abone olmak istiyorsanız, konuya bir kez yayınlama, bir kez de abone olma üzere iki kez erişmeniz gerekir.

Bir MQDestination nesnesi sağlamadan abonelik için bir MQTopic nesnesi oluşturursanız, yönetilen abonelik kabul edilir. Bir kuyruğu MQDestination nesnesi olarak geçirirseniz, yönetilmeyen abonelik kabul edilir. Ayarladığınız abonelik seçeneklerinin yönetilen ya da yönetilmeyen abonelik tutarlı olduğundan emin olmanız gerekir.

Hedef

destination bir MQQueue örneğidir. *destination* sağlanarak MQTopic , yönetilmeyen bir abonelik olarak açılır. Konuyla ilgili yayınlar, *destination* . olarak erişilen kuyruğa gönderilir.

topicName

Konu adının ikinci kısmı olan bir konu dizisi. *topicName* , *topicObject* yönetim konusu nesnesinde tanımlanan konu dizisiyle birleştirilmiştir. *topicName* değerini boş değere ayarlayabilirsiniz; bu durumda konu adı, *topicObject* içindeki konu dizisiyle tanımlanır.

topicObject

Girişte *topicObject* , konu adının ilk kısmını oluşturan konu dizisini içeren konu nesnesinin adıdır. *topicObject* içindeki konu dizisi *topicName* ile birleştirilmiştir. Konu dizileri oluşturma kuralları, [Konu dizilerini birleştirmeye](#) içinde tanımlanır.

Çıkışta, *topicObject* konu ağacında konu dizisiyle tanımlanan konuya en yakın eşleşen yönetim konusu nesnesinin adını içerir.

openAs

Yayınlamak ya da abone olmak için konuya erişin. Parametre aşağıdaki seçeneklerden yalnızca birini içerebilir:

- MQC.MQTOPIC_OPEN_AS_SUBSCRIPTION
- MQC.MQTOPIC_OPEN_AS_PUBLICATION

seçenekler

Yayın ya da abonelik için konunun açılmasını denetleyen seçenekleri birleştirin. Aboneliğe ilişkin bir konuya erişmek için MQC.MQSO_* değişmezlerini ve yayına ilişkin bir konuya erişmek için MQC.MQOO_* değişmezlerini kullanın.

Birden çok seçenek gerekiyorsa, değerleri birlikte ekleyin ya da bit şeklinde OR işlecini kullanarak seçenek değerlerini birleştirin.

alternateUserKimliği

İşlemi tamamlamak için gereken yetkiyi denetlemek için kullanılan diğer kullanıcı kimliğini belirtin. Seçenekler parametresinde MQC.MQOO_ALTERNATE_USER_AUTHORITY ya da MQC.MQSO_ALTERNATE_USER_AUTHORITY belirlendiye, *alternateUserId* belirtmeniz gerekir.

subscriptionName

MQC.MQSO_DURABLE ya da MQC.MQSO_ALTER seçenekleri sağlanırsa *subscriptionName* gereklidir. Her iki durumda da MQTopic, abonelik için örtük olarak açılır. MQC.MQSO_DURABLE ayarlandıysa, abonelik varsa ya da MQC.MQSO_ALTER ayarlandıysa ve abonelik yoksa bir özel durum oluşur.

Özellikler

Bir HASH çizelgesini kullanarak listelenen özel abonelik özelliklerinden herhangi birini ayarlayın. HASH çizelgesinde belirlenen girişler çıkış değerleriyle güncellenir. Girişler, çıkış değerlerini raporlamak için HASH çizelgesine eklenmez.

- 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

IMQObjectTrigger.NET arabirimi

`runmqdmn.NET` izleme programı tarafından iletilen iletileri işlemek için `IMQObjectTrigger` komutunu uygulayın.

Arabirim

```
public interface IBM.WMQMonitor.IMQObjectTrigger();
```

`runmqdmn` komutunda eşitleme noktası denetiminin belirtilip belirtilmediğine bağlı olarak, ileti `Execute` yöntemi dönmeye önce ya da döndükten sonra kuyruktan kaldırılır.

yöntemler

void Execute (MQQueueManager *queueManager*, MQQueue *queue*, MQMessage *message*, string *param*);

queueManager

İzlenmekte olan kuyruğu barındıran kuyruk yöneticisi.

kuyruk

Kuyruk izleniyor.

ileti

İleti kuyruktan okundu.

parametre

UserParameternesnesinden geçirilen veriler.

MQC.NET arabirimi

Sabit adı MQC . ile önceden düzeltir ve MQI deęişmezine bakın. MQC , MQI tarafından kullanılan tüm sabitleri tanımlar.

Arabirim

```
System.Object
└─ IBM.WMQ.MQC
```

```
public interface IBM.WMQ.MQC extends System.Object;
```

Örnek

```
MQQueue queue;
queue.closeOptions = MQC.MQCO_DELETE;
```

.NET uygulamaları için karakter kümesi tanıtıcıları

.NET IBM MQ iletilerini kodlamak için seçebileceğiniz karakter kümelerinin açıklamaları

Karakter kümesi	Açıklama
37	ibm037
437	ibm437 /PC Özgün
500	ibm500
819	iso-8859-1 / latin1 / ibm819
1200	Unicode
1208	UTF-8
273	ibm273
277	ibm277
278	ibm278
280	ibm280
284	ibm284
285	ibm285
297	ibm297
420	ibm420
424	ibm424
737	ibm737 /PC Yunanca

Karakter kümesi	Açıklama
775	ibm775 /PC Baltık
813	iso-8859-7 /Yunanca/ ibm813
838	ibm838
850	ibm850 /PC Latin 1
852	ibm852 /PC Latin 2
855	ibm855 /PC Kiril
856	ibm856
857	ibm857 /PC Türkçe
860	ibm860 /PC Portekizce
861	ibm861 /PC İzlandaca
862	ibm862 /PC İbranice
863	ibm863 /PC Kanada Fransızcası
864	ibm864 /PC Arapça
865	ibm865 /PC İskandinav
866	ibm866 /PC Rusça
868	ibm868
869	ibm869 /PC Modern Yunanca
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 /kiril/ ibm915
916	iso-8859-8 /hebrew/ ibm916
918	ibm918
920	iso-8859-9 / latin5 / ibm920
921	ibm921
922	ibm922
930	ibm930
932	PC Japonca
933	ibm933
935	ibm935
937	ibm937

Karakter kümesi	Açıklama
939	ibm939
942	ibm942
943	ibm943
948	ibm948
949	ibm949
950	ibm950 /Büyük 5 Geleneksel Çince
954	EUCJIS
964	ibm964 /CNS 11643 Geleneksel Çince
970	ibm970
1006	ibm1006
1025	ibm1025
1026	ibm1026
1089	iso-8859-6 /arabic/ ibm1089
1097	ibm1097
1098	ibm1098
1112	ibm1112
1122	ibm1122
1123	ibm1123
1124	ibm1124
1250	Windows Latin 2
1251	Windows Kiril
1252	Windows Latin 1
1253	Windows Yunanca
1254	Windows Türkçe
1255	Windows İbranice
1256	Windows Arapça
1257	Windows Baltık
1258	Windows Vietnam dili
1381	ibm1381
1383	ibm1383
2022	JIS
5601	ksc-5601 Korece
33722	ibm33722

IBM MQ C++ sınıfları

IBM MQ C++ sınıfları IBM MQ Message Queue Interface (MQI) olanağını kapsıyor. Bu sınıfların tümünü kapsayan tek bir C++ üstbilgi dosyası (**imqi.hpp**) vardır.

Her sınıf için aşağıdaki bilgiler gösterilir:

Sınıf sıradüzeni çizgesi

Sınıfı, üst sınıflarıyla (varsa) edinme ilişkisinde gösteren bir sınıf çizgesi.

Diğer ilgili sınıflar

Üst sınıflar gibi diğer ilgili sınıflara ve yöntem imzalarında kullanılan nesnelerin sınıflarına belge bağlantıları.

Nesne öznitelikleri

Sınıfın öznitelikleri. Bunlar, üst sınıflar için tanımlanan özniteliklere ek olarak gösterilir. Birçok öznitelik IBM MQ veri yapısı üyelerini yansıtır (bkz. [“C++ ve MQI çapraz başvurusu” sayfa 1757](#)). Ayrıntılı açıklamalar için bkz. [“Nesnelerin öznitelikleri” sayfa 789](#).

Oluşturucular

Sınıfın bir nesnesini yaratmak için kullanılan özel yöntemlerin imzaları.

Nesne yöntemleri (genel)

İşlem için sınıfın bir örneğini gerektiren ve kullanım kısıtlaması olmayan yöntemlerin imzaları.

Geçerli olduğu durumlarda, aşağıdaki bilgiler de gösterilir:

Sınıf yöntemleri (genel)

İşlem için sınıfın bir eşgörünümünü gerektirmeyen ve kullanım kısıtlaması olmayan yöntemlerin imzaları.

Aşırı yüklenmiş (üst sınıf) yöntemler

Üst sınıflarda tanımlanan bu sanal yöntemlerin imzaları, ancak bu sınıf için farklı, çok şekilli, davranış sergiliyor.

Nesne yöntemleri (korunmalı)

İşletimleri için sınıfın somut örneğini gerektiren ve türetilmiş sınıfların somutlamaları tarafından kullanılmak üzere ayrılmış yöntemlerin imzaları. Bu kısım, sınıf kullanıcılarının aksine, yalnızca sınıf yazarlarının ilgi alanıdır.

Nesne verileri (korunan)

Türetilmiş sınıfların somutlamaları tarafından kullanılacak nesne eşgörünümü verilerine ilişkin somutlama ayrıntıları. Bu kısım, sınıf kullanıcılarının aksine, yalnızca sınıf yazarlarının ilgi alanıdır.

Neden kodları

Başarısız olan yöntemlerden beklenebilecek MQRC_ * değerleri (bkz. [API tamamlama ve neden kodları](#)). Bir sınıfın nesnesi için oluşabilecek neden kodlarının ayrıntılı bir listesi için üst sınıf belgelerine bakın. Bir sınıfa ilişkin neden kodlarının belgelenmiş listesi, üst sınıflara ilişkin neden kodlarını içermez.

Not:

1. Bu sınıfların nesnelere iş parçacığı korunmalı değildir. Bu, en iyi performansı sağlar, ancak birden fazla iş parçacığından herhangi bir nesneye erişmemeye dikkat edin.
2. Çok iş parçacıklı bir program için, her iş parçacığı için ayrı bir ImqQueueManager nesnesi kullanılması önerilir. Her yönetici nesnesinin, farklı iş parçacıklarındaki nesnelerin birbirinden yalıtıldığından emin olarak, diğer nesnelere oluşan bağımsız bir derlemi olmalıdır.

Sınıflar şunlardır:

- [“ImqAuthenticationKayıt C++ sınıfı” sayfa 1772](#)
- [“ImqBinary C++ sınıfı” sayfa 1774](#)
- [“ImqCache C++ sınıfı” sayfa 1776](#)
- [“ImqChannel C++ sınıfı” sayfa 1779](#)
- [“ImqCICSBridgeHeader C++ sınıfı” sayfa 1784](#)

- [“ImqDeadLetterHeader C++ sınıfı” sayfa 1790](#)
- [“ImqDistributionListe C++ sınıfı” sayfa 1793](#)
- [“ImqError C++ sınıfı” sayfa 1794](#)
- [“ImqGetMessageOptions C++ sınıfı” sayfa 1795](#)
- [“ImqHeader C++ sınıfı” sayfa 1799](#)
- [“ImqIMSBridgeHeader C++ sınıfı” sayfa 1800](#)
- [“ImqItem C++ sınıfı” sayfa 1803](#)
- [“ImqMessage C++ sınıfı” sayfa 1805](#)
- [“ImqMessageİzleyici C++ sınıfı” sayfa 1811](#)
- [“ImqNamelist C++ sınıfı” sayfa 1814](#)
- [“ImqObject C++ sınıfı” sayfa 1816](#)
- [“ImqProcess C++ sınıfı” sayfa 1821](#)
- [“ImqPutMessageOptions C++ sınıfı” sayfa 1823](#)
- [“ImqQueue C++ sınıfı” sayfa 1825](#)
- [“ImqQueueManager C++ sınıfı” sayfa 1835](#)
- [“ImqReferenceHeader C++ sınıfı” sayfa 1851](#)
- [“ImqString C++ sınıfı” sayfa 1854](#)
- [“ImqTrigger C++ sınıfı” sayfa 1859](#)
- [“ImqWorkÜstbilgi C++ sınıfı” sayfa 1862](#)

C++ ve MQI çapraz başvurusu

Bu konu derlemi, C++ ile MQI ile ilgili bilgileri içerir.

[“MQI 'da kullanılan veri tipleri” sayfa 234](#) ile birlikte bu bilgileri okuyun.

Bu çizelge, MQI veri yapılarını C++ sınıflarıyla ilişkilendirir ve dosyaları içerir. Aşağıdaki konular, her C++ sınıfına ilişkin çapraz başvuru bilgilerini gösterir. Bu çapraz başvurular, temel IBM MQ yordamsal arabirimlerinin kullanımıyla ilgilidir. ImqBinary, ImqDistributionList ve ImqString sınıflarının bu kategoriye giren ve dışlanan öznelikleri yok.

<i>Çizelge 845. Veri yapısı, sınıf ve include-file çapraz başvurusu</i>		
Veri yapısı	Sınıf	Dosya ekle
MQAIR	ImqAuthenticationKaydı	imqair.hpp
	ImqBinary	imqbin.hpp
	ImqCache	imqcac.hpp
MQCD	ImqChannel	imqchl.hpp
MQCIH	İqCICSBridgeHeader	imqcih.hpp
MQDLH	ImqDeadLetterHeader	imqdlh.hpp
MQOR	ImqDistributionListesi	imqdst.hpp
	ImqError	imqerr.hpp
MQGMO	ImqGetMessageOptions	imqgmo.hpp
	ImqHeader	imqhdr.hpp
MQIIH	İqIMSBridgeHeader	imqiih.hpp
	ImqItem	imqitm.hpp

<i>Çizelge 845. Veri yapısı, sınıf ve include-file çapraz başvurusu (devamı var)</i>		
Veri yapısı	Sınıf	Dosya ekle
Mqmd	ImqMessage	imqmsg.hpp
	ImqMessageİzleyici	imqmtr.hpp
	ImqNamelist	imqnml.hpp
MQOD, MQRR	ImqObject	imqobj.hpp
MQPMO, MQPMR, MQRR	ImqPutMessageOptions	imqpmo.hpp
	ImqProcess	imqpro.hpp
	ImqQueue	imqqueue.hpp
MQBO, MQCNO, MQCSP	ImqQueueYöneticisi	imqmgr.hpp
MQRMH	ImqReferenceÜstbilgisi	imqrfh.hpp
	ImqString	imqstr.hpp
MQTM	ImqTrigger	imqtrg.hpp
MQTMCM		
MQTMCM2	ImqTrigger	imqtrg.hpp
MQXQH		
MQWIH	ImqWorkÜstbilgisi	imqwih.hpp

ImqAuthenticationKayıt çapraz başvurusu

ImqAuthenticationRecord C++ sınıfı için özniteliklerin, veri yapılarının, alanların ve çağrılarının çapraz başvurusu.

<i>Çizelge 846. Öznitelikler, veri yapıları, alanlar ve çağrılar</i>			
Öznitelik	Veri yapısı	Alan	Ara
Bağlantı adı	MQAIR	AuthInfoConnName	MQCONN
parola	MQAIR	LDAPPassword Parolası	MQCONN
tip	MQAIR	AuthInfoTipi	MQCONN
kullanıcı adı	MQAIR	LDAPUserNamePtr	MQCONN
	MQAIR	LDAPUserNameKayması	MQCONN
	MQAIR	LDAPUserNameUzunluğu	MQCONN

ImqCache çapraz başvurusu

ImqCache C++ sınıfı için özniteliklerin ve çağrılarının çapraz başvurusu.

<i>Çizelge 847. Öznitelikler ve çağrılar</i>	
Öznitelik	Ara
otomatik arabellek	MQGet
arabellek uzunluğu	MQGet
arabellek göstergesi	MQGET, MQPUT

<i>Çizelge 847. Öznitelikler ve çağrılar (devamı var)</i>	
Öznitelik	Ara
veri uzunluğu	MQGet
veri görelî konumu	MQGet
veri işaretçisi	MQGet
ileti uzunluğu	MQGET, MQPUT

ImqChannel çapraz başvurusu

ImqChannel C++ sınıfı için özniteliklere, veri yapılarına, alanlara ve çağrılara ilişkin çapraz başvuru.

<i>Çizelge 848. Öznitelikler, veri yapıları, alanlar ve çağrılar</i>			
Öznitelik	Veri yapısı	Alan	Ara
toplu kalp atışım	MQCD	BatchHeartbeat	MQCONN
Kanal Adı	MQCD	ChannelName	MQCONN
Bağlantı adı	MQCD	ConnectionName	MQCONN
	MQCD	ShortConnectionAdı	MQCONN
üstbilgi sıkıştırması	MQCD	HdrCompListesi	MQCONN
kalp atığı aralığı	MQCD	HeartbeatInterval	MQCONN
canlı tutma aralığı	MQCD	KeepAliveAralığı	MQCONN
Yerel adres	MQCD	LocalAddress	MQCONN
ileti uzunluğu üst sınırı	MQCD	MaxMsgUzunluğu	MQCONN
ileti sıkıştırma	MQCD	MsgCompListesi	MQCONN
kip adı	MQCD	ModeName	MQCONN
parola	MQCD	Parola	MQCONN
alma çıkış sayısı	MQCD		MQCONN
çıkış adlarını al	MQCD	ReceiveExit	MQCONN
	MQCD	ReceiveExitsTanımlı	MQCONN
	MQCD	ReceiveExitPtr	MQCONN
kullanıcı verilerini al	MQCD	ReceiveUserVerileri	MQCONN
	MQCD	ReceiveUserDataPtr	MQCONN
güvenlik çıkışı adı	MQCD	SecurityExit	MQCONN
güvenlik kullanıcısı verileri	MQCD	SecurityUserVerileri	MQCONN
çıkış gönderme sayısı	MQCD		MQCONN
çıkış adlarını gönder	MQCD	SendExit	MQCONN
	MQCD	SendExitsTanımlı	MQCONN
	MQCD	SendExitPtr	MQCONN
kullanıcı verileri gönder	MQCD	SendUserVerileri	MQCONN
	MQCD	SendUserDataPtr	MQCONN

Çizelge 848. Öznitelikler, veri yapıları, alanlar ve çağrılar (devamı var)

Öznitelik	Veri yapısı	Alan	Ara
SSL CipherSpec	MQCD	sslCipherBelirtimi	MQCONN
SSL istemcisi kimlik denetimi tipi	MQCD	sslClientKimlik Doğrulaması	MQCONN
SSL eş adı	MQCD	sslPeerAdı	MQCONN
hareket işleme programı adı	MQCD	TpName	MQCONN
iletim tipi	MQCD	TransportType	MQCONN
kullanıcı kimliği	MQCD	UserIdentifier	MQCONN

ImqCICSBridgeHeader çapraz başvurusu

ImqCICSBridgeHeader C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

Çizelge 849. Özniteliklerin, veri yapılarının ve alanların eşlenmesi

Öznitelik	Veri yapısı	Alan
köprü olağandışı bitiş kodu	MQCIH	AbendCode
ADS tanımlayıcısı	MQCIH	AdsDescriptor
uyarı tanıtıcısı	MQCIH	AttentionId
kimlik doğrulayıcı	MQCIH	Kimlik doğrulayıcı
köprü tamamlama kodu	MQCIH	BridgeCompletionKodu
köprü hatası görelî konumu	MQCIH	ErrorOffset
köprü neden kodu	MQCIH	BridgeReason
köprü iptal kodu	MQCIH	CancelCode
etkileşimli görev	MQCIH	ConversationalTask
imleç konumu	MQCIH	CursorPosition
olanak simgesi	MQCIH	Olanak
tesis tutma süresi	MQCIH	FacilityKeepSaati
benzer tesis	MQCIH	FacilityLike
işlev	MQCIH	İşlev
bekleme aralığını al	MQCIH	GetWaitAralığı
Bağlantı tipi	MQCIH	LinkType
sonraki işlem tanıtıcısı	MQCIH	NextTransactionTanıtıcısı
çıkış verileri uzunluğu	MQCIH	OutputDataUzunluğu
yanıtlama biçimi	MQCIH	ReplyToBiçimi
köprü dönüş kodu	MQCIH	ReturnCode
başlatma kodu	MQCIH	StartCode
görev bitiş durumu	MQCIH	TaskEndDurumu
işlem tanıtıcısı	MQCIH	TransactionId
uow denetimi	MQCIH	UowControl

<i>Çizelge 849. Özniteliklerin, veri yapılarının ve alanların eşlenmesi (devamı var)</i>		
Öznitelik	Veri yapısı	Alan
sürüm	MQCIH	Sürüm

ImqDeadLetterHeader çapraz başvuru

ImqDeadLetterHeader C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

<i>Çizelge 850. Özniteliklerin, veri yapılarının ve alanların eşlenmesi</i>		
Öznitelik	Veri yapısı	Alan
çıkamaz mektup neden kodu	MQDLH	Neden
Hedef kuyruk yöneticisi adı	MQDLH	DestQMgrAdı
hedef kuyruk adı	MQDLH	DestQName
Koyma Uygulaması Adı	MQDLH	PutApplAdı
Koyma Uygulaması Tipi	MQDLH	PutApplTipi
koyma tarihi	MQDLH	PutDate
koyma süresi	MQDLH	PutTime

ImqError çapraz başvurusu

ImqError C++ sınıfı için özniteliklerin ve çağrılarının çapraz başvurusu.

<i>Çizelge 851. Öznitelikler ve çağrılar</i>	
Öznitelik	Ara
tamamlama kodu	MQBACK, MQBEGIN, MQCLOSE, MQCMIT, MQCONN, MQCONNX, MQDISC, MQGET, MQINQ, MQOPEN, MQPUT, MQSET
Neden Kodu	MQBACK, MQBEGIN, MQCLOSE, MQCMIT, MQCONN, MQCONNX, MQDISC, MQGET, MQINQ, MQOPEN, MQPUT, MQSET

ImqGetMessageOptions çapraz başvurusu

ImqGetMessageOptions C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

<i>Çizelge 852. Özniteliklerin, veri yapılarının ve alanların eşlenmesi</i>		
Öznitelik	Veri yapısı	Alan
grup durumu	MQGMO	GroupStatus
eşleşme seçenekleri	MQGMO	MatchOptions
ileti simgesi	MQGMO	MessageToken
seçenekler	MQGMO	Seçenekler
çözülmüş kuyruk adı	MQGMO	ResolvedQName
döndürülen uzunluk	MQGMO	ReturnedLength
kesimlere ayırma	MQGMO	Bölümlleme
bölüm durumu	MQGMO	SegmentStatus
	MQGMO	Signal1

<i>Çizelge 852. Özniteliklerin, veri yapılarının ve alanların eşlenmesi (devamı var)</i>		
Öznitelik	Veri yapısı	Alan
	MQGM0	Signal2
eşitleme noktası katılımı	MQGM0	Seçenekler
Bekleme Aralığı	MQGM0	WaitInterval

ImqHeader çapraz başvurusu

ImqHeader C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

<i>Çizelge 853. Özniteliklerin, veri yapılarının ve alanların eşlenmesi</i>		
Öznitelik	Veri yapısı	Alan
karakter kümesi	MQDLH, MQIIH	CodedCharSetId
Kodlama	MQDLH, MQIIH	Kodlama
biçim	MQDLH, MQIIH	Biçim
üstbilgi işaretleri	MQIIH, MQRMH	İşaretler

ImqIMSBridgeHeader çapraz başvurusu

ImqAuthenticationRecord C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

<i>Çizelge 854. Özniteliklerin, veri yapılarının ve alanların eşlenmesi</i>		
Öznitelik	Veri yapısı	Alan
kimlik doğrulayıcı	MQIIH	Kimlik doğrulayıcı
kesinleştirme kipi	MQIIH	CommitMode
mantıksal uçbirim geçersiz kılma	MQIIH	LTermOverride
ileti biçimi hizmetleri eşlemi adı	MQIIH	MFSMapName
yanıtlama biçimi	MQIIH	ReplyToBiçimi
güvenlik kapsamı	MQIIH	SecurityScope
hareket eşgörünümü tanıtıcısı	MQIIH	TranInstanceTanıtıcısı
hareket durumu	MQIIH	TranState

ImqItem çapraz başvurusu

ImqItem C++ sınıfı için özniteliklerin ve çağrılarının çapraz başvurusu.

<i>Çizelge 855. Öznitelikler ve çağrılar</i>	
Öznitelik	Ara
yapı tanıtıcısı	MQGet

ImqMessage çapraz başvurusu

ImqMessage C++ sınıfı için özniteliklerin, veri yapılarının, alanların ve çağrılarının çapraz başvurusu.

<i>Çizelge 856. Öznitelikler, veri yapıları, alanlar ve çağrılar</i>			
Öznitelik	Veri yapısı	Alan	Ara
Uygulama Tnt Verisi	Mqmd	ApplIdentityVerileri	
Uygulama Kökeni Verisi	Mqmd	ApplOriginVerileri	
Geriletme Sayısı	Mqmd	BackoutCount	
karakter kümesi	Mqmd	CodedCharSetId	
Kodlama	Mqmd	Kodlama	
Son kullanma tarihi	Mqmd	Son kullanma tarihi	
biçim	Mqmd	Biçim	
İleti İşaretleri	Mqmd	MsgFlags	
ileti tipi	Mqmd	MsgType	
offset	Mqmd	Görelî Konum	
Özgün Uzunluk	Mqmd	OriginalLength	
Kalıcılık	Mqmd	Kalıcılık	
öncelik	Mqmd	Öncelik	
Koyma Uygulaması Adı	Mqmd	PutApplAdı	
Koyma Uygulaması Tipi	Mqmd	PutApplTipi	
koyma tarihi	Mqmd	PutDate	
koyma süresi	Mqmd	PutTime	
yanıtın gönderileceği kuyruk yöneticisi adı	Mqmd	ReplyToQMgr	
yanıtın gönderileceği kuyruk adı	Mqmd	ReplyToQ	
rapor	Mqmd	Rapor	
Sıra Numarası	Mqmd	MsgSeqNumarası	
toplam ileti uzunluğu		DataLength	MQGet
kullanıcı kimliği	Mqmd	UserIdentifer	

ImqMessageİzleyici çapraz başvurusu

ImqMessageTracker C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

<i>Çizelge 857. Özniteliklerin, veri yapılarının ve alanların eşlenmesi</i>		
Öznitelik	Veri yapısı	Alan
muhasebe simgesi	Mqmd	AccountingToken
İlinti Tanıtıcısı	Mqmd	CorrelId
Geribildirim	Mqmd	Geri bildirim
grup tanıtıcısı	Mqmd	GroupId
İleti Tnt	Mqmd	MsgId

ImqNamelist çapraz başvurusu

Öznitelikler, sorgular ve ImqNamelist C++ sınıfı çağrılar için çapraz başvuru.

<i>Çizelge 858. Öznitelikler, sorgular ve çağrılar</i>		
Öznitelik	Sorgu	Ara
ad sayısı	MQIA_AD_COUNT	MQINQ
ad listesi adı	MQCA_NAMELIST_ADı	MQINQ

ImqObject çapraz başvurusu

ImqObject C++ sınıfı için öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar için çapraz başvuru.

<i>Çizelge 859. Öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar</i>				
Öznitelik	Veri yapısı	Alan	Sorgu	Ara
değişiklik tarihi			MQCA_ALTERATION_DATE	MQINQ
değişiklik zamanı			MQCA_ALTERATION_TIME	MQINQ
diğer kullanıcı kimliği	MQOD	AlternateUserKimli ği		
alternatif güvenlik tanıtıcısı				
seçenekleri kapat				MQCLOSE
açıklama			MQCA_Q_DESC, MQCA_Q_MGR_DESC, MQCA_PROCESS_DESC	MQINQ
Ad	MQOD	ObjectName	MQCA_Q_MGR_NAME, MQCQ_Q_NAME, MQCA_PROCESS_NAME	MQINQ
seçenekleri aç				MQOPEN
açık durum				MQOPEN, MQCLOSE
kuyruk yöneticisi tanıtıcısı	kuyruk yöneticisi tanıtıcısı		MQCA_Q_MGR_IDENTIFIER	MQINQ

ImqProcess çapraz başvurusu

ImqAuthenticationRecord C++ sınıfı için özniteliklere, sorgulara ve çağrılara çapraz başvuru.

<i>Çizelge 860. Öznitelikler, sorgular ve çağrılar</i>		
Öznitelik	Sorgu	Ara
Uygulama tanıtıcısı	MQCA_APPL_ID	MQINQ
Uygulama Tipi	MQIA_APPL_TYPE	MQINQ
ortam verileri	MQCA_ENV_DATA	MQINQ
Kullanıcı verileri	MQCA_USER_DATA	MQINQ

ImqPutMessageOptions çapraz başvurusu

ImqAuthenticationRecord C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

Çizelge 861. Özniteliklerin, veri yapılarının ve alanların eşlenmesi

Öznitelik	Veri yapısı	Alan
bağlam başvurusu	MQPMO	Bağlam
	MQPMO	InvalidDestSayısı
	MQPMO	KnownDestSayısı
seçenekler	MQPMO	Seçenekler
kayıt alanları	MQPMO	PutMsgRecFields
çözülmüş kuyruk yöneticisi adı	MQPMO	ResolvedQMgrAdı
çözülmüş kuyruk adı	MQPMO	ResolvedQName
	MQPMO	Zamanaşımı
	MQPMO	UnknownDestSayısı
eşitleme noktası katılımı	MQPMO	Seçenekler

ImqQueue çapraz başvurusu

ImqQueue C++ sınıfına ilişkin öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar için çapraz başvuru.

Çizelge 862. ImqQueue çapraz başvurusu

Öznitelik	Veri yapısı	Alan	Sorgu	Ara
geriletme yeniden kuyruğa alma adı			MQCA_BACKOUT_REQ_Q_NAME	MQINQ
Geriletme eşiği			MQIA_BACKOUT_THRESHOLD	MQINQ
temel kuyruk adı			MQCA_BASE_Q_NAME	MQINQ
Küme adı			MQCA_CLUSTER_NAME	MQINQ
küme ad listesi adı			MQCA_CLUSTER_NAMELIST	MQINQ
küme iş yükü sırası			MQIA_CLWL_Q_RANK	MQINQ
küme iş yükü önceliği			MQIA_CLWL_Q_PRIORITY	MQINQ
küme iş yükü kullanım kuyruğu			MQIA_CLWL_USEQ	MQINQ
yaratma tarihi			MQCA_CREATION_DATE	MQINQ
Oluşturulma zamanı			MQCA_CREATION_TIME	MQINQ
yürürlükteki derinlik			MQIA_CURRENT_Q_DEPTH	MQINQ
varsayılan bağ tanımlama			MQIA_DEF_BIND	MQINQ
varsayılan giriş açma seçeneği			MQIA_DEF_INPUT_OPEN_OPTION	MQINQ
varsayılan kalıcılık			MQIA_DEF_PERSISTENCE	MQINQ

Çizelge 862. ImqQueue çapraz başvurusu (devamı var)				
Öznitelik	Veri yapısı	Alan	Sorgu	Ara
varsayılan öncelik			MQIA_DEF_PRIORITY	MQINQ
Tanımlama tipi			MQIA_DEFINITION_TYPE	MQINQ
derinlik yüksek olayı			MQIA_Q_DEPTH_HIGH_EVENT	MQINQ
derinlik yüksek sınırı			MQIA_Q_DEPTH_HIGH_LIMIT	MQINQ
derinlik düşük olayı			MQIA_Q_DEPTH_LOW_EVENT	MQINQ
derinlik alt sınırı			MQIA_Q_DEPTH_LOW_LIMIT	MQINQ
derinlik üst sınırı olayı			MQIA_Q_DEPTH_MAX_LIMIT	MQINQ
dağıtım listeleri			MQIA_DIST_LIST	MQINQ, MQSET
dinamik kuyruk adı	MQOD	DynamicQName		
sertleşin geri çekilin			MQIA_HARDEN_GET_BACKOUT	MQINQ
dizin tipi			MQIA_INDEX_TYPE	MQINQ
alma engelleme			MQIA_INHIBIT_GET	MQINQ, MQSET
koymayı engelle			MQIA_INHIBIT_PUT	MQINQ, MQSET
başlatma kuyruğu adı			MQCA_INITIATION_Q_NAME	MQINQ
derinlik üst sınırı			MQIA_MAX_Q_DEPTH	MQINQ
ileti uzunluğu üst sınırı			MQIA_MAX_MSG_LENGTH	MQINQ
ileti teslim sırası			MQIA_MSG_DELIVERY_SEQUENCE	MQINQ
sonraki dağıtılmış kuyruk				
kalıcı olmayan ileti sınıfı			MQIA_NPM_CLASS	MQINQ
açık giriş sayısı			MQIA_OPEN_INPUT_COUNT	MQINQ
açık çıkış sayısı			MQIA_OPEN_OUTPUT_COUNT	MQINQ
önceki dağıtılmış kuyruk				
İşlem adı			MQCA_PROCESS_NAME	MQINQ
kuyruk muhasebesi			MQIA_ACCOUNTING_Q	MQINQ
Kuyruk yöneticisi adı	MQOD	ObjectQMgrAdı		
kuyruk izleme			MQIA_MONITORING_Q	MQINQ
kuyruk istatistikleri			MQIA_STATISTICS_Q	MQINQ
kuyruk tipi			MQIA_Q_TYPE	MQINQ

Çizelge 862. ImqQueue çapraz başvurusu (devamı var)				
Öznitelik	Veri yapısı	Alan	Sorgu	Ara
uzak kuyruk yöneticisi adı			MQCA_REMOTE_Q_MGR_NAME	MQINQ
uzak kuyruk adı			MQCA_REMOTE_Q_NAME	MQINQ
çözülmüş kuyruk yöneticisi adı	MQOD	ResolvedQMgrAdı		
çözülmüş kuyruk adı	MQOD	ResolvedQName		
alikoyma aralığı			MQIA_RETENTION_INTERVAL	MQINQ
kapsam			MQIA_SCOPE	MQINQ
hizmet aralığı			MQIA_Q_SERVICE_INTERVAL	MQINQ
hizmet aralığı olayı			MQIA_Q_SERVICE_INTERVAL_EVENT	MQINQ
paylaşılabilir			MQIA_SHAREABILITY	MQINQ
depolama sınıfı			MQCA_STORAGE_CLASS	MQINQ
iletim kuyruğu adı			MQCA_XMIT_Q_NAME	MQINQ
tetikleme denetimi			MQIA_TRIGGER_CONTROL	MQINQ, MQSET
tetikleyici verileri			MQCA_TRIGGER_DATA	MQINQ, MQSET
tetikleyici derinliği			MQIA_TRIGGER_DEPTH	MQINQ, MQSET
tetikleyici ileti önceliği			MQIA_TRIGGER_MSG_PRIORITY	MQINQ, MQSET
Tetikleme Tipi			MQIA_TRIGGER_TYPE	MQINQ, MQSET
kullanıma			MQIA_USAGE	MQINQ

ImqQueueManager çapraz başvurusu

ImqQueueManager C++ sınıfına ilişkin öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar için çapraz başvuru.

Çizelge 863. Öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar				
Öznitelik	Veri yapısı	Alan	Sorgu	Ara
muhasabe bağlantıları geçersiz kılma			MQIA_ACCOUNTING_CONN_OVERRIDE	MQINQ
hesaplama aralığı			MQIA_ACCOUNTING_INTERVAL	MQINQ
etkinlik kaydı			MQIA_ACTIVITY_RECORDING	MQINQ
yeni mca denetimini kabul et			MQIA_ADOPTNEWMCA_CHECK	MQINQ

Çizelge 863. Öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar (devamı var)

Öznitelik	Veri yapısı	Alan	Sorgu	Ara
yeni mca tipini benimse			MQIA_ADOPTNEWMCA_TYPE	MQINQ
Kimlik Doğrulama Tipi	MQCSP	AuthenticationType		MQCONN
yetki olayı			MQIA_AUTHORITY_EVENT	MQINQ
başlangıç seçenekleri	MQBO	Seçenekler		MQBEGIN
köprü olayı			MQIA_BRIDGE_EVENT	MQINQ
kanal otomatik tanımı			MQIA_CHANNEL_AUTO_DEF	MQINQ
kanal otomatik tanımlama olayı			MQIA_CHANNEL_AUTO_EVENT	MQIA
kanal otomatik tanımlama çıkışı			MQIA_CHANNEL_AUTO_EXIT	MQIA
kanal olayı			MQIA_CHANNEL_EVENT	MQINQ
kanal başlatıcı bağdaştırıcıları			MQIA_CHINIT_ADAPTERS	MQINQ
kanal başlatıcı denetimi			MQIA_CHINIT_CONTROL	MQINQ
kanal başlatıcı dağıtıcıları			MQIA_CHINIT_DISPATCHERS	MQINQ
kanal başlatıcı izleme otomatik başlatma			MQIA_CHINIT_TRACE_AUTO_START	MQINQ
kanal başlatıcı izleme çizelgesi büyüklüğü			MQIA_CHINIT_TRACE_TABLE_SIZE	MQINQ
kanal izleme			MQIA_MONITORING_CHANNEL	MQINQ
kanal başvurusu	MQCD	ChannelType		MQCONN
kanal istatistikleri			MQIA_STATISTICS_CHANNEL	MQINQ
karakter kümesi			MQIA_CODED_CHAR_SET_ID	MQINQ
küme gönderen izlemesi			MQIA_MONITORING_AUTO_CLUSSDR	MQINQ
küme gönderen istatistikleri			MQIA_STATISTICS_AUTO_CLUSSDR	MQINQ
küme iş yükü verileri			MQCA_CLUSTER_WORKLOAD_DATA	MQINQ
küme iş yükü çıkışı			MQCA_CLUSTER_WORKLOAD_EXIT	MQINQ
küme iş yükü uzunluğu			MQIA_CLUSTER_WORKLOAD_LENGTH	MQINQ

Çizelge 863. Öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar (devamı var)

Öznitelik	Veri yapısı	Alan	Sorgu	Ara
küme iş yükü mru			MQIA_CLWL_MRU_KANAL	MQINQ
küme iş yükü kullanım kuyruğu			MQIA_CLWL_USEQ	MQINQ
komut olayı			MQIA_COMMAND_EVENT	MQINQ
komut giriş kuyruğu adı			MQCA_COMMAND_INPUT_Q_NAME	MQINQ
komut düzeyi			MQIA_COMMAND_LEVEL	MQINQ
komut sunucusu denetimi			MQIA_CMD_SERVER_CONTROL	MQINQ
bağlantı seçenekleri	MQCNO	Seçenekler		MQCONN, MQCONNX
Bağlantı Tanıtıcısı	MQCNO	ConnectionId		MQCONNX
Bağlantı durumu				MQCONN, MQCONNX, MQDISC
bağlantı etiketi	MQCD	ConnTag		MQCONNX
şifreleme donanımı	MQSCO	CryptoHardware		MQCONNX
ileti kuyruğu adı			MQCA_DEAD_LETTER_Q_NAME	MQINQ
varsayılan iletim kuyruğu adı			MQCA_DEF_XMIT_Q_NAME	MQINQ
dağıtım listeleri			MQIA_DIST_LIST	MQINQ
dns grubu			MQCA_DNS_GROUP	MQINQ
dns wlm			MQIA_DNS_WLM	MQINQ
ilk kimlik doğrulama kaydı	MQSCO	AuthInfoRecOffset		MQCONNX
	MQSCO	AuthInfoRecPtr		MQCONNX
etkinliği engelle			MQIA_INHIBIT_EVENT	MQINQ
ip adresi sürümü			MQIA_IP_ADDRESS_VERSION	MQINQ
anahtar havuzu	MQSCO	KeyRepository		MQCONNX
anahtar sıfırlama sayısı	MQSCO	KeyResetSayısı		MQCONNX
dinleyici süreölçeri			MQIA_LISTENER_TIMER	MQINQ
yerel olay			MQIA_LOCAL_EVENT	MQINQ
günlüğe kaydedici olayı			MQIA_LOGGER_EVENT	MQINQ
lu grup adı			MQCA_LU_GROUP_NAME	MQINQ
lu adı			MQCA_LU_ADı	MQINQ

Çizelge 863. Öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar (devamı var)

Öznitelik	Veri yapısı	Alan	Sorgu	Ara
lu62 kol soneki			MQCA_LU62_ARM_SUFFIX	MQINQ
lu62 kanalları			MQIA_LU62_CHANNELS	MQINQ
etkin kanal sayısı üst sınırı			MQIA_ACTIVE_KANAL	MQINQ
kanal üst sınırı			MQIA_MAX_KANAL SAYISI	MQINQ
çekme noktası üst sınırı			MQIA_MAX_TANITICI SAYISI	MQINQ
ileti uzunluğu üst sınırı			MQIA_MAX_MSG_LENGTH	MQINQ
öncelik üst sınırı			MQIA_MAX_PRIORITY	MQINQ
kesinleştirilmemiş ileti sayısı üst sınırı			MQIA_MAX_UNCOMMITTED_MSGS	MQINQ
mqi muhasebesi			MQIA_ACCOUNTING_MQI	MQINQ
mqi istatistikleri			MQIA_STATISTICS_MQI	MQINQ
giden kapı üst sınırı			MQIA_OUTBOUND_PORT_MAX	MQINQ
giden kapı alt sınırı			MQIA_OUTBOUND_PORT_MIN	MQINQ
parola	MQCSP	CSPPasswordPtr		MQCONN
	MQCSP	CSPPasswordOffset		MQCONN
	MQCSP	CSPPasswordLength		MQCONN
performans olayı			MQIA_PERFORMANCE_EVENT	MQINQ
platformu			MQIA_PLATFORM	MQINQ
kuyruk muhasebesi			MQIA_ACCOUNTING_Q	MQINQ
kuyruk izleme			MQIA_MONITORING_Q	MQINQ
kuyruk istatistikleri			MQIA_STATISTICS_Q	MQINQ
alma zamanaşımı			MQIA_RECEIVE_TIMEOUT	MQINQ
alma zamanaşımı alt sınırı			MQIA_RECEIVE_TIMEOUT_MIN	MQINQ
alma zamanaşımı tipi			MQIA_RECEIVE_TIMEOUT_TYPE	MQINQ
uzak olay			MQIA_REMOTE_EVENT	MQINQ
Havuz adı			MQCA_HAVUZU_ADI	MQINQ
havuz ad listesi			MQCA_REPOSITORY_NAMELIST	MQINQ
paylaşılan kuyruk yöneticisi adı			MQIA_SHARED_Q_Q_MGR_NAME	MQINQ
ssl olayı			MQIA_SSL_EVENT	MQINQ

Çizelge 863. Öznitelikler, veri yapıları, alanlar, sorgular ve çağrılar (devamı var)				
Öznitelik	Veri yapısı	Alan	Sorgu	Ara
ssl fips			MQIA_SSL_FIPS_REQUIRED	MQINQ
ssl anahtar sıfırlama sayısı			MQIA_SSL_RESET_COUNT	MQINQ
start-stop olayı			MQIA_START_STOP_EVENT	MQINQ
istatistik aralığı			MQIA_STATISTICS_INTERVAL	MQINQ
eşitleme noktası kullanılabilirliği			MQIA_SYNCPOINT	MQINQ
tcp kanalları			MQIA_TCP_CHANNELS	MQINQ
tcp canlı tut			MQIA_TCP_KEEP_ALIVE	MQINQ
tcp adı			MQCA_TCP_NAME	MQINQ
tcp yığın tipi			MQIA_TCP_STACK_TYPE	MQINQ
rota kaydını izle			MQIA_TRACE_ROUTE_RECORDING	MQINQ
tetikleme aralığı			MQIA_TRIGGER_INTERVAL	MQINQ
kullanıcı kimliği	MQCSP	CSPUserIdPtr		MQCONN
	MQCSP	CSPUserIdKayması		MQCONN
	MQCSP	CSPUserIdUzunluğu		MQCONN

ImqReferenceÜstbilgi çapraz başvurusu

ImqAuthenticationRecord C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

Çizelge 864. Özniteliklerin, veri yapılarının ve alanların eşlenmesi		
Öznitelik	Veri yapısı	Alan
hedef ortam	MQRMH	DestEnvUzunluğu, DestEnvGörelili Konumu
Hedef Adı	MQRMH	DestNameUzunluğu, DestNameKayması
Eşgörünüm Tanıtıcısı	MQRMH	ObjectInstanceTanıtıcısı
mantıksal uzunluk	MQRMH	DataLogicalLength (Veri Mantıksal)
mantıksal görelili konum	MQRMH	DataLogicalGörelili Konum
mantıksal görelili konum 2	MQRMH	DataLogicalOffset2
Başvuru tipi	MQRMH	ObjectType
Kaynak Ortamı	MQRMH	SrcEnvUzunluk, SrcEnvGörelili Konum
KAYNAK ADI	MQRMH	SrcNameUzunluğu, SrcNameGörelili Konumu

ImqTrigger çapraz başvurusu

ImqAuthenticationRecord C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

Çizelge 865. Özniteliklerin, veri yapılarının ve alanların eşlenmesi		
Öznitelik	Veri yapısı	Alan
Uygulama tanıtıcısı	MQTM	ApplId
Uygulama Tipi	MQTM	ApplType
ortam verileri	MQTM	EnvData
İşlem adı	MQTM	ProcessName
Kuyruk adı	MQTM	QName
tetikleyici verileri	MQTM	TriggerData
Kullanıcı verileri	MQTM	UserData

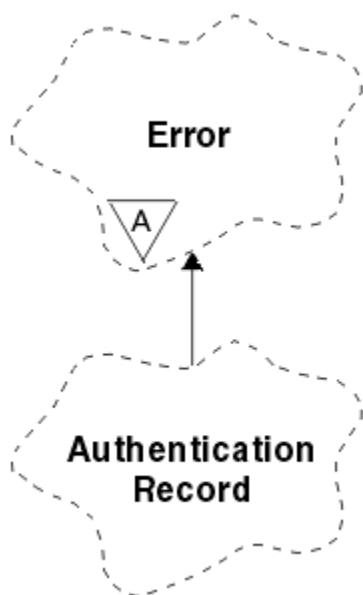
ImqWorkÜstbilgi çapraz başvurusu

ImqAuthenticationRecord C++ sınıfına ilişkin özniteliklerin, veri yapılarının ve alanların çapraz başvurusu.

Çizelge 866. Özniteliklerin, veri yapılarının ve alanların eşlenmesi		
Öznitelik	Veri yapısı	Alan
ileti simgesi	MQWIH	MessageToken
Hizmet Adı	MQWIH	ServiceName
hizmet adımı	MQWIH	ServiceStep

ImqAuthenticationKayıt C++ sınıfı

Bu sınıf, özel TLS istemci bağlantıları için ImqQueueManager: :connect yönteminin yürütülmesi sırasında kullanılacak bir kimlik doğrulama bilgileri kaydını (MQAIR) içerir.



Şekil 14. ImqAuthenticationKayıt sınıfı

Daha fazla ayrıntı için ImqQueueManager: :connect yönteminin tanımına bakın. Bu sınıf z/OS altyapısında yok.

- “Nesne öznitelikleri” sayfa 1773
- “Oluşturucular” sayfa 1773

- [“Nesne yöntemleri \(genel\)” sayfa 1773](#)
- [“Nesne yöntemleri \(korumalı\)” sayfa 1774](#)

Nesne öznitelikleri

Bağlantı adı

LDAP CRL sunucusu bağlantısının adı. Bu, IP adresi ya da DNS adıdır ve isteğe bağlı olarak kapı numarasından sonra parantez içinde gelir.

bağlantı başvurusu

Bir (yerel) kuyruk yöneticisine gerekli bağlantıyı sağlayan bir `ImqQueueManager` nesnesine başvuru. İlk değer sıfırdır. Bunu, adlandırılmış bir kuyruk için bir kuyruk yöneticisini (büyük olasılıkla uzak) tanıtan kuyruk yöneticisi adıyla karıştırmayın.

sonraki kimlik doğrulama kaydı

Bu sınıfın sonraki nesnesi, belirli bir sırada değil, bu nesneyle aynı **bağlantı başvurusuna** sahip. İlk değer sıfırdır.

parola

LDAP CRL sunucusuna bağlantı kimlik doğrulaması için sağlanan bir parola.

önceki kimlik doğrulama kaydı

Bu sınıfın önceki nesnesi, belirli bir sırada değil, bu nesneyle aynı **bağlantı başvurusuna** sahip. İlk değer sıfırdır.

tip

Kayıtta bulunan kimlik doğrulama bilgilerinin tipi.

kullanıcı adı

LDAP CRL sunucusuna yetkilendirme için sağlanan bir kullanıcı kimliği.

Oluşturucular

`ImqAuthenticationRecord ()`;

Varsayılan oluşturucu.

Nesne yöntemleri (genel)

`void operator = (const ImqAuthenticationRecord & air);`

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *hava'* dan kopyalar.

`const ImqString & connectionName () const;`

Bağlantı adını döndürür.

`void setConnectionAd (const ImqString & ad);`

Bağlantı adını belirler.

`void setConnectionAd (const char * ad = 0);`

Bağlantı adını belirler.

`ImqQueueManager * connectionReference () const;`

Bağlantı başvurusunu döndürür.

`void setConnectionReference (ImqQueueManager & manager);`

Bağlantı başvurusunu ayarlar.

`void setConnectionReference (ImqQueueManager * manager = 0);`

Bağlantı başvurusunu ayarlar.

`void copyOut (MQAIR * pAir);`

Eşgörünüm verilerini, var olan eşgörünüm verilerini değiştirerek *pAir* ününe kopyalar. Bu, bağımlı depolamanın ayrılmasını içerebilir.

`void clear (MQAIR * pAir);`

pAir tarafından başvuru yapıyı temizler ve bağımlı depolamayı serbest bırakır.

`ImqAuthenticationRecord * nextAuthenticationRecord () const;`

Sonraki kimlik doğrulama kaydını döndürür.

const ImqString & password () const;
password(parola) deęerini döndürür.

void setPassword (const ImqString & password);
Parola' yı ayarlar.

void setPassword (const char * password = 0);
Parola' yı ayarlar.

ImqAuthenticationRecord * previousAuthenticationRecord () const;
Önceki kimlik doęrulama kaydınıdöndürür.

MQLONG tip () const;
tipdeęerini döndürür.

void setType (const MQLONG tip);
Tip' i ayarlar.

const ImqString & userName () const;
kullanıcı adınıdöndürür.

void setUsername (const ImqString & name);
Kullanıcı adınıayarlar.

void setUsername (const char * name = 0);
Kullanıcı adınıayarlar.

Nesne yöntemleri (korumalı)

void setNextAuthenticationRecord (ImqAuthenticationRecord * pAir = 0);
Sonraki kimlik doęrulama kaydınıayarlar.

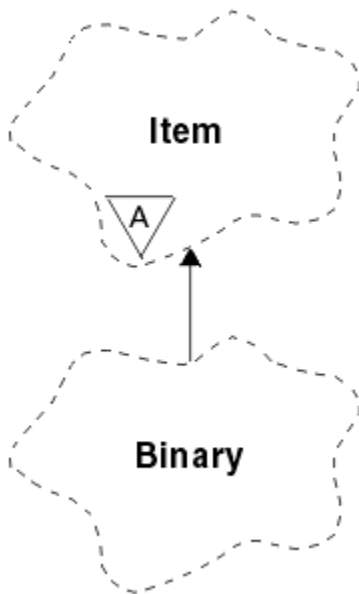
Dikkat: Bu işlevi, kimlik doęrulama kaydı listesinin bozulmayacaęından eminseniz kullanın.

void setPreviousAuthenticationRecord (ImqAuthenticationRecord * pAir = 0);
Önceki kimlik doęrulama kaydınıayarlar.

Dikkat: Bu işlevi, kimlik doęrulama kaydı listesinin bozulmayacaęından eminseniz kullanın.

ImqBinary C++ sınıfı

Bu sınıf, ImqMessage **muhasebe simgesi**, **ilinti tanıtıcısı** ve **ileti tanıtıcısı** deęerleri için kullanılabilir ikili byte dizisini içerir. Kolay atama, kopyalama ve karşılaştırmaya olanak sağlar.



Şekil 15. ImqBinary sınıfı

- “Nesne öznitelikleri” sayfa 1775
- “Oluşturucular” sayfa 1775
- “ImqItem yöntemleri aşırı yüklendi” sayfa 1775
- “Nesne yöntemleri (genel)” sayfa 1775
- “Nesne yöntemleri (korumalı)” sayfa 1776
- “Neden kodları” sayfa 1776

Nesne öznitelikleri

veri

İkili veri byte dizisi. İlk değer boş.

veri uzunluğu

Bayt sayısı. İlk değer sıfırdır.

veri işaretçisi

Verilerin ilk baytının adresi. İlk değer sıfırdır.

Oluşturucular

ImqBinary();

Varsayılan oluşturucu.

ImqBinary(const ImqBinary & binary);

Kopya oluşturucu.

ImqBinary(const void * data, const size_t length);

uzunluk baytı *verilerden* kopyalar.

ImqItem yöntemleri aşırı yüklendi

sanal ImqBoolean copyOut (ImqMessage & msg);

Var olan içeriği değiştirerek **verileri** ileti arabelleğine kopyalar. *msg biçimini* MQFMT_NONE olarak ayarlar.

Daha fazla ayrıntı için ImqItem sınıf yöntemi açıklamasına bakın.

sanal ImqBoolean pasteIn (ImqMessage & msg);

Var olan **verileri** değiştirerek, ileti arabelleğindeki geri kalan verileri aktararak **verileri** belirler.

Başarılı olmak için ImqMessage **biçimi** MQFMT_NONE olmalıdır.

Daha fazla ayrıntı için ImqItem sınıf yöntemi açıklamasına bakın.

Nesne yöntemleri (genel)

void operator = (const ImqBinary & binary);

İkili den bayt kopyalar.

ImqBoolean işleci == (const ImqBinary & binary);

Bu nesneyi *ikili* ile karşılaştırır. Eşit değilse FALSE, tersi durumda TRUE değerini döndürür. Nesneler aynı **veri uzunluğuna** sahipse ve bayt eşleşiyorsa, nesneler eşittir.

ImqBoolean copyOut (void * buffer, const size_t length, const char pad = 0);

veri işaretçisinden en çok *uzunluk* baytı *arabelleğ* e kopyalar. **Veri uzunluğu** yetersizse, *arabellekte* kalan yer *doldurma* byte 'ları ile doldurulur. *buffer*, *length* değeri de sıfır ise sıfır olabilir. *uzunluk* eksi olmamalıdır. Başarılı olursa TRUE değerini döndürür.

size_t dataLength () const ;

Veri uzunluğunu döndürür.

ImqBoolean setDataLength (const size_t length);

Veri uzunluğunu ayarlar. Bu yöntemin sonucu olarak **veri uzunluğu** değiştirilirse, nesnedeki veriler kullanıma hazırlanmaz. Başarılı olursa TRUE değerini döndürür.

void * dataPointer () const ;

Veri işaretçisinidöndürür.

ImqBoolean isNull () const ;

Veri uzunluğu sıfır ya da tüm **veri** baytları sıfırda TRUE değerini döndürür. Ters durumda FALSE değerini döndürür.

ImqBoolean set (const void * buffer, const size_t length);

uzunluk baytı **arabellekten**kopyalar. Başarılı olursa TRUE değerini döndürür.

Nesne yöntemleri (korumalı)

void clear ();

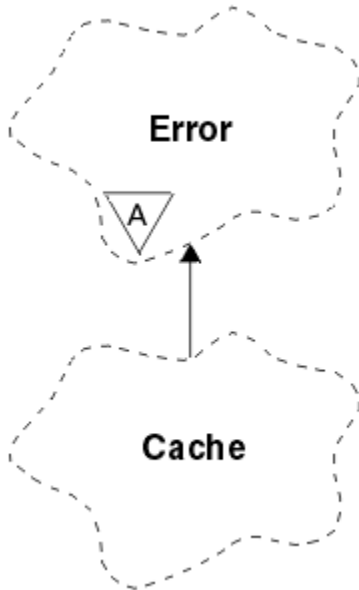
Veri uzunluğunu sıfıra indirir.

Neden kodları

- MQRC_NO_BUFFER
- MQRC_STORAGE_KULLANILAMIYOR
- MQRC_TUTARSIZ_BIÇIMI

ImqCache C++ sınıfı

Verileri bellekte tutmak ya da sıralamak için bu sınıfı kullanın.



Şekil 16. ImqCache sınıfı

Verileri bellekte tutmak ya da sıralamak için bu sınıfı kullanın. Sabit büyüklükteki bir arabelleği aday gösterebilir ya da sistem otomatik olarak esnek bir bellek miktarı sağlayabilir. Bu sınıf, [“ImqCache çapraz başvurusu”](#) sayfa 1758’inde listelenen MQI çağrılarını ile ilgilidir.

- [“Nesne öznitelikleri”](#) sayfa 1777
- [“Oluşturucular”](#) sayfa 1777
- [“Nesne yöntemleri \(genel\)”](#) sayfa 1777
- [“Neden kodları”](#) sayfa 1778

Nesne öznitelikleri

otomatik arabellek

Arabellek belleğinin sistem tarafından otomatik olarak mı (TRUE), yoksa kullanıcı tarafından mı (FALSE) sağlandığını belirtir. Başlangıçta TRUE olarak ayarlanır.

Bu öznitelik doğrudan ayarlanmadı. **useEmptyBuffer** ya da **useFullBuffer** yöntemi kullanılarak dolaylı olarak ayarlanır.

Kullanıcı saklama alanı sağlanırsa, bu öznitelik FALSE olur, arabellek büyüyemez ve arabellek taşması hataları oluşabilir. Arabelleğin adresi ve uzunluğu sabit kalır.

Kullanıcı saklama alanı sağlanmazsa, bu öznitelik TRUE olur ve arabellek, isteğe bağlı ileti verilerini barındırmak için artımlı olarak büyüyebilir. Ancak, arabellek büyüdüğünde arabelleğin adresi değişebilir, bu nedenle **arabellek işaretçisini** ve **veri işaretçisini** kullanırken dikkatli olun.

arabellek uzunluğu

Arabellekteki bayt sayısı. İlk değer sıfırdır.

arabellek göstergesi

Arabellek belleğinin adresi. İlk değer boş.

veri uzunluğu

Veri işaretçisini takip eden bayt sayısı. Bu, **ileti uzunluğuna** eşit ya da ondan küçük olmalıdır. İlk değer sıfırdır.

veri görelî konumu

Veri işaretçisinin önündeki bayt sayısı. Bu, **ileti uzunluğuna** eşit ya da ondan küçük olmalıdır. İlk değer sıfırdır.

veri işaretçisi

Arabelleğin sonraki arabelleğe yazılacak ya da sonraki arabellekten okunacak kısmının adresi. İlk değer boş.

ileti uzunluğu

Arabellekteki önemli verilerin bayt sayısı. İlk değer sıfırdır.

Oluşturucular

ImqCache();

Varsayılan oluşturucu.

ImqCache(const ImqCache & cache);

Kopya oluşturucu.

Nesne yöntemleri (genel)

void operator = (const ImqCache & cache);

cache nesnesinden nesneye en çok **message length** bayt veri kopyalar. **Otomatik arabellek** FALSE ise, **arabellek uzunluğu** kopyalanan verileri barındırmak için yeterli olmalıdır.

ImqBoolean automaticBuffer () const ;

Otomatik arabellek değerini döndürür.

size_t bufferLength () const ;

buffer length(arabellek uzunluğu) değerini döndürür.

char * bufferPointer () const ;

Arabellek işaretçisini döndürür.

void clearMessage ();

İleti uzunluğunu ve **veri görelî konumunu** sıfıra ayarlar.

size_t dataLength () const ;

Veri uzunluğunu döndürür.

size_t dataOffset () const ;

Veri görelî konumunu döndürür.

ImqBoolean setDataOffset (const size_t offset);

Veri görelî konumunu ayarlar. Gerekîyorsa, **ileti uzunluęu** , **veri görelî konumundan** az olmadıęından emin olmak için artırılır. Bu yöntem başarılı olursa TRUE deęerini döndürür.

char * dataPointer () const ;

Veri işaretcîsinin bir kopyasını döndürür.

size_t messageLength () const ;

İleti uzunluęu deęerini döndürür.

ImqBoolean setMessageLength (const size_t length);

İleti uzunluęunu ayarlar. İleti uzunluęunun **arabellek uzunluęundan** büyük olmadıęından emin olmak için gerekirse **arabellek uzunluęunu** artırır. Gerekirse, **veri görelî konumunu ileti uzunluęundan** büyük olmadıęından emin olmak için azaltır. Başarılı olursa TRUE deęerini döndürür.

ImqBoolean moreBytes (const size_t byte-required);

Veri işaretcîsi ile arabelleęin sonu arasında *byte-gerekli* byte 'ın daha fazla kullanılabilir (yazma için) olduęunu garanti eder. Başarılı olursa TRUE deęerini döndürür.

Otomatik arabellek TRUE ise, gerektięinde daha fazla bellek edinilir; tersi durumda, **arabellek uzunluęu** yeterli olmalıdır.

ImqBoolean read (const size_t length, char * & external-buffer);

Arabellekten, **veri işaretcîsi** konumundan başlayarak *uzunluk* byte 'ı *dış arabelleęe* kopyalar. Veriler kopyalandıktan sonra, **veri görelî konumu** *uzunluk* kadar artırılır. Bu yöntem başarılı olursa TRUE deęerini döndürür.

ImqBoolean resizeBuffer (const size_t length);

Otomatik arabellek 'ın TRUE olması koşuluyla, **arabellek uzunluęunu** deęişir. Bu, arabellek belleęinin yeniden yerleřtirilmesiyle gerçekleştirilir. Var olan arabellekteki verilerin en çok **ileti uzunluęu** byte 'ı yeniye kopyalanır. Kopyalanan maksimum sayı *uzunluk* bayttır. **Arabellek göstergesi** deęiřtirildi. **İleti uzunluęu** ve **veri görelî konumu** , yeni arabelleęin sınırları içinde olabildięince yakın bir şekilde korunur. Başarılı olursa TRUE, **otomatik arabellek** FALSE ise FALSE deęerini döndürür.

Not: Sistem kaynaklarıyla ilgili bir sorun varsa, bu yöntem MQRC_STORAGE_NOT_ALLOWED ile başarısız olabilir.

ImqBoolean useEmptyBuffer (const char * external-buffer, const size_t length);

Boş bir kullanıcı arabelleęini tanımlar, **arabellek göstergesi** 'nı *dış arabellek* 'e, **arabellek uzunluęu** 'nı *uzunluk* 'a ve **ileti uzunluęu** 'nı sifıra ayarlar. Bir **clearMessage** gerçekleştirir. Arabellek verilerle tam olarak kullanıma hazır durumdaysa, bunun yerine **useFullBuffer** yöntemini kullanın. Arabellek kısmen verilerle hazırlandıysa, doęru miktarı belirtmek için **setMessageLength** yöntemini kullanın. Bu yöntem başarılı olursa TRUE deęerini döndürür.

Bu yöntem, daha önce açıkladıęı gibi sabit bir bellek miktarını tanımlamak için kullanılabilir (*dış arabellek* boş deęildir ve *uzunluk* sıfır deęildir); bu durumda **otomatik arabellek** FALSE olarak ayarlanır ya da sistem tarafından yönetilen esnek belleęe geri dönmek için kullanılabilir (*dış arabellek* boş ve *uzunluk* sıfırdır); bu durumda **otomatik arabellek** TRUE olarak ayarlanır.

ImqBoolean useFullBuffer (const char * externalBuffer, const size_t length);

useEmptyBuffer için, **ileti uzunluęu** *length* olarak ayarlı olması dışında. Başarılı olursa TRUE deęerini döndürür.

ImqBoolean write (const size_t length, const char * external-buffer);

length byte 'ı *dış arabellekten*, **veri işaretcîsi** konumundan başlayarak arabelleęe kopyalar. Veriler kopyalandıktan sonra, **veri görelî konumu** *uzunluk* kadar artırılır ve gerekirse, yeni **veri görelî konumu** deęerinden küçük olmadıęından emin olmak için **ileti uzunluęu** artırılır. Bu yöntem başarılı olursa TRUE deęerini döndürür.

Otomatik arabellek TRUE ise, yeterli miktarda bellek garanti edilir; tersi durumda, en yüksek **veri görelî konumu arabellek uzunluęunu** aşmamalıdır.

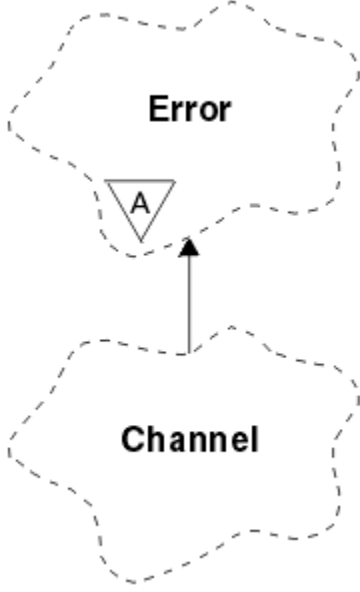
Neden kodları

- MQRC_BUFFER_NOT_AUTOMATIC

- MQRD_DATA_TRUNCATED
- MQRD_INSUYETERINCE arabellek
- MQRD_VERIMSIZ_VERI
- MQRD_NULL_POINTER
- MQRD_STORAGE_KULLANILAMIYOR
- MQRD_ZERO_LENGTH

ImqChannel C++ sınıfı

Bu sınıf, özel istemci bağlantıları için Manager: :connect yönteminin yürütülmesi sırasında kullanılacak bir kanal tanımlamasını (MQRD) içerir.



Şekil 17. ImqChannel sınıfı

Daha fazla bilgi için Manager: :connect yönteminin ve [Sample program HELLO WORLD \(imqworld.cpp\)](#) programının açıklamasına bakın.

Listelenen yöntemlerin tümü tüm platformlar için geçerli değildir. Ek bilgi için [DEFINE CHANNEL](#) ve [ALTER CHANNEL](#) komutlarına bakın.

ImqChannel sınıfı z/OS üzerinde desteklenmez.

- [“Nesne öznitelikleri” sayfa 1779](#)
- [“Oluşturucular” sayfa 1781](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1781](#)
- [“Neden kodları” sayfa 1784](#)

Nesne öznitelikleri

toplu kalp atışım

Uzak bir kanalın etkin olup olmadığını denetleyen saniye sayısı. İlk değer 0 'dir.

Kanal Adı

Kanalın adı. İlk değer boş.

Bağlantı adı

Bağlantının adı. Örneğin, bir anasistem bilgisayarının IP adresi. İlk değer boş.

üstbilgi sıkıştırması

Kanal tarafından desteklenen üstbilgi veri sıkıştırma tekniklerinin listesi. İlk değerlerin tümü MQCOMPRESS_NOT_ALLOWED olarak ayarlandı.

kalp atığı aralığı

Bir bağlantının çalışmaya devam edişini denetleyen saniye sayısı. Başlangıç değeri 300 'dür.

canlı tutma aralığı

Kanal için canlı tutma zamanlamasını belirten iletişim yığına geçirilen saniye sayısı. İlk değer MQKAI_AUTO.

Yerel adres

Kanalın yerel iletişim adresi.

ileti uzunluğu üst sınırı

Tek bir iletişimde kanal tarafından desteklenen ileti uzunluğu üst sınırı. İlk değer 4 194 304 'tür.

ileti sıkıştırma

Kanal tarafından desteklenen ileti veri sıkıştırma tekniklerinin listesi. İlk değerlerin tümü MQCOMPRESS_NOT_ALLOWED olarak ayarlandı.

kip adı

Kipin adı. İlk değer boş.

parola

Bağlantı kimlik doğrulaması için sağlanan bir parola. İlk değer boş.

alma çıkış sayısı

Alma çıkışlarının sayısı. İlk değer sıfırdır. Bu öznitelik salt okunur.

çıkış adlarını al

Alma çıkışlarının adları.

kullanıcı verilerini al

Alma çıkışlarıyla ilişkili veriler.

güvenlik çıkışı adı

Bağlantının sunucu tarafında çağrılacak güvenlik çıkışının adı. İlk değer boş.

güvenlik kullanıcısı verileri

Güvenlik çıkışına geçirilecek veriler. İlk değer boş.

çıkış gönderme sayısı

Gönderme çıkışlarının sayısı. İlk değer sıfırdır. Bu öznitelik salt okunur.

çıkış adlarını gönder

Gönderme çıkışlarının adları.

kullanıcı verileri gönder

Gönderme çıkışlarıyla ilişkili veriler.

SSL CipherSpec

TLS ile kullanılmak üzere CipherSpec .

SSL istemcisi kimlik denetimi tipi

TLS ile kullanılacak istemci kimlik doğrulaması tipi.

SSL eş adı

TLS ile kullanılacak eş adı.

hareket işleme programı adı

Hareket programının adı. İlk değer boş.

iletim tipi

Bağlantının iletim tipi. İlk değer: MQXPT_LU62.

kullanıcı kimliği

Yetki için sağlanan bir kullanıcı kimliği. İlk değer boş.

Olusturucular

ImqChannel() ;

Varsayılan olusturucu.

ImqChannel(const ImqChannel & kanal);

Kopya olusturucu.

Nesne yontemleri (genel)

void operator = (const ImqChannel & kanal);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *kanal'* dan kopyalar.

MQLONG batchHeartBeat () const;

Toplu iş kalp atışını döndürür.

ImqBoolean setBatchHeartBeat(const MQLONG sağlıklı işletim bildirim = 0L);

Toplu iş kalp atışını ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString channelName() const;

Kanal adını döndürür.

ImqBoolean setChannelName (const char * name = 0);

Kanal adını belirler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString connectionName() const;

Bağlantı adını döndürür.

ImqBoolean setConnectionName (const char * name = 0);

Bağlantı adını belirler. Bu yöntem başarılı olursa TRUE değerini döndürür.

size_t headerCompressionCount () const;

Desteklenen üstbilgi veri sıkıştırma teknikleri sayısını döndürür.

ImqBoolean headerCompression(const size_t sayısı, MQLONG compress []) const;

compress içinde desteklenen üstbilgi veri sıkıştırma tekniklerinin kopyalarını döndürür. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setHeaderSıkıştırma (const size_t sayısı, const MQLONG compress []);

Desteklenen üstbilgi veri sıkıştırma tekniklerini **sıkıştırı** olarak ayarlar.

Desteklenen üstbilgi veri sıkıştırma teknikleri sayısını **sayı** olarak ayarlar.

Bu yöntem başarılı olursa TRUE değerini döndürür.

MQLONG heartBeatInterval () const;

heart-beat interval(kalp atışı aralığı) değerini döndürür.

ImqBoolean setHeartBeatInterval(const MQLONG interval = 300L);

heart-beat interval(Kalp atırım aralığı) değerini ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

MQLONG keepAliveAralığı () const;

Canlı tutma aralığı döndürür.

ImqBoolean setKeepAliveInterval(const MQLONG interval = MQKAI_AUTO);

Canlı tutma aralığı belirler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString localAddress() const;

Yerel adres döndürür.

ImqBoolean setLocalAddress (const char * address = 0);

Yerel adres belirler. Bu yöntem başarılı olursa TRUE değerini döndürür.

MQLONG maximumMessageLength () const;

İleti uzunluğu üst sınırı döndürür.

ImqBoolean setMaximumMessageLength(const MQLONG length = 4194304L);

İleti uzunluğu üst sınırı belirler. Bu yöntem başarılı olursa TRUE değerini döndürür.

size_t messageCompressionCount () const;

Desteklenen ileti veri sıkıştırma teknikleri sayısını döndürür.

ImqBoolean messageCompression(const size_t count, MQLONG compress []) const;

compress içinde desteklenen ileti veri sıkıştırma tekniklerinin kopyalarını döndürür. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setMessageCompression (const size_t count, const MQLONG compress []);

Sıkıştırılması için desteklenen ileti veri sıkıştırma tekniklerini ayarlar.

Desteklenen ileti veri sıkıştırma teknikleri sayısını sayıya ayarlar.

Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString modeName() const;

kip adını döndürür.

ImqBoolean setModeName (const char * name = 0);

kip adını ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString password () const;

password(parola) değerini döndürür.

ImqBoolean setPassword(const char * password = 0);

Parola' yı ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

size_t receiveExitCount () const;

Alma çıkış sayısını döndürür.

ImqString receiveExitAd ();

Varsa, **alma çıkış adlarının** birincini döndürür. **Alma çıkış sayısı** sıfırsa, boş bir dizgi döndürür.

ImqBoolean receiveExitNames (const size_t count, ImqString * names []);

adlarda alma çıkış adlarının kopyalarını döndürür. **Alma çıkış sayısını** aşan **adları** boş değerli dizgilere ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setReceiveExitName(const char * ad = 0);

Alma çıkış adlarını tek **ad** olarak ayarlar. **ad** boş değerli ya da boş olabilir. **Alma çıkış sayısını** 1 ya da sıfır olarak ayarlar. **Kullanıcı verilerini** al seçeneğini temizler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setReceiveExitNames(const size_t count, const char * names []);

Alma çıkış adlarını **adlar** olarak ayarlar. Tek tek **adlar** değerleri boş değerli ya da boş olmamalıdır.

Alma çıkış sayısını **sayı** olarak ayarlar. **Kullanıcı verilerini** al seçeneğini temizler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setReceiveExitNames(const size_t count, const ImqString * names []);

Alma çıkış adlarını **adlar** olarak ayarlar. Tek tek **adlar** değerleri boş değerli ya da boş olmamalıdır.

Alma çıkış sayısını **sayı** olarak ayarlar. **Kullanıcı verilerini** al seçeneğini temizler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString receiveUserData ();

Varsa, **kullanıcı verilerini al** öğelerinin ilkini döndürür. **Alma çıkış sayısı** sıfırsa, boş bir dizgi döndürür.

ImqBoolean receiveUserData (const size_t count, ImqString * data []);

receive user data (kullanıcı verilerini al) öğelerinin **data**(veri) içindeki kopyalarını döndürür. **Alma çıkış sayısını** aşan **verileri** boş değerli dizgilere ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setReceiveUserData(const char * data = 0);

Kullanıcı verilerini al seçeneğini tek öğe **verilerine** ayarlar. **Veri** boş değilse, **alma çıkış sayısı** en az 1 olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setReceiveUserData(const size_t count, const char * data []);

Kullanıcı verilerini al seçeneğini **veri** olarak ayarlar. **sayı** , **alma çıkış sayısından** büyük olmamalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setReceiveUserData(const size_t count, const ImqString * data []);
Kullanıcı verilerini al seçeneğini *veri* olarak ayarlar. *sayı* , **alma çıkış sayısından** büyük olmamalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString securityExitName () const;
Güvenlik çıkışı adını döndürür.

ImqBoolean setSecurityExitName(const char * ad = 0);
Güvenlik çıkışı adını ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString securityUserData () const;
Güvenlik kullanıcı verilerini döndürür.

ImqBoolean setSecurityUserData(const char * data = 0);
Güvenlik kullanıcı verilerini ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

size_t sendExitCount () const;
Gönderme çıkış sayısını döndürür.

ImqString sendExitAd ();
Varsa, çıkış adlarını gönder öğesinin ilk adını döndürür. **Gönderme çıkış sayısı** sıfırda boş bir dizgi döndürür.

ImqBoolean sendExitNames (const size_t count, ImqString * names []);
Adlarda çıkış adlarını gönder öğesinin kopyalarını döndürür. **Gönderme çıkış sayısını** aşan *adları* boş değerli dizgilere ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setSendExitName(const char * ad = 0);
Gönderme çıkış adlarını tek *ad* olarak ayarlar. *ad* boş değerli ya da boş olabilir. **send exit count** değerini 1 ya da sıfır olarak ayarlar. **Kullanıcı verilerini gönder** seçeneğini temizler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setSendExitNames(const size_t count, const char * names []);
Çıkış adlarını gönder seçeneğini *adlar* olarak ayarlar. Tek tek *adlar* değerleri boş değerli ya da boş olmamalıdır. **send exit count** (Çıkış sayısını gönder) seçeneğini *count*(sayı) olarak ayarlar. **Kullanıcı verilerini gönder** seçeneğini temizler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setSendExitNames(const size_t count, const ImqString * names []);
Çıkış adlarını gönder seçeneğini *adlar* olarak ayarlar. Tek tek *adlar* değerleri boş değerli ya da boş olmamalıdır. **send exit count** (Çıkış sayısını gönder) seçeneğini *count*(sayı) olarak ayarlar. **Kullanıcı verilerini gönder** seçeneğini temizler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString sendUserData ();
Varsa, kullanıcı verileri gönder öğelerinin ilkini döndürür. **Gönderme çıkış sayısı** sıfırda boş bir dizgi döndürür.

ImqBoolean sendUserData (const size_t count, ImqString * data []);
data(veri) içindeki **send user data** (kullanıcı verileri gönder) öğelerinin kopyalarını döndürür. **send exit count** değerini aşan *verileri* boş değerli dizgilere ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setSendUserData(const char * data = 0);
Kullanıcı verilerini gönder seçeneğini tek öğe *verilerine* ayarlar. *Veri* boş değilse, **gönderme çıkış sayısı** en az 1 olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setSendUserData(const size_t count, const char * data []);
Kullanıcı verilerini gönder seçeneğini *veri* olarak ayarlar. *count* (sayı), **send exit count** değerinden büyük olmamalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setSendUserData(const size_t count, const ImqString * data []);
Kullanıcı verilerini gönder seçeneğini *veri* olarak ayarlar. *count* (sayı), **send exit count** değerinden büyük olmamalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString sslCipherSpecification () const;
TLS şifre belirtimini döndürür.

ImqBoolean setSslCipherSpecification(const char * name = 0);
TLS şifre belirtimini ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

MQLONG sslClientKimlik Doğrulaması () const;

TLS istemci kimlik doğrulama tipini döndürür.

ImqBoolean setSslClientAuthentication(const MQLONG auth = MQSCA_REQUIRED);

TLS istemci kimlik doğrulaması tipini ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString sslPeerName () const;

TLS eş adını döndürür.

ImqBoolean setSslPeerName(const char * ad = 0);

TLS eş adını ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString transactionProgramAdı () const;

Hareket programı adını döndürür.

ImqBoolean setTransactionProgramName(const char * ad = 0);

Hareket programı adını belirler. Bu yöntem başarılı olursa TRUE değerini döndürür.

MQLONG transportType() const;

iletim tipini döndürür.

ImqBoolean setTransportType (const MQLONG tip = MQXPT_LU62);

İletim tipini ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString userId() const;

Kullanıcı kimliğini döndürür.

ImqBoolean setUserId (const char * id = 0);

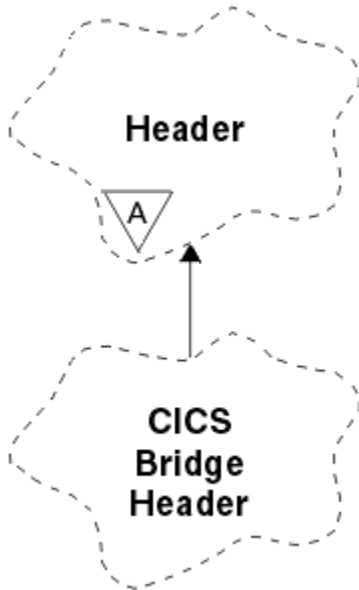
Kullanıcı kimliğini ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

Neden kodları

- MQRC_DATA_LENGTH_HATA
- MQRC_ITEM_COUNT_HATA
- MQRC_NULL_POINTER
- MQRC_SOURCE_BUFFER_ERROR (QRC_SOURCE_BUFFER_ERROR)

ImqCICSBridgeHeader C++ sınıfı

Bu sınıf, MQCIH veri yapısının belirli özelliklerini içerir.



Şekil 18. ImqCICSBridgeHeader sınıfı

Bu sınıftaki nesnelere, IBM MQ for z/OS aracılığıyla CICS bridge 'e' ileti gönderen uygulamalar tarafından kullanılır.

- [“Nesne öznitelikleri” sayfa 1785](#)
- [“Oluşturucular” sayfa 1787](#)
- [“İmqItem yöntemleri aşırı yüklendi” sayfa 1787](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1787](#)
- [“Nesne verileri \(korunan\)” sayfa 1790](#)
- [“Neden kodları” sayfa 1790](#)
- [“Dönüş kodları” sayfa 1790](#)

Nesne öznitelikleri

ADS tanımlayıcısı

ADS tanımlayıcısını gönder/al. Bu, MQCADSD_NONE kullanılarak ayarlanır. İlk değer MQCADSD_NONE. Aşağıdaki ek değerler kullanılabilir:

- MQCADSD_NONE
- MQCADSD_SEND
- MQCADSD_RECV
- MQCADSD_MSGFORMAT

uyarı tanıtıcısı

AID anahtarı. Alan MQ_ATTENTION_ID_LENGTH uzunluğunda olmalıdır.

kimlik doğrulayıcı

RACF parolası ya da geçiş bileti. İlk değer, MQ_AUTHENTICATOR_LENGTH uzunluğundaki boşluklar içeriyor.

köprü olağandışı bitiş kodu

MQ_ABEND_CODE_LENGTH uzunluğunda köprü olağandışı bitiş kodu. İlk değer dört boş karakterdir. Bu alanda döndürülen değer dönüş koduna bağlıdır. Daha fazla ayrıntı için bkz. [Çizelge 867 sayfa 1790](#).

köprü iptal kodu

Köprü işlem kodunu olağandışı sonlandırdı. Alan ayrılmış, boşluk içermeli ve MQ_CANCEL_CODE_LENGTH uzunluğunda olmalıdır.

köprü tamamlama kodu

IBM MQ tamamlama kodunu ya da CICS EIBRESP değerini içerebilen tamamlanma kodu. Alan, MQCC_OK başlangıç değerini içeriyor. Bu alanda döndürülen değer dönüş koduna bağlıdır. Daha fazla ayrıntı için bkz. [Çizelge 867 sayfa 1790](#).

köprü hatası görelî konumu

Köprü hatası görelî konumu. İlk değer sıfırdır. Bu öznitelik salt okunur.

köprü neden kodu

Neden kodu. Bu alan IBM MQ nedenini ya da CICS EIBRESP2 değerini içerebilir. Alan, MQRC_NONE başlangıç değerini içeriyor. Bu alanda döndürülen değer dönüş koduna bağlıdır. Daha fazla ayrıntı için bkz. [Çizelge 867 sayfa 1790](#).

köprü dönüş kodu

CICS bridge' den gelen dönüş kodu. İlk değer MQCRC_OK.

etkileşimli görev

Görevin sohbet amaçlı olup olamayacağını belirler. İlk değer MQCCT_NO. Aşağıdaki ek değerler kullanılabilir:

- MQCCT_YES
- MQCCT_NO

imleç konumu

İmleç konumu. İlk değer sıfırdır.

tesis tutma süresi

CICS bridge tesis serbest bırakma süresi.

benzer tesis

Uçbirim öykünülen özneteliği. Alan MQ_FACILITY_LIKE_LENGTH uzunluğunda olmalıdır.

olanak simgesi

BVT simgesi değeri. Alan MQ_FACILITY_LENGTH uzunluğunda olmalıdır. İlk değer MQCFAC_NONE.

işlev

IBM MQ çağrı adını ya da CICS EIBFN işlevini içerebilen işlev. Alanın ilk değeri MQCFUNC_NONE, uzunluğu MQ_FUNCTION_LENGTH. Bu alanda döndürülen değer dönüş koduna bağlıdır. Daha fazla ayrıntı için bkz. [Çizelge 867 sayfa 1790](#) .

İşlev bir IBM MQ çağrı adı içerdiğinde aşağıdaki ek değerler kullanılabilir:

- MQCFUNC_MQCONN
- MQCFUNC_MQGET
- MQCFUNC_MQINQ
- MQCFUNC_NONE
- MQCFUNC_MQOPEN
- MQCFUNC_PUT
- MQCFUNC_MQPUT1

bekleme aralığını al

CICS bridge görevi tarafından verilen bir MQGET çağrısı için bekleme aralığı. İlk değer MQCGWI_DEFAULT. Alan yalnızca **uow denetimi** MQCUOWC_FIRST değerine sahip olduğunda geçerlidir. Aşağıdaki ek değerler kullanılabilir:

- MQCGWI_DEFAULT
- MQWI_SINIRSIZ

Bağlantı tipi

Bağlantı tipi. İlk değer MQCLT_PROGRAM. Aşağıdaki ek değerler kullanılabilir:

- MQCLT_PROGRAM
- MQCLT_TRANSACTION

sonraki işlem tanıtıcısı

Eklenecek sonraki işlemin tanıtıcısı. Alan, MQ_TRANSACTION_ID_LENGTH uzunluğunda olmalıdır.

çıkış verileri uzunluğu

COMMAREA veri uzunluğu. İlk değer MQCODL_AS_INPUT.

yanıtlama biçimi

Yanıt iletilisinin biçim adı. İlk değer MQFMT_NONE ve uzunluğu MQ_FORMAT_LENGTH.

başlatma kodu

İşlem başlatma kodu. Alan MQ_START_CODE_LENGTH uzunluğunda olmalıdır. İlk değer MQCSC_NONE. Aşağıdaki ek değerler kullanılabilir:

- MQCSC_START
- MQCSC_STARTDATA
- MQCSC_TERMINPUT
- MQCSC_NONE

görev bitiş durumu

Görev bitiş durumu. İlk değer MQCTES_NOSYNC. Aşağıdaki ek değerler kullanılabilir:

- MQCTES_COMMIT

- MQCTES_BACKOUT
- MQCTES_ENDTASK
- MQCTES_NOSYNC

İşlem tanıtıcısı

Eklenecek işlemin tanıtıcısı. İlk değer boşluk içermelidir ve MQ_TRANSACTION_ID_LENGTH uzunluğunda olmalıdır. Alan yalnızca **uow denetimi** MQCUOWC_FIRST ya da MQCUOWC_ONLY değerine sahip olduğunda geçerlidir.

UOW denetimi

UOW kontrol. İlk değer MQCUOWC_ONLY 'dir. Aşağıdaki ek değerler kullanılabilir:

- MQCUOWC_FIRST
- MQCUOWC_MIDDLE
- MQCUOWC_SON
- YALNIZCA MQCUOWC_ONLY
- MQCUOWC_COMMIT
- MQCUOWC_BACKOUT
- MQCUOWC_CONTINUE

sürüm

MQCIH sürüm numarası. İlk değer: MQCIH_VERSION_2. Desteklenen diğer tek değer MQCIH_VERSION_1' dir.

Oluşturucular

ImqCICSBridgeHeader();

Varsayılan oluşturucu.

ImqCICSBridgeHeader(const ImqCICSBridgeHeader & header);

Kopya oluşturucu.

ImqItem yöntemleri aşırı yüklendi

sanal ImqBoolean copyOut(ImqMessage & msg);

Başlangıçta ileti arabelleğine bir MQCIH veri yapısı ekler, var olan ileti verilerini daha ileriye taşıyarak ileti biçimini MQFMT_CICS olarak ayarlar.

Daha fazla ayrıntı için üst sınıf yöntemi tanımına bakın.

sanal ImqBoolean pasteIn(ImqMessage & msg);

İleti arabelleğindeki bir MQCIH veri yapısını okur. Başarılı olmak için msg nesnesinin kodlaması MQENC_NATIVE olmalıdır. MQGMO_CONVERT ile MQENC_NATIVE iletilerini alın. Başarılı olmak için ImqMessage biçimi MQFMT_CICS olmalıdır.

Daha fazla ayrıntı için üst sınıf yöntemi tanımına bakın.

Nesne yöntemleri (genel)

void işleci = (const ImqCICSBridgeHeader & header);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *üstbilgisinden* kopyalar.

MQLONG ADSDescriptor () const;

ADS tanımlayıcısının bir kopyasını döndürür.

void setADSDescriptor(const MQLONG tanımlayıcı = MQCADSD_NONE);

ADS tanımlayıcısını belirler.

ImqString attentionIdentifier() const;

MQ_ATTENTION_ID_LENGTH uzunluğuna kadar sondaki boşluklarla doldurarak **uyarı tanıtıcısının** bir kopyasını döndürür.

void setAttentionIdentifier (const char * data = 0);

Sondaki boşluklarla doldurularak **uyarı tanıtıcısını** MQ_ATTENTION_ID_LENGTH uzunluğuna ayarlar. *Veri* sağlanmazsa, **uyarı tanıtıcısını** ilk değere sıfırlar.

ImqString kimlik doğrulayıcı () const;

MQ_AUTHENTICATOR_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **authenticator**' un bir kopyasını döndürür.

void setAuthenticator(const char * data = 0);

Sondaki boşluklarla doldurularak **kimlik doğrulayıcıyı** MQ_AUTHENTICATOR_LENGTH uzunluğuna ayarlar. *Veri* sağlanmazsa, **kimlik doğrulayıcı** ' yı ilk değere sıfırlar.

ImqString bridgeAbendCode () const;

MQ_ABEND_CODE_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **köprü olağandışı bitiş kodunun** bir kopyasını döndürür.

ImqString bridgeCancelCode () const;

MQ_CANCEL_CODE_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **köprü iptal kodunun** bir kopyasını döndürür.

void setBridgeCancelCode(const char * data = 0);

Sondaki boşluklarla doldurularak **köprü iptal kodunu** MQ_CANCEL_CODE_LENGTH uzunluğuna ayarlar. *Veri* sağlanmazsa, **köprü iptal kodunu** ilk değere sıfırlar.

MQLONG bridgeCompletionCode () const;

Köprü tamamlama kodunun bir kopyasını döndürür.

MQLONG bridgeErrorGörelî Konumu () const;

Köprü hata görelî konumunun bir kopyasını döndürür.

MQLONG bridgeReasonCode () const;

Köprü neden kodunun bir kopyasını döndürür.

MQLONG bridgeReturnKodu () const;

Köprü dönüş kodunu döndürür.

MQLONG conversationalTask() const;

Dönüştürücü görevin bir kopyasını döndürür.

void setConversationalGörev (const MQLONG görev = MQCCT_NO);

Etkileşimli görevi ayarlar.

MQLONG cursorPosition() const;

İmleç konumunun bir kopyasını döndürür.

void setCursorPosition (const MQLONG position = 0);

İmleç konumunu belirler.

MQLONG facilityKeepTime () const;

Tesis tutma süresi' nin bir kopyasını döndürür.

void setFacilityKeepTime(const MQLONG time = 0);

Tesis tutma süresini ayarlar.

ImqString facilityLike() const;

MQ_FACILITY_LIKE_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **olanağının** gibi bir kopyasını döndürür.

void setFacilityLike (const char * name = 0);

olanağının gibi, sondaki boşluklarla doldurularak MQ_FACILITY_LIKE_LENGTH uzunluğuna ayarlar. *ad* sağlanmazsa, **olanağının ilk değer** gibi ilk durumuna getirir.

ImqBinary facilityToken() const;

tesis simgesinin bir kopyasını döndürür.

ImqBoolean setFacilitySimgesi (const ImqBinary & simgesi);

tesis simgesini ayarlar. *simge* için **veri uzunluğu** sıfır ya da MQ_FACILITY_LENGTH olmalıdır. Başarılı olursa TRUE değerini döndürür.

void setFacilityBelirteci (const MQBYTE8 *simge* = 0);

tesis simgesini ayarlar. *simge* sıfır olabilir; bu, MQCFAC_NONE ile aynıdır. *simge* sıfır dışında bir değerse, ikili verilerin MQ_FACILITY_LENGTH byte 'larına yanıt vermelidir. MQCFAC_NONE gibi önceden tanımlanmış değerleri kullanırken, imza eşleşmesini sağlamak için bir tip dönüşümü yapmanız gerekebilir. Örneğin, (MQBYTE *) MQCFAC_NONE.

ImqString işlevi () const;

MQ_FUNCTION_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **işlevin** bir kopyasını döndürür.

MQLONG getWaitInterval () const;

get wait interval (Bekleme aralığını al) değerinin bir kopyasını döndürür.

void setGetWaitInterval(const MQLONG *interval* = MQCGWI_DEFA

get wait interval (Bekleme süresini al) değerini ayarlar.

MQLONG linkType() const;

Bağlantı tipi 'nin bir kopyasını döndürür.

void setLinkTip (const MQLONG *tip* = MQCLT_PROGRAM);

Bağlantı tipini ayarlar.

ImqString nextTransactionIdentifier () const;

MQ_TRANSACTION_ID_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **sonraki hareket tanıtıcısı** verilerinin bir kopyasını döndürür.

MQLONG outputDataLength () const;

Çıkış verileri uzunluğunun bir kopyasını döndürür.

void setOutputDataLength(const MQLONG *uzunluk* = MQCODL_AS_INPUT);

Çıkış verileri uzunluğunu ayarlar.

ImqString replyToFormat () const;

MQ_FORMAT_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **yanıt-biçimi** adının bir kopyasını döndürür.

void setReplyToFormat(const char * *ad* = 0);

Sondaki boşluklarla doldurularak **yanıt-biçimi** 'ni MQ_FORMAT_LENGTH uzunluğuna ayarlar. *ad* sağlanmazsa, **yanıt-biçimi** ilk değerine geri ayarlanır.

ImqString startCode() const;

MQ_START_CODE_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **başlangıç kodunun** bir kopyasını döndürür.

void setStartCode (const char * *data* = 0);

Sondaki boşluklarla doldurularak **başlangıç kodu** verilerini MQ_START_CODE_LENGTH uzunluğuna ayarlar. *Veri* sağlanmazsa, **başlangıç kodunu** ilk değere sıfırlar.

MQLONG taskEndDurumu () const;

Görev bitiş durumunun bir kopyasını döndürür.

ImqString transactionIdentifier() const;

MQ_TRANSACTION_ID_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak **hareket tanıtıcısı** verilerinin bir kopyasını döndürür.

void setTransactionIdentifier (const char * *data* = 0);

Sondaki boşluklarla doldurularak **hareket tanıtıcısını** MQ_TRANSACTION_ID_LENGTH uzunluğuna ayarlar. Bir *veri* sağlanmazsa, **hareket tanıtıcısı** ilk değere sıfırlanır.

MQLONG UOWControl () const;

UOW denetiminin bir kopyasını döndürür.

void setUOWControl(const MQLONG *control* = MQCUOWC_ONLY);

UOW denetimini ayarlar.

MQLONG sürümü () const;

version numarasını döndürür.

ImqBoolean setVersion(const MQLONG *version* = MQCIH_VERSION_2);

version numarasını ayarlar. Başarılı olursa TRUE değerini döndürür.

Nesne verileri (korunan)

MQLONG olVersion

opcihiçin ayrılan depolamaya yerleştirilebilecek MQCIH sürüm numarası üst sınırı.

PMQCIH opcih

MQCIH veri yapısının adresi. Ayrılan saklama alanı miktarı olVersionile gösterilir.

Neden kodları

- MQRC_BINARY_DATA_LENGTH_HATA
- MQRC_YANLIŞ_SÜRÜMÜ

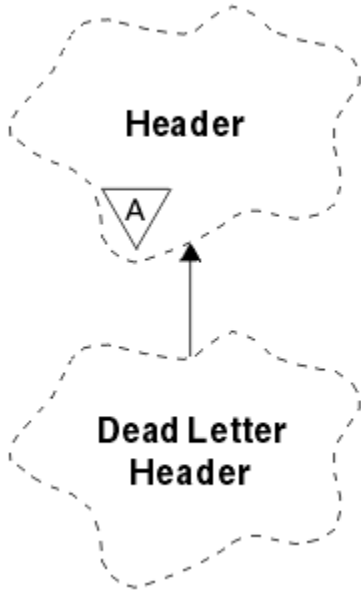
Dönüş kodları

Çizelge 867. ImqCICSBridgeHeader sınıfı dönüş kodları

Dönüş Kodu	İşlev	CompCode	Neden	Olağandışı Bitiş Kodu
MQCRC_OK				
MQCRC_BRIDGE_HATA			MQFB_CICS	
MQCRC_MQ_API_ERROR	IBM MQ arama adı	IBM MQ CompCode	IBM MQ Neden	
MQCRC_BRIDGE_TIMEOUT	IBM MQ arama adı	IBM MQ CompCode	IBM MQ Neden	
MQCRC_CICS_EXEC_ERROR (MQCRC_CICS_HATASI)	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	
MQCRC_SECURITY_HATA	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	
MQCRC_PROGRAM_YOK	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	
MQCRC_TRANSID_NOT_KULLANILAMIYOR	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	
MQCRC_BRIDGE_ABEND				CICS KODU
MQCRC_APPLICATION_ABEND				CICS KODU

ImqDeadLetterHeader C++ sınıfı

Bu sınıf, MQDLH veri yapısının özelliklerini içerir.



Şekil 19. *ImqDeadLetterHeader* sınıfı

Bu sınıftaki nesnelere genellikle işlenemeyen bir iletiyle karşılaşan bir uygulama tarafından kullanılır. Teslim edilmeyen harf üstbilgisinden oluşan yeni bir ileti ve ileti içeriği, teslim edilmeyen iletiler kuyruğuna yerleştirilir ve ileti atılır.

- “Nesne öznitelikleri” sayfa 1791
- “Oluşturucular” sayfa 1792
- “ImqItem yöntemleri aşırı yüklendi” sayfa 1792
- “Nesne yöntemleri (genel)” sayfa 1792
- “Nesne verileri (korunan)” sayfa 1793
- “Neden kodları” sayfa 1793

Nesne öznitelikleri

çıkma mektup neden kodu

İletinin gitmeyen iletiler kuyruğuna gelmesinin nedeni. İlk değer MQRC_NONE.

Hedef kuyruk yöneticisi adı

Özgün hedef kuyruk yöneticisinin adı. Ad, MQ_Q_MGR_NAME_LENGTH uzunluğundaki bir dizedir. İlk değeri boş.

hedef kuyruk adı

Özgün hedef kuyruğun adı. Ad, MQ_Q_NAME_LENGTH uzunluğundaki bir dizedir. İlk değeri boş.

Koyma Uygulaması Adı

İletiyi gönderilmeyen iletiler kuyruğuna koyan uygulamanın adı. Ad, MQ_PUT_APPL_NAME_LENGTH uzunluklu bir dizedir. İlk değeri boş.

Koyma Uygulaması Tipi

İletiyi gönderilmeyen iletiler kuyruğuna koyan uygulama tipi. İlk değer sıfırdır.

koyma tarihi

İletinin gönderilmeyen iletiler kuyruğuna yerleştirildiği tarih. Tarih, MQ_PUT_DATE_LENGTH uzunluğundaki bir dizedir. İlk değeri boş değerli bir dizedir.

koyma süresi

İletinin gönderilmeyen iletiler kuyruğuna konma zamanı. Zaman, MQ_PUT_TIME_LENGTH uzunluğundaki bir dizedir. İlk değeri boş değerli bir dizedir.

Olusturucular

ImqDeadLetterHeader();

Varsayılan olusturucu.

ImqDeadLetterHeader(const ImqDeadLetterHeader & header);

Kopya olusturucu.

ImqItem yöntemleri aşırı yüklendi

sanal ImqBoolean copyOut (ImqMessage & msg);

Var olan ileti verilerini daha ileriye taşıyarak, başlangıçta ileti arabelleğine bir MQDLH veri yapısı ekler. *msg* biçimini MQFMT_DEAD_LETTER_HEADER olarak ayarlar.

Daha fazla ayrıntı için "[ImqHeader C++ sınıfı](#)" sayfa 1799 . sayfadaki ImqHeader sınıf yöntemi açıklamasına bakın.

sanal ImqBoolean pasteIn (ImqMessage & msg);

İleti arabelleğindeki bir MQDLH veri yapısını okur.

Başarılı olmak için ImqMessage biçimi MQFMT_DEAD_LETTER_HEADER olmalıdır.

Daha fazla ayrıntı için "[ImqHeader C++ sınıfı](#)" sayfa 1799 . sayfadaki ImqHeader sınıf yöntemi açıklamasına bakın.

Nesne yöntemleri (genel)

void işleci = (const ImqDeadLetterHeader & header);

Eşgörünüm verilerini kopyalar, var olan eşgörünüm verileri yerine *üstbilgiseçeneğinden* kopyalanır.

MQLONG deadLetterReasonCode () const;

Neden kodunu döndürür.

void setDeadLetterReasonCode (const MQLONG reason);

Alınan harfin neden kodunu belirler.

ImqString destinationQueueManagerName () const;

Hedef kuyruk yöneticisi adını, sondaki boşlukların soyularak döndürür.

void setDestinationQueueManagerAd (const char * ad);

Hedef kuyruk yöneticisi adını belirler. MQ_Q_MGR_NAME_LENGTH (48 karakter) değerinden uzun verileri keser.

ImqString destinationQueueName () const;

Hedef kuyruk adının, sondaki boşluklardan oluşan bir kopyasını döndürür.

void setDestinationQueueName (const char * ad);

Hedef kuyruk adını belirler. MQ_Q_NAME_LENGTH (48 karakter) uzunluğundaki verileri keser.

ImqString putApplicationName () const;

Sondaki boşlukları kaldırarak, koyma uygulaması adının bir kopyasını döndürür.

void setPutApplicationName (const char * ad = 0);

Koyma uygulaması adını belirler. MQ_PUT_APPL_NAME_LENGTH (28 karakter) uzunluğundaki verileri keser.

MQLONG putApplicationTipi () const;

Koyma uygulaması tipini döndürür.

void setPutApplicationType (const MQLONG tip = MQAT_NO_CONTEXT);

Koyma uygulaması tipini ayarlar.

ImqString putDate () const;

Sondaki boşlukları kaldırarak, koyma tarihinin bir kopyasını döndürür.

void setPutDate (const char * date = 0);

Koyma tarihini ayarlar. MQ_PUT_DATE_LENGTH (8 karakter) uzunluğundaki verileri keser.

ImqString putTime () const;

Sondaki boşlukları sökmüş olarak, koyma zamanının bir kopyasını döndürür.

void setPutTime (const char * time = 0);

Koyma süresini ayarlar. MQ_PUT_TIME_LENGTH (8 karakter) değerinden uzun verileri keser.

Nesne verileri (korunan)

MQDLH omqdlh

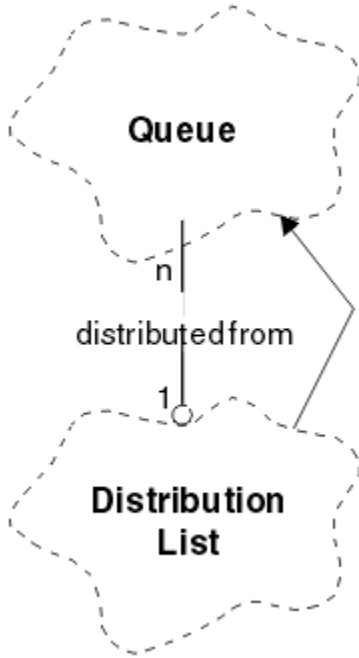
MQDLH veri yapısı.

Neden kodları

- MQRC_TUTARSIZ_BIÇIMI
- MQRC_STRUC_HATA
- MQRC_ENCODING_HATA

ImqDistributionListe C++ sınıfı

Bu sınıf, birden çok hedefe ileti ya da ileti göndermek amacıyla bir ya da daha çok kuyruğa gönderme yapan dinamik bir dağıtım listesini içerir.



Şekil 20. ImqDistributionListe sınıfı

- [“Nesne öznitelikleri” sayfa 1793](#)
- [“Oluşturucular” sayfa 1794](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1794](#)
- [“Nesne yöntemleri \(korunmalı\)” sayfa 1794](#)

Nesne öznitelikleri

ilk dağıtılmış kuyruk

Dağıtım listesi başvurusunun bu nesneyi ele aldığı, sınıfın bir ya da daha çok nesnesinin ilki, belirli bir sıra olmadan.

Başlangıçta böyle bir nesne yoktur. ImqDistributionlistesini başarıyla açmak için bu tür en az bir nesne olmalıdır.

Not: Bir `ImqDistributionListe` nesnesi açıldığında, bu nesneye başvuran tüm açık nesnelere otomatik olarak kapatılır.

Oluşturucular

`ImqDistributionListesi ()`;

Varsayılan oluşturucu.

`ImqDistributionListesi (const ImqDistributionListesi & listesi)`;

Kopya oluşturucu.

Nesne yöntemleri (genel)

`void operator = (const ImqDistributionListe & liste)`;

Kopyalamadan önce **this** nesnesine başvuran tüm nesnelere ilişkin başvurular kalakalır. Bu yöntem çağırıldıktan sonra hiçbir nesne **this** nesnesine başvuramaz.

`* firstDistributedQueue () const` ;

İlk dağıtılmış kuyruğu döndürür.

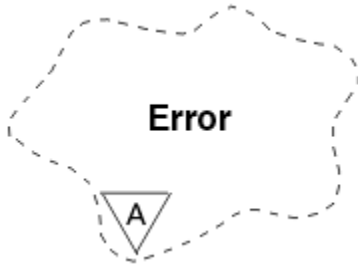
Nesne yöntemleri (korumalı)

`void setFirstDistributedQueue (* kuyruk = 0)`;

İlk dağıtılmış kuyruğu belirler.

ImqError C++ sınıfı

Bu soyut sınıf, bir nesneyle ilişkili hatalarla ilgili bilgi sağlar.



Şekil 21. `ImqError` sınıfı

- [“Nesne öznitelikleri” sayfa 1794](#)
- [“Oluşturucular” sayfa 1795](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1795](#)
- [“Nesne yöntemleri \(korumalı\)” sayfa 1795](#)
- [“Neden kodları” sayfa 1795](#)

Nesne öznitelikleri

tamamlama kodu

En son tamamlama kodu. İlk değer sıfırdır. Aşağıdaki ek değerler kullanılabilir:

- `MQCC_OK`
- `MQCC_UYARISI`
- `MQCC_FAILED`

Neden Kodu

En son neden kodu. İlk değer sıfırdır.

Olusturucular

ImqError();

Varsayılan olusturucu.

ImqError(const ImqError & error);

Kopya olusturucu.

Nesne yontemleri (genel)

void operator = (const ImqError & error);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *hataseçeneğinden* kopyalar.

void clearErrorCodes ();

Tamamlama kodu ve **neden kodu** için her ikisini de sıfır olarak ayarlar.

MQLONG completionCode () const ;

Tamamlama kodunu döndürür.

MQLONG reasonCode () const ;

Neden kodunu döndürür.

Nesne yontemleri (korumalı)

ImqBoolean checkReadPointer (const void * pointer, const size_t length);

İşaretçi ve uzunluk birleşiminin salt okunur erişim için geçerli olduğunu doğrular ve başarılı olursa TRUE değerini döndürür.

ImqBoolean checkWritePointer (const void * pointer, const size_t length);

İşaretçi ve uzunluk birleşiminin okuma-yazma erişimi için geçerli olduğunu doğrular ve başarılı olursa TRUE değerini döndürür.

void setCompletionCode (const MQLONG kod = 0);

Tamamlama kodunu ayarlar.

void setReasonCode (const MQLONG kod = 0);

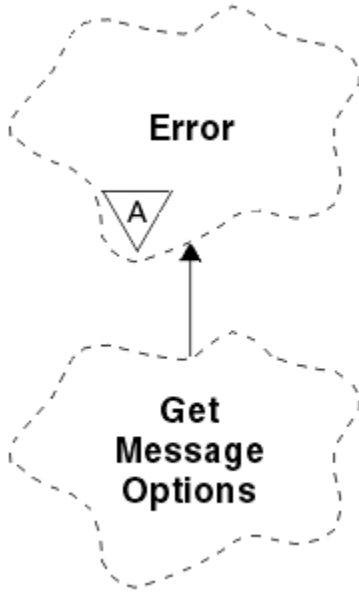
Neden kodu' nı ayarlar.

Neden kodları

- MQRC_BUFFER_ERROR

ImqGetMessageOptions C++ sınıfı

Bu sınıf MQGMO veri yapısını içerir



Şekil 22. *ImqGetMessageOptions* sınıfı

- [“Nesne öznitelikleri” sayfa 1796](#)
- [“Oluşturucular” sayfa 1797](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1797](#)
- [“Nesne yöntemleri \(korunmalı\)” sayfa 1799](#)
- [“Nesne verileri \(korunan\)” sayfa 1799](#)
- [“Neden kodları” sayfa 1799](#)

Nesne öznitelikleri

grup durumu

Bir ileti grubuna ilişkin iletinin durumu. İlk değer MQGS_NOT_IN_GROUP. Aşağıdaki ek değerler kullanılabilir:

- MQGS_MSG_IN_GROUP
- MQGS_LAST_MSG_IN_GROUP

eşleşme seçenekleri

Gelen iletileri seçme seçenekleri. İlk değer: MQMO_MATCH_MSG_ID | MQMO_MATCH_CORREL_ID. Aşağıdaki ek değerler kullanılabilir:

- MQMO_GRUP_Tnt
- MQMO_MATCH_MSG_SEQ_NUMBER
- MQMO_MATCH_OFFSET
- MQMO_MSG_TOKEN
- MQMO_NONE

ileti simgesi

İleti simgesi. MQ_MSG_TOKEN_LENGTH uzunluğunda bir ikili değer (MQBYTE16). İlk değer MQMTOK_NONE.

seçenekler

Bir ileti için geçerli seçenekler. İlk değer MQGMO_NO_WAIT. Aşağıdaki ek değerler kullanılabilir:

- 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_KİLİK
- MQGMO_UNLOCK
- MQGMO_ACCEPT_TRUNCATED_MSG
- MQGMO_SET_SIGNAL
- MQGMO_FAIL_IF QUIESCING
- MQGMO_CONVERT
- MQGMO_LOGICAL_ORDER
- MQGMO_COMPLETE_MSG
- MQGMO_ALL_MSGS_VAR
- MQGMO_ALL_SEGMENTS_VAR
- MQGMO_NONE

çözölmüş kuyruk adı

Çözölmüş kuyruk adı. Bu öznitelik salt okunur. Adlar hiçbir zaman 48 karakterden uzun değildir ve bu uzunluğa boş karakterlerle doldurulabilirler. Başlangıç değeri boş değeri bir dizgi.

döndürölen uzunluk

Döndürölen uzunluk. İlk değeri MQRL_UNDEFINED. Bu öznitelik salt okunur.

kesimlere ayırma

Bir iletiyi bölümlenme yeteneđi. İlk değeri MQSEG_INHIBE. MQSEG_ALLOWED ek değeri mümkün.

bölüm durumu

Bir iletinin bölümlenme durumu. İlk değeri: MQSS_NOT_A_SEGMENT. Aşağıdaki ek değeri kullanılabilir:

- MQSS_SEGMENT
- MQSS_LAST_SEGMENT

eşitleme noktası katılımı

İletiler eşitleme noktası denetimi altında alındığında TRUE olur.

Bekleme Aralığı

Sınıf alma yönteminin, uygun bir iletinin gelmesini beklerken (önceden yoksa) durakladığı süre. Başlangıç değeri sıfırdır, bu da belirsiz bir bekleme işlemini etkiler. MQWI_UNLIMITED ek değeri mümkün. Seçenekler MQGMO_WAIT içermedikçe bu öznitelik yoksayıdır.

Oluşturucular

ImqGetMessageOptions();

Varsayılan oluşturucu.

ImqGetMessageOptions(const ImqGetMessageOptions & gmo);

Kopya oluşturucu.

Nesne yöntemleri (genel)

void işleci = (const ImqGetMessageOptions & gmo);

Var olan eşgörünüm verilerini değıştirerek eşgörünüm verilerini gmo' dan kopyalar.

MQCHAR groupStatus () const;

Grup durumunu döndürür.

void setGroupStatus (const MQCHAR status);

Grup durumunu ayarlar.

MQLONG matchOptions () const;

Eşleşme seçeneklerini döndürür.

void setMatchOptions (const MQLONG options);

Eşleştirme seçeneklerini belirler.

ImqBinary messageToken() const;

İleti simgesini döndürür.

ImqBoolean setMessageToken (const ImqBinary & token);

İleti simgesini ayarlar. *simge* veri uzunluğu sıfır ya da MQ_MSG_TOKEN_LENGTH olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

void setMessageToken (const MQBYTE16 token = 0);

İleti simgesini ayarlar. *simge* sıfır olabilir; bu, MQMTOK_NONE ile aynıdır. *simge* sıfır değilse, ikili verilerin MQ_MSG_TOKEN_LENGTH byte 'larına yanıt vermelidir.

MQMTOK_NONE gibi önceden tanımlanmış değerleri kullanırken, imza eşleşmesini sağlamak için tip dönüşümü yapmanız gerekemeyebilir; örneğin, (MQBYTE *) MQMTOK_NONE.

MQLONG seçenekleri () const;

Seçenekleri döndürür.

void setOptions (const MQLONG seçenekleri);

Eşitleme noktası katılım değeri de içinde olmak üzere seçenekleri ayarlar.

ImqString resolvedQueueName () const;

Çözülen kuyruk adının bir kopyasını döndürür.

MQLONG returnedLength() const;

Döndürülen uzunluğu döndürür.

MQCHAR bölümlenmesi () const;

Bölümleme değerini döndürür.

void setSegmentation (const MQCHAR value);

Segmentasyonu ayarlar.

MQCHAR segmentStatus () const;

Bölüm durumunu döndürür.

void setSegmentStatus (const MQCHAR status);

Bölüm durumunu ayarlar.

ImqBoolean syncPointKatılım () const;

Seçenekler MQGMO_SYNCPOINT ya da MQGMO_SYNCPOINT_IF_PERSISTENT değerini içeriyorsa TRUE olan eşitleme noktası katılım değerini döndürür.

void setSyncPointParticipation (const ImqBoolean sync);

Eşitleme noktası katılım değerini ayarlar. *sync* TRUE ise, MQGMO_SYNCPOINT 'i dahil etmek ve hem MQGMO_NO_SYNCPOINT 'i hem de MQGMO_SYNCPOINT_IF_PERSISTENT 'i dışlamak için seçenekleri değiştirir. *sync* FALSE ise, MQGMO_NO_SYNCPOINT 'i dahil etmek ve hem MQGMO_SYNCPOINT 'i hem de MQGMO_SYNCPOINT_IF_PERSISTENT 'i dışlamak için seçenekleri değiştirir.

MQLONG waitInterval () const;

Bekleme aralığını döndürür.

void setWaitInterval (const MQLONG interval);

Bekleme aralığını ayarlar.

Nesne yöntemleri (korumalı)

static void setVersionSupported (const MQLONG);

MQGMÖ sürümünü ayarlar. Varsayılan deęer: MQGMÖ_VERSION_3.

Nesne verileri (korunan)

MQGMÖ omqgmo

MQGMÖ Sürüm 2 veri yapısı. Yalnızca MQGMÖ_VERSION_2 için desteklenen MQGMÖ alanlarına erişin.

PMQGMÖ opgmo

MQGMÖ veri yapısının adresi. Bu adresin sürüm numarası *olVersion* içinde gösterilir. MQGMÖ alanlarına erişmeden önce sürüm numarasını inceleyin ve bunların var olduğundan emin olun.

MQLONG olVersion

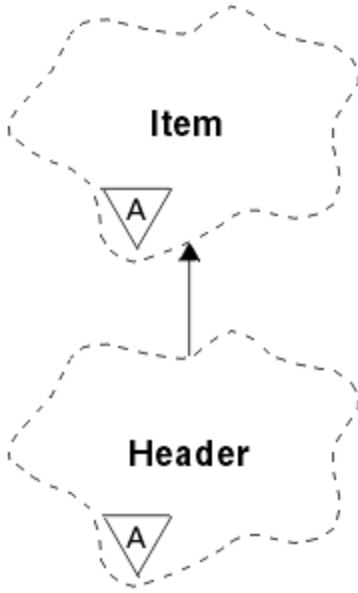
opgmo tarafından adreslenen MQGMÖ veri yapısının sürüm numarası.

Neden kodları

- MQRC_BINARY_DATA_LENGTH_HATA

ImqHeader C++ sınıfı

Bu soyut sınıf, MQDLH veri yapısının ortak özelliklerini içerir.



Şekil 23. ImqHeader sınıfı

- [“Nesne öznitelikleri” sayfa 1799](#)
- [“Oluşturucular” sayfa 1800](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1800](#)

Nesne öznitelikleri

karakter kümesi

Özgün kodlanmış karakter takımı tanıtıcısı. Başlangıçta MQCCSI_Q_MGR.

Kodlama

Orijinal kodlama. Başlangıçta MQENC_NATIVE.

biçim

Özgün biçim. Başlangıçta MQFMT_NONE.

üstbilgi işaretleri

İlk değerler şunlardır:

- ImqDeadLetterHeader sınıfındaki nesnelere için sıfır
- ImqIMSBridgeHeader sınıfının nesnelere için MQIIH_NONE
- ImqReferenceüstbilgi sınıfındaki nesnelere için MQRMHF_LAST
- ImqCICSBridgeHeader sınıfının nesnelere için MQCIH_NONE
- ImqWorkÜstbilgi sınıfının nesnelere için MQWIH_NONE

Oluşturucular

ImqHeader();

Varsayılan oluşturucu.

ImqHeader(const ImqHeader & üstbilgisi);

Kopya oluşturucu.

Nesne yöntemleri (genel)

void operator = (const ImqHeader & header);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *üstbilgiseçeneğinden* kopyalar.

sanal MQLONG characterSet () const ;

Karakter kümesi' ni döndürür.

virtual void setCharacterSet (const MQLONG ccsid = MQCCSI_Q_MGR);

Karakter kümesi' ni ayarlar.

sanal MQLONG kodlama () const ;

kodlamadeğerini döndürür.

virtual void setEncoding (const MQLONG kodlama = MQENC_NATIVE);

Kodlamayı ayarlar.

sanal ImqString format () const ;

Sondaki boşluklar da içinde olmak üzere **format'** in bir kopyasını döndürür.

virtual void setFormat (const char * ad = 0);

Biçim' i, sondaki boşluklarla 8 karaktere doldurarak ayarlar.

sanal MQLONG headerFlags () const ;

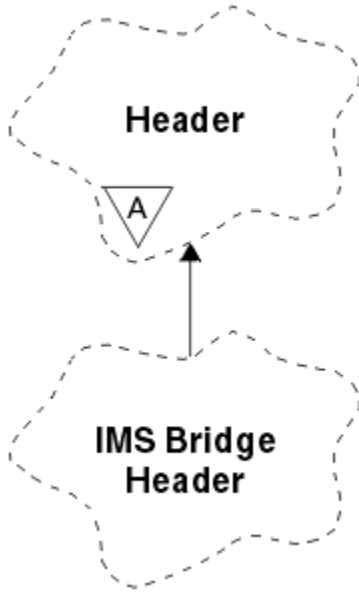
üstbilgi işaretlerini döndürür.

virtual void setHeaderİşaretler (const MQLONG işaretler = 0);

Üstbilgi işaretlerini ayarlar.

ImqIMSBridgeHeader C++ sınıfı

Bu sınıf, MQIIH veri yapısının özelliklerini içerir.



Şekil 24. *ImqIMSBridgeHeader* sınıfı

Bu sınıftaki nesnelere, IBM MQ for z/OS aracılığıyla IMS köprüsüne ileti gönderen uygulamalar tarafından kullanılır.

Not: *ImqHeader* karakter kümesi ve kodlaması varsayılan değerlere sahip olmalı ve başka bir değere ayarlanmamalıdır.

- “Nesne öznitelikleri” sayfa 1801
- “Oluşturucular” sayfa 1802
- “*ImqItem* yöntemleri aşırı yüklendi” sayfa 1802
- “Nesne yöntemleri (genel)” sayfa 1802
- “Nesne verileri (korunan)” sayfa 1803
- “Neden kodları” sayfa 1803

Nesne öznitelikleri

kimlik doğrulayıcı

MQ_AUTHENTICATOR_LENGTH uzunluğundaki RACF parolası ya da geçiş bileti. İlk değer *MQIAUT_NONE*.

kesinleştirme kipi

Kesinleştirme kipi. IMS kesinleştirme kiplerine ilişkin ek bilgi için *OTMA User's Guide* adlı yayına bakın. İlk değer *MQICM_COMMIT_THEN_SEND*. Ek değer (*MQICM_SEND_THEN_COMMIT*) olabilir.

mantıksal uçbirim geçersiz kılma

MQ_LTERM_OVERRIDE_LENGTH uzunluğundaki mantıksal uçbirim geçersiz kılma değeri. Başlangıç değeri boş değerli bir dizgi.

ileti biçimi hizmetleri eşlemi adı

MQ_MFS_MAP_NAME_LENGTH uzunluğundaki MFS eşlemi adı. Başlangıç değeri boş değerli bir dizgi.

yanıtlama biçimi

MQ_FORMAT_LENGTH uzunluğundaki herhangi bir yanıtın biçimi. İlk değer *MQFMT_NONE*.

güvenlik kapsamı

IMS güvenlik işleminin kapsamı. İlk değer *MQISS_CHECK*. Ek değer olan *MQISS_FULL* olabilir.

hareket eşgörünümü tanıtıcısı

Hareket eşgörünümü tanıtıcısı, *MQ_TRAN_INSTANCE_ID_LENGTH* uzunluklu bir ikili (*MQBYTE16*) değeri. İlk değer *MQITII_NONE*.

hareket durumu

IMS görüşmesinin durumu. İlk değer MQITS_NOT_IN_ETKILEŞİM. MQITS_IN_ETKILEŞİM ek değeri olabilir.

Oluşturucular

ImqIMSBridgeHeader();

Varsayılan oluşturucu.

ImqIMSBridgeHeader(const ImqIMSBridgeHeader & header);

Kopya oluşturucu.

ImqItem yöntemleri aşırı yüklendi

sanal ImqBoolean copyOut (ImqMessage & msg);

Var olan ileti verilerini daha ileriye taşıyarak, başlangıçta ileti arabelleğine bir MQIIH veri yapısı ekler. *msg* biçimini MQFMT_IMS olarak ayarlar.

Daha fazla ayrıntı için üst sınıf yöntemi tanımına bakın.

sanal ImqBoolean pasteIn (ImqMessage & msg);

İleti arabelleğindeki bir MQIIH veri yapısını okur.

Başarılı olmak için *msg* nesnesinin kodlaması MQENC_NATIVE olmalıdır. MQGMO_CONVERT ile MQENC_NATIVE iletilerini alın.

Başarılı olmak için *ImqMessage* biçimi MQFMT_IMS olmalıdır.

Daha fazla ayrıntı için üst sınıf yöntemi tanımına bakın.

Nesne yöntemleri (genel)

void işleci = (const ImqIMSBridgeHeader & header);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *üstbilgiseçeneğinden* kopyalar.

ImqString kimlik doğrulayıcı () const;

MQ_AUTHENTICATOR_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak kimlik doğrulayıcının bir kopyasını döndürür.

void setAuthenticator (const char * ad);

Kimlik doğrulayıcıyı ayarlar.

MQCHAR commitMode () const;

Kesinleştirme kipini döndürür.

void setCommitMode (const MQCHAR kip);

Kesinleştirme kipini ayarlar.

ImqString logicalTerminalOverride () const;

Mantıksal uçbirim geçersiz kılma değerinin bir kopyasını döndürür.

void setLogicalTerminalOverride (const char * override);

Mantıksal uçbirim geçersiz kılma değerini ayarlar.

ImqString messageFormatServicesMapName () const;

İleti biçimi hizmetleri eşlemi adının bir kopyasını döndürür.

void setMessageFormatServicesMapName (const char * ad);

İleti biçimi hizmetleri eşlemi adını belirler.

ImqString replyToFormat () const;

MQ_FORMAT_LENGTH uzunluğuna kadar sondaki boşluklarla doldurularak yanıt biçiminin bir kopyasını döndürür.

void setReplyToFormat (const char * format);

Sondaki boşluklarla doldurularak yanıt biçimini MQ_FORMAT_LENGTH uzunluğuna ayarlar.

MQCHAR securityScope () const;

Güvenlik kapsamını döndürür.

void setSecurityScope (const MQCHAR scope);

Güvenlik kapsamını ayarlar.

ImqBinary transactionInstanceId () const;

Hareket eşgörünümü tanıtıcısının bir kopyasını döndürür.

ImqBoolean setTransactionInstanceId (const ImqBinary & id);

Hareket eşgörünümü tanıtıcısını ayarlar. *simge* veri uzunluğu sıfır ya da

MQ_TRAN_INSTANCE_ID_LENGTH olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

void setTransactionInstanceId (const MQBYTE16 id = 0);

Hareket eşgörünümü tanıtıcısını ayarlar. *tanıtıcı* sıfır olabilir; bu, MQITII_NONE ile aynıdır. *id* sıfır değilse, ikili verilerin MQ_TRAN_INSTANCE_ID_LENGTH byte 'larına yanıt vermelidir. MQITII_NONE gibi önceden tanımlanmış değerleri kullanırken, imza eşleşmesini sağlamak için bir tip dönüşümü yapmanız gerekebilir; örneğin, (MQBYTE *) MQITII_NONE.

MQCHAR transactionState () const;

Hareket durumunu döndürür.

void setTransactionState (const MQCHAR state);

Hareket durumunu ayarlar.

Nesne verileri (korunan)

MQIIH omqiih

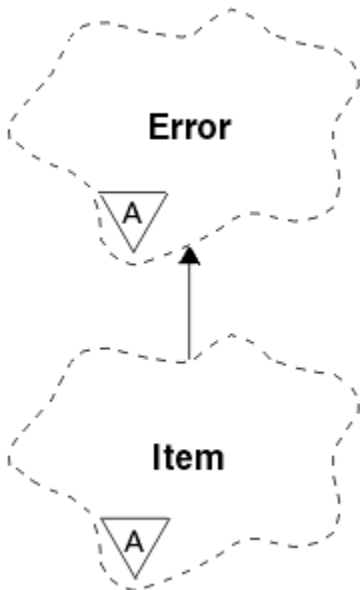
MQIIH veri yapısı.

Neden kodları

- MQRC_BINARY_DATA_LENGTH_HATA
- MQRC_TUTARSIZ_BIÇIMI
- MQRC_ENCODING_HATA
- MQRC_STRUC_HATA

ImqItem C++ sınıfı

Bu soyut sınıf, bir ileti içindeki bir öğeyi (belki de birkaç öğeden birini) temsil eder.



Şekil 25. ImqItem sınıfı

Öğeler bir ileti arabelleğinde birbirine bitiştilir. Her uzmanlık, bir yapı tanıtıcısıyla başlayan belirli bir veri yapısıyla ilişkilendirilir.

Bu soyut sınıftaki çok biçimli yöntemler, öğelerin iletilere ve iletilerden kopyalanmasına izin verir. ImqMessage sınıfı **readItem** ve **writeItem** yöntemleri, uygulama programları için daha doğal olan bu çok biçimli yöntemleri çağırmak için başka bir stil sağlar.

- “Nesne öznitelikleri” sayfa 1804
- “Oluşturucular” sayfa 1804
- “Sınıf yöntemleri (genel)” sayfa 1804
- “Nesne yöntemleri (genel)” sayfa 1804
- “Neden kodları” sayfa 1805

Nesne öznitelikleri

yapı tanıtıcısı

Veri yapısının başında dört karakterlik bir dizgi. Bu öznitelik salt okunur. Türetilmiş sınıflar için bu özniteliği dikkate alın. Otomatik olarak dahil edilmez.

Oluşturucular

ImqItem();

Varsayılan oluşturucu.

ImqItem(const ImqItem & item);

Kopya oluşturucu.

Sınıf yöntemleri (genel)

statik ImqBoolean structureId (const char * structure-id-to-test, const ImqMessage & msg);

Gelen *iletideki* sonraki ImqItem öğesinin **yapı tanıtıcısı** *yapı-tanıtıcısı-test* ile aynıysa TRUE değerini döndürür. Sonraki öğe, ImqCache **veri işaretçisi** tarafından adreslenen ileti arabelleğinin o parçası olarak tanımlanır. Bu yöntem **yapı tanıtıcısına** dayanır ve bu nedenle tüm ImqItem türetilmiş sınıflar için çalışacağı garanti edilmez.

Nesne yöntemleri (genel)

void operator = (const ImqItem & item);

Var olan eşgörünüm verilerinin yerine, *öge* öğesinden eşgörünüm verilerini kopyalar.

sanal ImqBoolean copyOut (ImqMessage & msg) = 0;

Bu nesneyi giden ileti arabelleğinde sonraki öğe olarak yazar ve var olan öğelerin sonuna ekler. Yazma işlemi başarılı olursa, ImqCache **veri uzunluğunu** artırır. Bu yöntem başarılı olursa TRUE değerini döndürür.

Belirli bir alt sınıfla çalışmak için bu yöntemi geçersiz kılın.

sanal ImqBoolean pasteIn (ImqMessage & msg) = 0;

Bu nesneyi gelen ileti arabelleğinden *yıkıcı* olarak okur. Okuma işlemi, ImqCache **veri işaretçisinin** taşınmasında yıkıcı olur. Ancak, arabellek içeriği aynı kalır; böylece veriler ImqCache **veri işaretçisi** sıfırlanarak yeniden okunabilir.

Bu nesnenin (alt) sınıfı, *msg* nesnesinin ileti arabelleğinde bulunan **yapı tanıtıcısı** ile tutarlı olmalıdır.

msg nesnesinin **kodlaması** MQENC_NATIVE olmalıdır. İletilerin ImqMessage **kodlaması** MQENC_NATIVE olarak ve MQGMO_CONVERT de içinde olmak üzere ImqGetMessageOptions **seçenekleri** ile alınması önerilir.

Okuma işlemi başarılı olursa, ImqCache **veri uzunluğu** azaltılır. Bu yöntem başarılı olursa TRUE değerini döndürür.

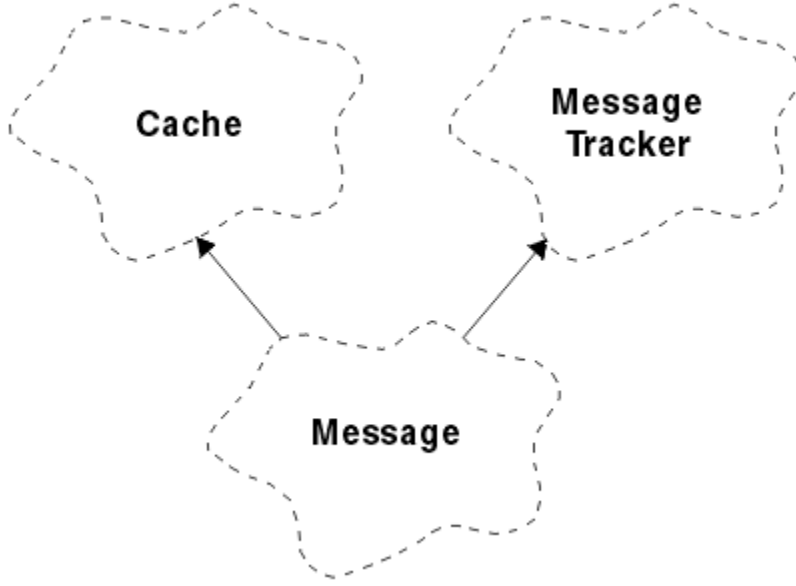
Belirli bir alt sınıfla çalışmak için bu yöntemi geçersiz kılın.

Neden kodları

- MQRC_ENCODING_HATA
- MQRC_STRUC_HATA
- MQRC_TUTARSIZ_BIÇIMI
- MQRC_INSUYETERINCE arabellek
- MQRC_VERIMSIZ_VERI

ImqMessage C++ sınıfı

Bu sınıf bir MQMD veri yapısını kapsıyor ve ileti verilerinin oluşturulmasını ve yeniden oluşturulmasını işleyecek.



Şekil 26. ImqMessage sınıfı

- [“Nesne öznitelikleri” sayfa 1805](#)
- [“Oluşturucular” sayfa 1809](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1809](#)
- [“Nesne yöntemleri \(korunmalı\)” sayfa 1811](#)
- [“Nesne verileri \(korunan\)” sayfa 1811](#)

Nesne öznitelikleri

Uygulama Tnt Verisi

Bir iletiyle ilişkili kimlik bilgileri. Başlangıç değeri boş değerli bir dizgi.

Uygulama Kökeni Verisi

Bir iletiyle ilişkili başlangıç bilgileri. Başlangıç değeri boş değerli bir dizgi.

Geriletme Sayısı

Bir iletinin geçici olarak alınma ve daha sonra geriletilme sayısı. İlk değer sıfırdır. Bu öznitelik salt okunur.

karakter kümesi

Kodlanmış Karakter Takımı Tanıtıcısı. İlk değer MQCCSI_Q_MGR 'dir. Aşağıdaki ek değerler kullanılabilir:

- MQCCSI_INHERIT
- MQCCSI_EMBEDDED

Ayrıca, seçtiğiniz bir Kodlanmış Karakter Kümesi Tanıtıcısı da kullanabilirsiniz. Bu konuda bilgi için bkz. [“Kod sayfası dönüştürme” sayfa 926.](#)

Kodlama

İleti verilerinin makine kodlaması. İlk değer MQENC_NATIVE.

Son kullanma tarihi

IBM MQ 'in alınmamış bir iletiyi atmadan önce ne kadar süre alıkoyduğunu denetleyen zamana bağlı bir miktar. İlk değer MQEI_UNLIMITED.

biçim

Arabellekteki verilerin düzenini tanımlayan biçimin (şablon) adı. Sekiz karakterden uzun adlar sekiz karaktere kesilir. Adlar her zaman sekiz karaktere kadar boşluklarla doldurulur. İlk değişmez değeri MQFMT_NONE. Aşağıdaki ek sabitler olabilir:

- 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

Ayrıca, seçtiğiniz uygulamaya özgü bir dizgiyi de kullanabilirsiniz. Bu konuda daha fazla bilgi için, ileti tanımlayıcısının (MQMD) [“MQMD için biçim \(MQCHAR8\)” sayfa 444](#) alanına bakın.

İleti İşaretleri

Bölümlene denetimi bilgileri. İlk değer MQMF_SEGMENTATION_INHIBE. Aşağıdaki ek değerler kullanılabilir:

- MQMF_SEGMENTATION_ALLOWED
- MQMF_MSG_IN_GROUP
- MQMF_LAST_MSG_IN_GROUP
- MQMF_SEGMENT
- MQMF_LAST_SEGMENT
- MQMF_NONE

ileti tipi

Bir mesajın geniş kategorilere ayrılması. İlk değer MQMT_DATAGRAM. Aşağıdaki ek değerler kullanılabilir:

- MQMT_SYSTEM_FIRST
- MQMT_SYSTEM_LAST
- MQMT_DATAGRAM

- MQMT_REQUEST
- MQMT_REPLY
- MQMT_REPORT
- MQMT_APPL_FIRST
- MQMT_APPL_LAST

Ayrıca, seçtiğiniz uygulamaya özgü bir değeri de kullanabilirsiniz. Bu konuda daha fazla bilgi için, ileti tanımlayıcısının (MQMD) [“MQMD için MsgType \(MQLONG\)” sayfa 435](#) alanına bakın.

offset

Görelî konum bilgileri. İlk değeri sıfırdır.

Özgün Uzunluk

Bölümlenmiş iletinin özgün uzunluğu. İlk değeri MQOL_UNDEFINED.

Kalıcılık

İletinin önemli olduğunu ve sürekli saklama alanı kullanılarak her zaman yedeklenmesi gerektiğini belirtir. Bu seçenek bir performans cezası anlamına gelir. İlk değeri: MQPER_PERSISTENCE_AS_Q_DEF. Aşağıdaki ek değerler kullanılabilir:

- MQPER_KALICI
- MQPER_NOT_KALICI

öncelik

İletim ve teslim için görelî öncelik. Aynı önceliğe sahip iletiler genellikle sağlandıkları sırayla teslim edilir (bunu garanti etmek için yerine getirilmesi gereken birkaç ölçüt olmasına rağmen). İlk değeri: MQPRI_PRIORITY_AS_Q_DEF.

özellik geçerlilik denetimi

İletinin bir özelliği ayarlandığında özelliklerin doğrulanmasının gerçekleşip gerçekleşmeyeceğini belirtir. İlk değeri **MQCMHO_DEFAULT_VALIDATION**. Aşağıdaki ek değerler kullanılabilir:

- MQCMHO_VALIDATE
- MQCMHO_NO_VALIDATION

Aşağıdaki yöntemler **özellik geçerlilik denetimi** üzerinde işlem yapar:

MQLONG propertyValidation() const;

Özellik geçerlilik denetimi seçeneğini döndürür.

void setPropertyValidation (const MQLONG seçeneği);

Özellik geçerlilik denetimi seçeneğini belirler.

Koyma Uygulaması Adı

Bir ileti koyan uygulamanın adı. Başlangıç değeri boş değerli bir dizgi.

Koyma Uygulaması Tipi

Bir ileti koyan uygulamanın tipi. İlk değeri MQAT_NO_CONTEXT. Aşağıdaki ek değerler kullanılabilir:

- 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_BILINMIYOR
- MQAT_USER_FIRST
- MQAT_USER_LAST

Ayrıca, seçtiğiniz uygulamaya özgü bir dizgiyi de kullanabilirsiniz. Bu konuda daha fazla bilgi için, ileti tanımlayıcısının (MQMD) [“MQMD için PutApplTipi \(MQLONG\)”](#) sayfa 458 alanına bakın.

koyma tarihi

İletinin konduğu tarih. Başlangıç değeri boş değerli bir dizgi.

koyma süresi

Bir iletinin konduğu zaman. Başlangıç değeri boş değerli bir dizgi.

yanıtın gönderileceği kuyruk yöneticisi adı

Herhangi bir yanıtın gönderilmesi gereken kuyruk yöneticisinin adı. Başlangıç değeri boş değerli bir dizgi.

yanıtın gönderileceği kuyruk adı

Herhangi bir yanıtın gönderilmesi gereken kuyruğun adı. Başlangıç değeri boş değerli bir dizgi.

rapor

Bir iletiyle ilişkili geribildirim bilgileri. İlk değer MQRO_NONE. Aşağıdaki ek değerler kullanılabilir:

- MQRO_EXCEPTION
- MQRO_EXCEPTION_WITH_DATA
- MQRO_EXCEPTION_WITH_FULL_DATA *
- MQRO_SÜRE SONU
- 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

Burada *, IBM MQ for z/OSüzerinde desteklenmeyen değerleri gösterir.

Sıra Numarası

Grup içindeki bir iletiyi tanıtan sıra bilgileri. İlk değer birdir.

toplam ileti uzunluđu

Bir iletiyi en son okuma giriřimi sırasında kullanılabilir bayt sayısı. Bu sayı, son ileti kesildiyse ya da kesme gerekleřeceği için son ileti okunmamıřsa, ImqCache **ileti uzunluđundan** büyük olur. Bu öznitelik salt okunur. İlk deđer sıfırdır.

Bu öznitelik, kesilmiş iletileri içeren herhangi bir durumda yararlı olabilir.

kullanıcı kimliđi

Bir iletiyle iliřkili kullanıcı kimliđi. Bařlangı deđerı boş deđerli bir dizgi.

Oluřturucular

ImqMessage();

Varsayılan oluřturucu.

ImqMessage(const ImqMessage & msg);

Kopya oluřturucu. Ayrıntılar için **operator** = yöntemine bakın.

Nesne yöntemleri (genel)

void operator = (const ImqMessage & msg);

MQMD 'yi ve ileti verilerini *msg*' den kopyalar. Kullanıcı tarafından bu nesne için bir arabellek sađlandıysa, kopyalanan veri miktarı, kullanılabilir arabellek büyüklüđüyle sınırlıdır. Ters durumda, sistem, kopyalanan veriler için yeterli büyüklükte bir arabelleđin kullanılabilir olmasını sađlar.

ImqString applicationIdData () const ;

Uygulama tanıtıcısı verilerinin bir kopyasını döndürür.

void setApplicationIdData (const char * data = 0);

Uygulama tanıtıcısı verilerini ayarlar.

ImqString applicationOriginData () const ;

Uygulama kaynađı verilerinin bir kopyasını döndürür.

void setApplicationOriginData (const char * data = 0);

Uygulama bařlangı verilerini ayarlar.

MQLONG backoutCount () const ;

geriletme sayısını döndürür.

MQLONG characterSet () const ;

Karakter kümesi' ni döndürür.

void setCharacterSet (const MQLONG ccsid = MQCCSI_Q_MGR);

Karakter kümesi' ni ayarlar.

MQLONG kodlama () const ;

kodlama deđerini döndürür.

void setEncoding (const MQLONG kodlama = MQENC_NATIVE);

Kodlamayı ayarlar.

MQLONG süre bitimi () const ;

süre bitimini döndürür.

void setExpiry (const MQLONG expiry);

süre bitimini ayarlar.

ImqString format () const ;

Sondaki boşluklar da içinde olmak üzere **format'** in bir kopyasını döndürür.

ImqBoolean formatIs (const char * format-to-test) const ;

format , *format-to-test* ile aynıysa TRUE deđerini döndürür.

void setFormat (const char * ad = 0);

Biim' i, sondaki boşluklarla sekiz karaktere kadar doldurarak ayarlar.

MQLONG messageFlags () const ;

ileti iřaretlerini döndürür.

void setMessageİşaretler (const MQLONG işaretler);
İleti işaretlerini ayarlar.

MQLONG messageType () const ;
ileti tipini döndürür.

void setMessageType (const MQLONG type);
İleti tipini ayarlar.

MQLONG offset () const ;
offset değerini döndürür.

void setOffset (const MQLONG offset);
Görelî konumu belirler.

MQLONG originalLength () const ;
Özgün uzunluk değerini döndürür.

void setOriginalLength (const MQLONG length);
Özgün uzunluğu belirler.

MQLONG persistence () const ;
persistence (kalıcı saklama) değerini döndürür.

void setPersistence (const MQLONG persistence);
Kalıcı saklama' yı ayarlar.

MQLONG öncelik () const ;
Öncelik değerini döndürür.

void setPriority (const MQLONG öncelik);
Öncelik değerini ayarlar.

ImqString putApplicationAd () const ;
koyma uygulaması adının bir kopyasını döndürür.

void setPutApplicationName (const char * ad = 0);
koyma uygulaması adını ayarlar.

MQLONG putApplicationTipi () const ;
koyma uygulaması tipini döndürür.

void setPutApplicationType (const MQLONG tip = MQAT_NO_CONTEXT);
koyma uygulaması tipini ayarlar.

ImqString putDate () const ;
koyma tarihinin bir kopyasını döndürür.

void setPutDate (const char * date = 0);
koyma tarihini ayarlar.

ImqString putTime () const ;
koyma süresi' nin bir kopyasını döndürür.

void setPutTime (const char * time = 0);
koyma süresini ayarlar.

ImqBoolean readItem (ImqItem & item);
ImqItem **pasteIn** yöntemini kullanarak ileti arabelleğindeki *item* nesnesini okur. Başarılı olursa TRUE değerini döndürür.

ImqString replyToQueueManagerName () const ;
Yanıt kuyruğu yöneticisi adının bir kopyasını döndürür.

void setReplyToQueueManagerName (const char * name = 0);
Yanıt kuyruğu yöneticisi adını belirler.

ImqString replyToQueueName () const ;
yanıt kuyruğu adının bir kopyasını döndürür.

void setReplyToQueueAdı (const char * ad = 0);
Yanıt kuyruğu adını belirler.

MQLONG rapor () const ;
rapor döndürür.

void setReport (const MQLONG rapor);
raporu ayarlar.

MQLONG sequenceNumber () const ;
sıra numarasını döndürür.

void setSequenceNumber (const MQLONG number);
Sıra numarası' nı ayarlar.

size_t totalMessageLength () const ;
Toplam ileti uzunluğunu döndürür.

ImqString userId () const ;
Kullanıcı kimliği' nin bir kopyasını döndürür.


void setUserId (const char * id = 0);
Kullanıcı kimliğini ayarlar.


ImqBoolean writeItem (ImqItem & item);
ImqItem **copyOut** yöntemini kullanarak *item* nesnesinden ileti arabelleğine yazar. Yazma, ekleme, değiştirme ya da ekleme şeklinde olabilir. Bu, *öge* nesnesinin sınıfına bağlıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

Nesne yöntemleri (korunmalı)

static void setVersionSupported (const MQLONG);
MQMD sürümünü ayarlar. Varsayılan değer: **MQMD_VERSION_2**.

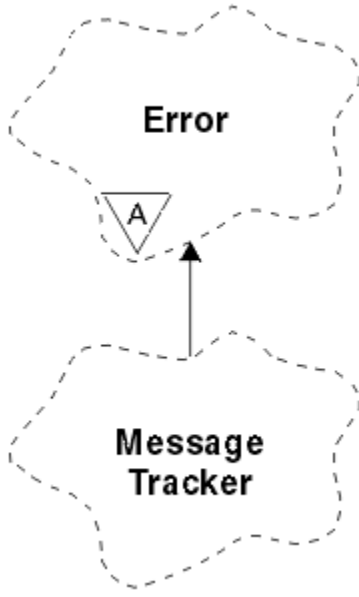
Nesne verileri (korunan)

 **MQMD1 omqmd**
z/OS üzerindeki MQMD veri yapısı.

 **MQMD2 omqmd**
Çoklu platformlar üzerindeki MQMD veri yapısı.

ImqMessageİzleyici C++ sınıfı

Bu sınıf, herhangi bir nesneyle ilişkilendirilebilecek bir ImqMessage ya da ImqQueue nesnesinin özniteliklerini içerir.



Şekil 27. *ImqMessageİzleyici sınıfı*

Bu sınıf, [“ImqMessageİzleyici çapraz başvurusu” sayfa 1763](#) içinde listelenen MQI çağruları ile ilgilidir.

- [“Nesne öznitelikleri” sayfa 1812](#)
- [“Oluşturucular” sayfa 1813](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1813](#)
- [“Neden kodları” sayfa 1814](#)

Nesne öznitelikleri

muhasabe simgesi

MQ_ACCOUNTING_TOKEN_LENGTH uzunluğundaki bir ikili değer (MQBYTE32). İlk değer MQACT_NONE.

İlinti Tanıtıcısı

İletileri ilintilendirmek için atadığınız MQ_CORREL_ID_LENGTH uzunluğundaki ikili değer (MQBYTE24). İlk değer MQCI_NONE. MQCI_NEW_SESSION ek değeri olabilir.

Geribildirim

İletiyi birlikte gönderilecek geribildirim bilgileri. İlk değer MQFB_NONE. Aşağıdaki ek değerler kullanılabilir:

- MQFB_SYSTEM_FIRST
- MQFB_SYSTEM_LAST
- MQFB_APPL_FIRST
- MQFB_APPL_LAST
- MQFB_COA
- MQFB_COD
- MQFB_SÜRE SONU
- MQFB_PAN
- MQFB_NAN
- MQFB_QUIT
- MQFB_DATA_LENGTH_ZERO
- MQFB_DATA_LENGTH_NEGATIF
- MQFB_DATA_LENGTH_TOO_BIG

- MQFB_BUFFER_OVERFLOW
- MQFB_LENGTH_OFF_BY_ONE
- MQFB_IIH_HATA
- MQFB_NOT_AUTHORIZED_FOR_IMS
- MQFB_IMS_HATASI
- MQFB_IMS_FIRST
- MQFB_IMS_SON
- MQFB_CICS_APPL_ABENDED
- MQFB_CICS_APPL_NOT_STARTED
- MQFB_CICS_BRIDGE_FAILURE
- MQFB_CICS_CCSID_HATA
- MQFB_CICS_CIH_HATASI
- MQFB_CICS_COMMAREA_HATA
- MQFB_CICS_CORREL_ID_HATA
- MQFB_CICS_DLQ_HATA
- MQFB_CICS_ENCODING_ERROR
- MQFB_CICS_INTERNAL_HATA
- MQFB_CICS_NOT_YETKILI
- MQFB_CICS_UOW_BACKED_OUT
- MQFB_CICS_UOW_HATASI

Ayrıca, seçtiğiniz uygulamaya özgü bir dizgiyi de kullanabilirsiniz. Bu konuda daha fazla bilgi için, ileti tanımlayıcısının (MQMD) [“MQMD için geribildirim \(MQLONG\)”](#) sayfa 439 alanına bakın.

grup tanıtıcısı

MQ_GROUP_ID_LENGTH uzunluğundaki ikili değer (MQBYTE24) bir kuyruk içinde benzersiz. İlk değer MQGI_NONE.

İleti Tnt

Bir kuyruk içinde benzersiz MQ_MSG_ID_LENGTH uzunluğundaki ikili değer (MQBYTE24). İlk değer MQMI_NONE.

Oluşturucular

ImqMessageTracker ();

Varsayılan oluşturucu.

ImqMessageİzleyici (const ImqMessageİzleyici & izleyici);

Kopya oluşturucu. Ayrıntılar için **operator =** yöntemine bakın.

Nesne yöntemleri (genel)

void operator = (const ImqMessageTracker & tracker);

Var olan eşgörünüm verilerini değiştirerek *izleyici*' den eşgörünüm verilerini kopyalar.

ImqBinary accountingToken () const ;

Muhasebe simgesi' nin bir kopyasını döndürür.

ImqBoolean setAccountingToken (const ImqBinary & token);

Muhasebe simgesiniayarlar. *simge* için **veri uzunluğu** sıfır ya da MQ_ACCOUNTING_TOKEN_LENGTH olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

void setAccountingToken (const MQBYTE32 simge = 0);

Muhasebe simgesiniayarlar. *simge* sıfır olabilir; bu, MQACT_NONE ile aynıdır. *simge* sıfır değilse, ikili verilerin MQ_ACCOUNTING_TOKEN_LENGTH byte 'larına yanıt vermelidir. MQACT_NONE gibi önceden

tanımlanmış değerleri kullanırken, imza eşleşmesini sağlamak için bir tip dönüşümü yapmanız gerekebilir; örneğin, (MQBYTE *) MQACT_NONE.

ImqBinary correlationId () const ;
ilinti tanıtıcısı' nin bir kopyasını döndürür.

ImqBoolean setCorrelationId (const ImqBinary & token);
İlinti tanıtıcısınıayarlar. *token* ile ilgili **veri uzunluğu** sıfır ya da MQ_CORREL_ID_LENGTH olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

void setCorrelationId (const MQBYTE24 id = 0);
İlinti tanıtıcısınıayarlar. *tanıtıcı* sıfır olabilir; bu, MQCI_NONE ile aynıdır. *id* sıfır dışında bir değerse, ikili verilerin MQ_CORREL_ID_LENGTH baytlarını adreslemesi gerekir. MQCI_NONE gibi önceden tanımlanmış değerleri kullanırken, imza eşleşmesini sağlamak için bir tip dönüşümü yapmanız gerekebilir; örneğin, (MQBYTE *) MQCI_NONE.

MQLONG geribildirim () const ;
feedback(geribildirim) değerini döndürür.

void setFeedback (const MQLONG geribildirim);
Geribildirim' i ayarlar.

ImqBinary groupId () const ;
grup tanıtıcısı' nin bir kopyasını döndürür.

ImqBoolean setGroupId (const ImqBinary & token);
Grup kimliğiniayarlar. *simge* için **veri uzunluğu** sıfır ya da MQ_GROUP_ID_LENGTH olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

void setGroupId (const MQBYTE24 id = 0);
Grup kimliğiniayarlar. *tanıtıcı* sıfır olabilir; bu, MQGI_NONE ile aynıdır. *id* sıfır değilse, ikili verilerin MQ_GROUP_ID_LENGTH byte 'larına yanıt vermelidir. MQGI_NONE gibi önceden tanımlanmış değerleri kullanırken, imza eşleşmesi sağlamak için bir tip dönüşümü yapmanız gerekebilir; örneğin, (MQBYTE *) MQGI_NONE.

ImqBinary messageId () const ;
message id(ileti tanıtıcısı) değerinin bir kopyasını döndürür.

ImqBoolean setMessageId (const ImqBinary & token);
İleti tanıtıcısınıayarlar. *token* değerinin **veri uzunluğu** sıfır ya da MQ_MSG_ID_LENGTH olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

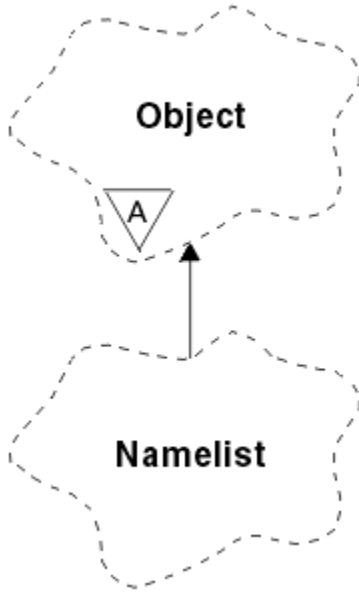
void setMessageId (const MQBYTE24 id = 0);
İleti tanıtıcısınıayarlar. *tanıtıcı* sıfır olabilir; bu, MQMI_NONE ile aynıdır. *id* sıfır dışında bir değerse, ikili verilerin MQ_MSG_ID_LENGTH baytlarını adreslemesi gerekir. MQMI_NONE gibi önceden tanımlanmış değerleri kullanırken, imza eşleşmesini sağlamak için bir tip dönüşümü yapmanız gerekebilir; örneğin, (MQBYTE *) MQMI_NONE.

Neden kodları

- MQRC_BINARY_DATA_LENGTH_HATA

ImqNamelist C++ sınıfı

Bu sınıf bir namelist kapsıyor.



Şekil 28. *ImqNamelist* sınıfı

Bu sınıf, [“ImqNamelist çapraz başvurusu” sayfa 1764](#) içinde listelenen MQI çağruları ile ilgilidir.

- [“Nesne öznitelikleri” sayfa 1815](#)
- [“Oluşturucular” sayfa 1815](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1815](#)
- [“Neden kodları” sayfa 1816](#)

Nesne öznitelikleri

ad sayısı

Ad listesi adlarındaki nesne adlarının sayısı. Bu öznitelik salt okunur.

ad listesi adları

Ad sayısı ile gösterilen nesne adları. Bu öznitelik salt okunur.

Oluşturucular

ImqNamelist();

Varsayılan oluşturucu.

ImqNamelist(const ImqNamelist & list);

Kopya oluşturucu. **ImqObject open status** false değeridir.

ImqNamelist(const char * ad);

ImqObject adını **ad** olarak ayarlar.

Nesne yöntemleri (genel)

void işleci = (const ImqNamelist & list);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *list* listesinden kopyalar. **ImqObject open status** false değeridir.

ImqBoolean nameCount(MQLONG & count);

ad sayısının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG nameCount ();

Olası hataları göstermeden **ad sayısını** döndürür.

ImqBoolean namelistName (const MQLONG index, ImqString & name);

Sıfır tabanlı dizine göre **ad listesi adlarının** bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString namelistName (const MQLONG dizin);

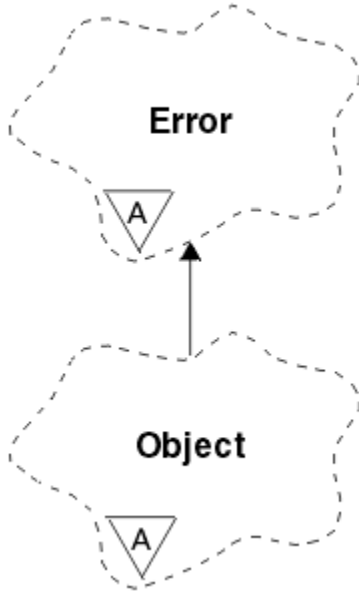
Olası hataları göstermeden, sıfır tabanlı dizine göre **ad listesi adlarından** birini döndürür.

Neden kodları

- MQR_INDEX_HATA
- MQR_INDEX_NOT_PRESENT

ImqObject C++ sınıfı

Bu sınıf soyuttur. Bu sınıftaki bir nesne yok edildiğinde, otomatik olarak kapatılır ve ImqQueueManager bağlantısı kesilir.



Şekil 29. ImqObject sınıfı

Bu sınıf, [“ImqObject çapraz başvurusu”](#) sayfa 1764’inde listelenen MQI çağruları ile ilgilidir.

- [“Sınıf öznitelikleri”](#) sayfa 1816
- [“Nesne öznitelikleri”](#) sayfa 1817
- [“Oluşturucular”](#) sayfa 1818
- [“Sınıf yöntemleri \(genel\)”](#) sayfa 1818
- [“Nesne yöntemleri \(genel\)”](#) sayfa 1818
- [“Nesne yöntemleri \(korunmalı\)”](#) sayfa 1820
- [“Nesne verileri \(korunan\)”](#) sayfa 1821
- [“Neden kodları”](#) sayfa 1821
-

Sınıf öznitelikleri

davranış

Örtük açılış davranışını denetler.

IMQ_IMPL_OPEN (8L)

Örtük açılışa izin verilir. Bu varsayılandır.

Nesne öznitelikleri

değişiklik tarihi

Değişiklik tarihi. Bu öznitelik salt okunur.

değişiklik zamanı

Değişiklik zamanı. Bu öznitelik salt okunur.

diğer kullanıcı kimliği

MQ_USER_ID_LENGTH karakterlerine kadar diğer kullanıcı kimliği. Başlangıç değeri boş değerli bir dizgi.

alternatif güvenlik tanıtıcısı

Diğer güvenlik tanıtıcısı. MQ_SECURITY_ID_LENGTH uzunluğunda bir ikili değer (MQBYTE40). İlk değer MQSID_NONE.

seçenekleri kapat

Bir nesne kapatıldığında geçerli olan seçenekler. İlk değer MQCO_NONE. Bu öznitelik, MQCO_NONE değerinin her zaman kullanıldığı örtük yeniden açma işlemleri sırasında yoksayılr.

bağlantı başvurusu

Bir (yerel) kuyruk yöneticisine gerekli bağlantıyı sağlayan bir ImqQueueManager nesnesine başvuru. Bir ImqQueueManager nesnesi için, nesnenin kendisidir. İlk değer sıfırdır.

Not: Bunu, adlandırılmış bir kuyruk için bir kuyruk yöneticisini (büyük olasılıkla uzak) tanıtan kuyruk yöneticisi adıyla karıştırmayın.

açıklama

Kuyruk yöneticisi, kuyruk, ad listesi ya da işlemin açıklayıcı adı (en çok 64 karakter). Bu öznitelik salt okunur.

Ad

Kuyruk yöneticisi, kuyruk, ad listesi ya da işlemin adı (en çok 48 karakter). Başlangıç değeri boş değerli bir dizgi. Bir model kuyruğunun adı, **open** ' den sonra sonuçtaki dinamik kuyruğun adıyla değişir.

Not: ImqQueueManager, varsayılan kuyruk yöneticisini gösteren boş bir ada sahip olabilir. Ad, başarılı bir açılıştan sonra gerçek kuyruk yöneticisine değişir. ImqDistributionlistesi dinamiktir ve boş değerli bir ada sahip olmalıdır.

sonraki yönetilen nesne

Bu, belirli bir sırada olmayan, bu nesneyle aynı bağlantı başvurusuna sahip olan bu sınıfın sonraki nesnesidir. İlk değer sıfırdır.

seçenekleri aç

Bir nesne açıldığında geçerli olan seçenekler. İlk değer MQOO_INQUIRE. Uygun değerleri ayarlamamanın iki yolu vardır:

1. Açma seçeneklerini ayarlamayın ve açma yöntemini kullanmayın. IBM MQ , açma seçeneklerini otomatik olarak ayarlar ve nesnelere gerektiği şekilde otomatik olarak açar, yeniden açar ve kapatır. Bu, IBM MQ openFor yöntemini kullandığından ve bu yalnızca artımlı olarak açık seçenekleri eklediğinden gereksiz yeniden açma işlemleriyle sonuçlanabilir.
2. MQI çağrısıyla sonuçlanan yöntemleri kullanmadan önce açma seçeneklerini ayarlayın (bkz. [“C++ ve MQI çapraz başvurusu” sayfa 1757](#)). Bu, gereksiz yeniden açma işlemlerinin gerçekleşmemesini sağlar. Olası yeniden açma sorunlarından birinin oluşma olasılığı varsa, açık seçenekleri belirttik olarak ayarlayın (bkz. [Yeniden Açma](#)).

Açma yöntemini kullanıyorsanız, önce açma seçeneklerinin uygun olduğundan emin olmanız gerekir. Ancak, açık yöntemin kullanılması zorunlu değildir; IBM MQ yine de durum 1 ile aynı davranışı sergiler, ancak bu durumda davranış verimlidir.

Sıfır geçerli bir değer değil; nesneyi açmaya çalışmadan önce uygun değeri ayarlayın. Bu, **setOpenOptions** (*IOpenOptions*) ve ardından **open** () ya da **openFor** (*IRequiredOpenOption*) kullanılarak yapılabilir.

Not:

1. MQOO_OUTPUT, dağıtım listesine ilişkin **open** yöntemi sırasında MQOO_INQUIRE yerine konu; MQOO_OUTPUT şu anda tek geçerli **open option** . Ancak, **open** yöntemini kullanan uygulama programlarında MQOO_OUTPUT belirtik olarak ayarlanmalıdır.
2. Sınıfın **resolved queue manager name** ve **resolved queue name** özniteliklerini kullanmak istiyorsanız MQOO_RESOLVE_NAMES değerini belirtin.

açık durum

Nesnenin açık mı (TRUE), yoksa kapalı mı (FALSE) olduğunu belirtir. İlk değer FALSE. Bu öznitelik salt okunur.

önceki yönetilen nesne

Bu sınıfın önceki nesnesi, belirli bir sırada değil, bu nesneyle aynı bağlantı başvurusuna sahip. İlk değer sıfırdır.

kuyruk-yöneticisi-tanıtıcısı

Kuyruk yöneticisi tanıtıcısı. Bu öznitelik salt okunur.

Oluşturucular

ImqObject();

Varsayılan oluşturucu.

ImqObject(const ImqObject & nesnesi);

Kopya oluşturucu. Açık durumu FALSE olacak.

Sınıf yöntemleri (genel)

statik MQLONG davranışı ();

Davranışı döndürür.

void setBehavior(const MQLONG davranış = 0);

Davranışı ayarlar.

Nesne yöntemleri (genel)

void işleci = (const ImqObject & nesne);

Gerekirse bir kapatma işlemi gerçekleştirir ve eşgörünüm verilerini *nesnedenkopyalar*. Açık durumu FALSE olacak.

ImqBoolean alterationDate(ImqString & date);

Değişiklik tarihinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString alterationDate();

Olası hataları göstermeden değişiklik tarihini döndürür.

ImqBoolean alterationTime(ImqString & time);

Değişiklik zamanının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString alterationTime();

Olası hataları göstermeden değişiklik süresini döndürür.

ImqString alternateUserId () const;

Diğer kullanıcı kimliğinin bir kopyasını döndürür.

ImqBoolean setAlternateUserId (const char * id);

Diğer kullanıcı kimliğini ayarlar. Diğer kullanıcı kimliği yalnızca açık durumu FALSE olduğunda ayarlanabilir. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBinary alternateSecurityId () const;

Diğer güvenlik tanıtıcısının bir kopyasını döndürür.

ImqBoolean setAlternateSecurityId(const ImqBinary & token);

Diğer güvenlik tanıtıcısını ayarlar. Diğer güvenlik tanıtıcısı yalnızca açma durumu FALSE olduğunda ayarlanabilir. *simge* veri uzunluğu sıfır ya da MQ_SECURITY_ID_LENGTH olmalıdır. Başarılı olursa TRUE değerini döndürür.

ImqBoolean setAlternateSecurityId(const MQBYTE* simgesi = 0);

Diğer güvenlik tanıtıcısını ayarlar. *simge* sıfır olabilir; bu, MQSID_NONE ile aynıdır. *simge* sıfır değilse, ikili verilerin MQ_SECURITY_ID_LENGTH byte 'larına yanıt vermelidir. MQSID_NONE gibi önceden tanımlanmış değerleri kullanırken, imzanın eşleşmesini sağlamak için bir tip dönüşümü yapmanız gerekebilir; örneğin, (MQBYTE *) MQSID_NONE.

Diğer güvenlik tanıtıcısı yalnızca açık durum TRUE olduğunda ayarlanabilir. Başarılı olursa TRUE değerini döndürür.

ImqBoolean setAlternateSecurityId(const unsigned char * id = 0);

Diğer güvenlik tanıtıcısını ayarlar.

ImqBoolean close ();

Açma durumunu FALSE olarak ayarlar. Başarılı olursa TRUE değerini döndürür.

MQLONG closeOptions () const;

Kapatma seçeneklerini döndürür.

void setCloseOptions (const MQLONG options);

Kapatma seçeneklerini belirler.

ImqQueueManager * connectionReference () const;

Bağlantı başvurusunu döndürür.

void setConnectionReference (ImqQueueManager & manager);

Bağlantı başvurusunu ayarlar.

void setConnectionReference (ImqQueueManager * manager = 0);

Bağlantı başvurusunu ayarlar.

sanal ImqBoolean tanımı (ImqString & tanım) = 0;

Açıklamanın bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString description ();

Olası hataları göstermeden açıklamanın bir kopyasını döndürür.

sanal ImqBoolean adı (ImqString & ad);

Adın bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString ad ();

Olası hataları göstermeden adın bir kopyasını döndürür.

ImqBoolean setName (const char * ad = 0);

Adı belirler. Ad yalnızca açık durumu FALSE olduğunda ve bağlantı durumu FALSE olduğunda ImqQueueManager için ayarlanabilir. Başarılı olursa TRUE değerini döndürür.

ImqObject * nextManagedObject () const;

Sonraki yönetilen nesneyi döndürür.

ImqBoolean open ();

Nesneyi gerektiği gibi açarak, diğer özniteliklerin yanı sıra açma seçeneklerini ve adı kullanarak açma durumunu TRUE olarak değiştirir. Bu yöntem, ImqQueueManager bağlantı durumunun TRUE olduğundan emin olmak için gerekirse, bağlantı başvuru bilgilerini ve ImqQueueManager bağlantı yöntemini kullanır. Açık durumu döndürür.

ImqBoolean openFor (const MQLONG gerekli-seçenekler = 0);

Nesnenin açık seçeneklerle ya da *required-options* parametre değerinin belirttiği davranışı garanti eden açık seçeneklerle açık olduğundan emin olmak için girişimde bulunur.

required-options sıfır ise, giriş gereklidir ve herhangi bir giriş seçeneği yeterlidir. Bu nedenle, açma seçenekleri aşağıdakilerden birini içeriyorsa:

- MQOO_INPUT_AS_Q_DEF
- MQOO_INPUT_SHARED
- MQOO_INPUT_EXCLUSIVE

Açma seçenekleri zaten tatmin edici ve değiştirilmedi; açma seçenekleri bu seçeneklerden herhangi birini içermiyorsa, açma seçeneklerinde MQOO_INPUT_AS_Q_DEF ayarlanır.

required-options sıfır değilse, gerekli seçenekler açık seçeneklere eklenir; *required-options* bu seçeneklerden biriye, diğerleri sıfırlanır.

Açık seçeneklerden herhangi biri değiştirilirse ve nesne zaten açıksa, açık seçenekleri ayarlamak için nesne geçici olarak kapatılır ve yeniden açılır.

Başarılı olursa TRUE değerini döndürür. Başarı, nesnenin uygun seçeneklerle açık olduğunu gösterir.

MQLONG openOptions () const;

Açma seçeneklerini döndürür.

ImqBoolean setOpenSeçenekleri (const MQLONG seçenekler);

Açma seçeneklerini belirler. Açma seçenekleri yalnızca açma durumu FALSE olduğunda ayarlanabilir. Başarılı olursa TRUE değerini döndürür.

ImqBoolean openStatus () const;

Açık durumunu döndürür.

ImqObject * previousManagedObject () const;

Önceki yönetilen nesneyi döndürür.

ImqBoolean queueManagerTanıtıcısı (ImqString & id);

Kuyruk yöneticisi tanıtıcısının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString queueManagerIdentifier ();

Olası hataları göstermeden kuyruk yöneticisi tanıtıcısını döndürür.

Nesne yöntemleri (korumalı)

sanal ImqBoolean closeTemporarily ();

Yeniden açmadan önce bir nesneyi güvenli bir şekilde kapatır. Başarılı olursa TRUE değerini döndürür. Bu yöntem, açık durumun TRUE olduğunu varsayar.

MQHCONN connectionHandle () const;

Bağlantı başvurusuyla ilişkilendirilmiş MQHCONN değerini döndürür. Bağlantı başvurusu yoksa ya da Yönetici bağlı değilse bu değer sıfırdır.

ImqBoolean inquire (const MQLONG int-attr, MQLONG & value);

Dizini MQIA_ * değeri olan bir tamsayı değeri döndürür. Hata olması durumunda, değer MQIAV_UNDEFINED olarak ayarlanır.

ImqBoolean inquire (const MQLONG char-attr, char * & buffer, const size_t length);

Dizini MQCA_ * değeri olan bir karakter dizisi döndürür.

Not: Bu yöntemlerin her ikisi de tek bir öznitelik değeri döndürür. Değerlerin anında birbiriyle tutarlı olduğu bir *anlık görünüm* için birden fazla değer gerekiyorsa, IBM MQ C++ bu olanağı sağlamaz ve MQINQ çağrısının uygun değiştirgelerle kullanılması gerekir.

sanal geçersiz openInformationDisperse ();

MQOPEN çağrısından hemen sonra MQOD veri yapısının değişken bölümündeki bilgileri dağıtır.

sanal ImqBoolean openInformationPrepare ();

MQOD veri yapısının değişken bölümüne ilişkin bilgileri MQOPEN çağrısından hemen önce hazırlar ve başarılı olursa TRUE değerini döndürür.

ImqBoolean set (const MQLONG int-attr, const MQLONG value);

Bir IBM MQ tamsayı özniteliği ayarlar.

ImqBoolean set (const MQLONG char-attr, const char * buffer, const size_t required-length);

Bir IBM MQ karakter özniteliğini ayarlar.

void setNextManagedObject (const ImqObject * object = 0);

Sonraki yönetilen nesneyi ayarlar.

Dikkat: Bu işlevi, yönetilen nesne listesini bozmayacağından eminseniz kullanın.

void setPreviousManagedObject (const ImqObject * object = 0);

Önceki yönetilen nesneyi ayarlar.

Dikkat: Bu işlevi, yönetilen nesne listesini bozmayacağından eminseniz kullanın.

Nesne verileri (korunan)

MQHOBj ohobj

IBM MQ nesne tanıtıcısı (yalnızca açık durum TRUE olduğunda geçerlidir).

MQOD omqod

Yerleşik MQOD veri yapısı. Bu veri yapısı için ayrılan saklama alanı miktarı, bir MQOD Sürüm 2 için gereklidir. (*omqod.Version*) sürüm numarasını inceleyin ve diğer alanlara aşağıdaki gibi erişin:

MQOD_VERSION_1

omqod içindeki diğer tüm alanlara erişilebilir.

MQOD_VERSION_2

omqod içindeki diğer tüm alanlara erişilebilir.

MQOD_VERSION_3

omqod.pmqod , dinamik olarak ayrılmış, daha büyük bir MQOD ' ye ilişkin bir işaretçidir. *omqod* içindeki başka hiçbir alana erişilemez. *omqod.pmqod* ile adreslenen tüm alanlara erişilebilir.

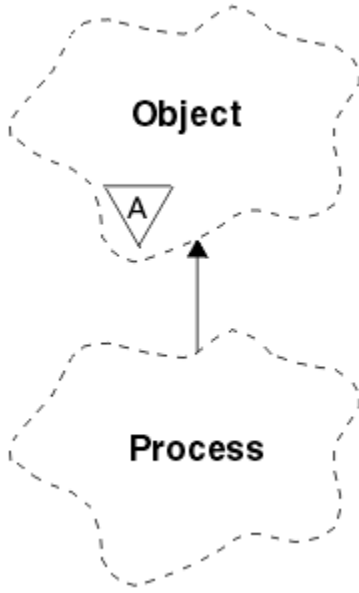
Not: *omqod.pmqod.Version* , *omqod.Version* değerinden küçük olabilir; bu, IBM MQ MQI client ' ın IBM MQ sunucusundan daha fazla işlevselliğe sahip olduğunu gösterir.

Neden kodları

- MQRC_ATTRIBUTE_LOCKED
- MQRC_TUTARSIZ_NESNE_DURUMU
- MQRC_NO_CONNECTION_REFERENCE
- MQRC_STORAGE_KULLANILAMIYOR
- MQRC_REOPEN_SAVED_CONTEXT_ERR
- (MQCLOSE neden kodları)
- (MQCONN ' daki neden kodları)
- (MQINQ ' dan neden kodları)
- (MQOPEN ' den neden kodları)
- (MQSET ' ten neden kodları)

ImqProcess C++ sınıfı

Bu sınıf, bir tetikleyici izleyicisi tarafından tetiklenebilen bir uygulama işlemini (MQOT_PROCESS tipinde bir IBM MQ nesnesi) içerir.



Şekil 30. *ImqProcess* sınıfı

- “Nesne öznitelikleri” sayfa 1822
- “Oluşturucular” sayfa 1822
- “Nesne yöntemleri (genel)” sayfa 1822

Nesne öznitelikleri

Uygulama tanıtıcısı

Uygulama sürecinin tanıtıcısı. Bu öznitelik salt okunur.

Uygulama Tipi

Uygulama sürecinin tipi. Bu öznitelik salt okunur.

ortam verileri

Sürece ilişkin ortam bilgileri. Bu öznitelik salt okunur.

Kullanıcı verileri

Sürece ilişkin kullanıcı verileri. Bu öznitelik salt okunur.

Oluşturucular

ImqProcess();

Varsayılan oluşturucu.

ImqProcess(const ImqProcess & process);

Kopya oluşturucu. ImqObject **açık durumu** : FALSE.

ImqProcess(const char * ad);

ImqObject **adını**ayarlar.

Nesne yöntemleri (genel)

void operator = (const ImqProcess & process);

Gerekirse bir kapatma işlemi gerçekleştirir ve eşgörünüm verilerini *süreçten*kopyalar. ImqObject **open status** FALSE olur.

ImqBoolean applicationId (ImqString & id);

Uygulama tanıtıcısı' nin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString applicationId ();

Olası hataları göstermeden **uygulama tanıtıcısını** döndürür.

ImqBoolean applicationType (MQLONG & type);

Uygulama tipinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG applicationType ();

Olası hataları göstermeden **uygulama tipini** döndürür.

ImqBoolean environmentData (ImqString & data);

Ortam verilerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString environmentData ();

Olası hataları göstermeden **ortam verilerini** döndürür.

ImqBoolean userData (ImqString & data);

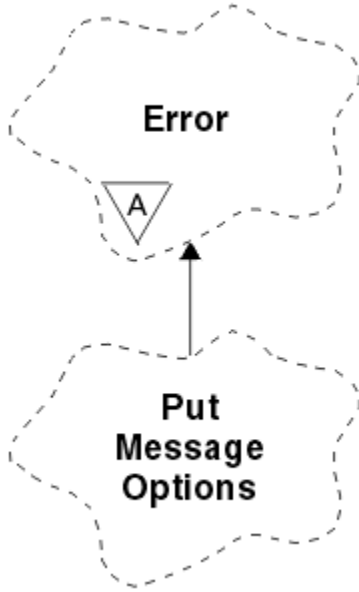
Kullanıcı verilerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString userData ();

Olası hataları göstermeden **kullanıcı verilerini** döndürür.

ImqPutMessageOptions C++ sınıfı

Bu sınıf MQPMO veri yapısını kapsıyor.



Şekil 31. ImqPutMessageOptions sınıfı

- [“Nesne öznitelikleri” sayfa 1823](#)
- [“Oluşturucular” sayfa 1824](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1824](#)
- [“Nesne verileri \(korunan\)” sayfa 1825](#)
- [“Neden kodları” sayfa 1825](#)

Nesne öznitelikleri

bağlam başvurusu

İletiler için bağlam sağlayan bir ImqQueue (ImqQueue). Başlangıçta başvuru yoktur.

seçenekler

Koyma iletisi seçenekleri. İlk değer MQPMO_NONE. Aşağıdaki ek değerler kullanılabilir:

- 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

kayıt alanları

Bir ileti konduğunda koyma iletisi kayıtlarının eklenmesini denetleyen işaretler. İlk değer MQPMRF_NONE. Aşağıdaki ek değerler kullanılabilir:

- MQPMRF_MSG_ID
- MQPMRF_CORREL_ID
- MQPMRF_GROUP_ID
- MQPMRF_FEEDBACK
- MQPMRF_ACCOUNTING_TOKEN

ImqMessageBelirtilen herhangi bir alana ilişkin izleyici öznitelikleri nesneden alınır.

ImqMessageİzleyici öznitelikleri, belirtilmeyen herhangi bir alan için ImqMessage nesnesinden alınır.

çözülmüş kuyruk yöneticisi adı

Bir koyma sırasında belirlenen hedef kuyruk yöneticisinin adı. İlk değer boş. Bu öznitelik salt okunur.

çözülmüş kuyruk adı

Bir koyma işlemi sırasında belirlenen hedef kuyruğun adı. İlk değer boş. Bu öznitelik salt okunur.

eşitleme noktası katılımı

İletiler eşitleme noktası denetimi altına konduğunda TRUE olur.

Oluşturucular

ImqPutMessageOptions();

Varsayılan oluşturucu.

ImqPutMessageOptions(const ImqPutMessageOptions & pmo);

Kopya oluşturucu.

Nesne yöntemleri (genel)

void işleci = (const ImqPutMessageOptions & pmo);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *pmo*' dan kopyalar.

ImqQueue * contextReference () const;

Bağlam başvurusunu döndürür.

void setContextReference (const ImqQueue & queue);

Bağlam başvurusunu ayarlar.

void setContextReference (const ImqQueue * queue = 0);

Bağlam başvurusunu ayarlar.

MQLONG seçenekleri () const;

Seçenekleri döndürür.

void setOptions (const MQLONG seçenekleri);

Eşitleme noktası katılım değeri de içinde olmak üzere seçenekleri ayarlar.

MQLONG recordFields () const;

Kayıt alanlarını döndürür.

void setRecordFields (const MQLONG fields);

Kayıt alanlarını ayarlar.

ImqString resolvedQueueManagerName () const;

Çözülen kuyruk yöneticisi adının bir kopyasını döndürür.

ImqString resolvedQueueName () const;

Çözülen kuyruk adının bir kopyasını döndürür.

ImqBoolean syncPointKatılım () const;

Seçenekler MQPMO_SYNCPOINT içeriyorsa TRUE olan eşitleme noktası katılım değerini döndürür.

void setSyncPointParticipation (const ImqBoolean sync);

Eşitleme noktası katılım değerini ayarlar. *sync* TRUE ise, seçenekler MQPMO_SYNCPOINT 'i içerecek ve MQPMO_NO_SYNCPOINT 'i dışlayacak şekilde değiştirilir. *sync* FALSE ise, seçenekler MQPMO_NO_SYNCPOINT 'i içerecek ve MQPMO_SYNCPOINT 'i dışlayacak şekilde değiştirilir.

Nesne verileri (korunan)**MQPMO omqpmo**

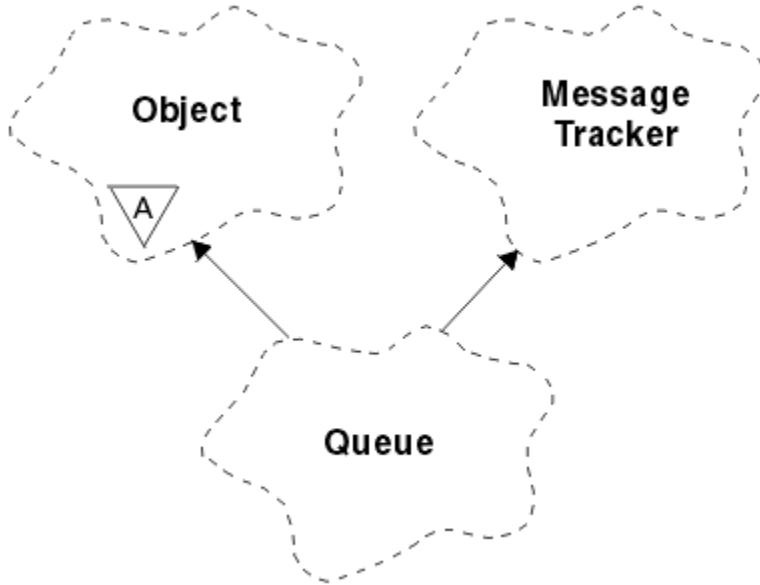
MQPMO veri yapısı.

Neden kodları

- MQRC_STORAGE_KULLANILAMIYOR

ImqQueue C++ sınıfı

Bu sınıf bir ileti kuyruğunu (MQOT_Q tipinde bir IBM MQ nesnesi) içerir.



Şekil 32. ImqQueue sınıfı

Bu sınıf, Çizelge 862 sayfa 1765’inde listelenen MQI çağrıları ile ilgilidir.

- “Nesne öznitelikleri” sayfa 1826
- “Oluşturucular” sayfa 1828
- “Nesne yöntemleri (genel)” sayfa 1829
- “Nesne yöntemleri (korunmalı)” sayfa 1835
- “Neden kodları” sayfa 1835

Nesne öznitelikleri

geriletme yeniden kuyruğa alma adı

Aşırı geriletme yeniden kuyruğa alma adı. Bu öznitelik salt okunur.

Geriletme eşiği

Geriletme eşiği. Bu öznitelik salt okunur.

temel kuyruk adı

Diğer adın çözümleneceği kuyruğun adı. Bu öznitelik salt okunur.

Küme adı

Küme adı. Bu öznitelik salt okunur.

küme ad listesi adı

Küme ad listesi adı. Bu öznitelik salt okunur.

küme iş yükü sırası

Küme iş yükü sırası. Bu öznitelik salt okunur.

küme iş yükü önceliği

Küme iş yükü önceliği. Bu öznitelik salt okunur.

küme iş yükü kullanım kuyruğu

Küme iş yükü kullanım kuyruğu değeri. Bu öznitelik salt okunur.

yaratma tarihi

Kuyruk yaratma verileri. Bu öznitelik salt okunur.

Oluşturulma zamanı

Kuyruk oluşturma zamanı. Bu öznitelik salt okunur.

yürürlükteki derinlik

Kuyruktaki iletilerin sayısı. Bu öznitelik salt okunur.

varsayılan bağ tanımlama

Varsayılan bağ tanımlama. Bu öznitelik salt okunur.

varsayılan giriş açma seçeneği

Varsayılan giriş için aç seçeneği. Bu öznitelik salt okunur.

varsayılan kalıcılık

Varsayılan ileti kalıcılığı. Bu öznitelik salt okunur.

varsayılan öncelik

Varsayılan ileti önceliği. Bu öznitelik salt okunur.

Tanımlama tipi

Kuyruk tanımlaması tipi. Bu öznitelik salt okunur.

derinlik yüksek olayı

Kuyruk derinliği yüksek olayları için denetim özniteliği. Bu öznitelik salt okunur.

derinlik yüksek sınırı

Kuyruk derinliği için üst sınır. Bu öznitelik salt okunur.

derinlik düşük olayı

Kuyruk derinliği düşük olayları için denetim özniteliği. Bu öznitelik salt okunur.

derinlik alt sınırı

Kuyruk derinliği için alt sınır. Bu öznitelik salt okunur.

derinlik üst sınırı olayı

Kuyruk derinliği olay sayısı üst sınırı için denetim özniteliği. Bu öznitelik salt okunur.

dağıtım listesi başvurusu

İletileri bu ileti de içinde olmak üzere birden çok kuyruğa dağıtmak için kullanılabilecek bir ImqDistributionlistesine isteğe bağlı başvuru. İlk değer boş.

Not: Bir ImqQueue nesnesi açıldığında, başvurduğu ImqDistributionliste nesnesi otomatik olarak kapatılır.

dağıtım listeleri

İletim kuyruğunun dağıtım listelerini destekleme yeteneği. Bu öznitelik salt okunur.

dinamik kuyruk adı

Dinamik kuyruk adı. İlk değer AMQ.* tüm AIX, Linux, and Windows platformları için.

sertleşin geri çekilin

Geri sayımı sertleştirip sertleşmeyeceğini. Bu öznitelik salt okunur.

dizin tipi

Dizin tipi. Bu öznitelik salt okunur.

alma engelleme

Alma işlemlerine izin verilip verilmediğini belirler. İlk değer, kuyruk tanımlamasına bağlıdır. Bu öznitelik yalnızca bir diğer ad ya da yerel kuyruk için geçerlidir.

koymayı engelle

Koyma işlemlerine izin verilip verilmediğini belirler. İlk değer, kuyruk tanımlamasına bağlıdır.

başlatma kuyruğu adı

Başlatma kuyruğunun adı. Bu öznitelik salt okunur.

derinlik üst sınırı

Kuyrukta izin verilen ileti sayısı üst sınırı. Bu öznitelik salt okunur.

ileti uzunluğu üst sınırı

Bu kuyruktaki herhangi bir ileti için uzunluk üst sınırı; bu değer, ilişkili kuyruk yöneticisi tarafından yönetilen herhangi bir kuyruk için üst sınırdan az olabilir. Bu öznitelik salt okunur.

ileti teslim sırası

İleti önceliğinin ilgili olup olmadığını belirler. Bu öznitelik salt okunur.

sonraki dağıtılmış kuyruk

Bu sınıfın sonraki nesnesi, belirli bir sırada değil, bu nesneyle aynı **dağıtım listesi başvurusuna** sahip. İlk değer sıfırdır.

Bir zincirdeki bir nesne silinirse, dağıtılmış kuyruk bağlantıları silinen nesneyi artık göstermeyecek şekilde önceki nesne ve sonraki nesne güncellenir.

kalıcı olmayan ileti sınıfı

Bu kuyruğa konan kalıcı olmayan iletiler için güvenilirlik düzeyi. Bu öznitelik salt okunur.

açık giriş sayısı

Giriş için açık olan ImqQueue nesnelerinin sayısı. Bu öznitelik salt okunur.

açık çıkış sayısı

Çıkış için açık olan ImqQueue nesnelerinin sayısı. Bu öznitelik salt okunur.

önceki dağıtılmış kuyruk

Bu sınıfın önceki nesnesi, belirli bir sırada değil, bu nesneyle aynı **dağıtım listesi başvurusuna** sahip. İlk değer sıfırdır.

Bir zincirdeki bir nesne silinirse, dağıtılmış kuyruk bağlantıları silinen nesneyi artık göstermeyecek şekilde önceki nesne ve sonraki nesne güncellenir.

İşlem adı

Süreç tanımlamasının adı. Bu öznitelik salt okunur.

kuyruk muhasebesi

Kuyruklara ilişkin hesap bilgileri düzeyi. Bu öznitelik salt okunur.

kuyruk-yöneticisi-adi

Kuyruğun bulunduğu kuyruk yöneticisinin (büyük olasılıkla uzak) adı. Burada adı belirtilen kuyruk yöneticisini, bağlantı sağlayan (yerel) kuyruk yöneticisine başvuran ImqObject **bağlantı başvurusuylakarıştırmayın**. İlk değer boş.

kuyruk izleme

Kuyruğa ilişkin izleme verileri toplama düzeyi. Bu öznitelik salt okunur.

kuyruk istatistikleri

Kuyruğa ilişkin istatistik verileri düzeyi. Bu öznitelik salt okunur.

kuyruk tipi

Kuyruk tipi. Bu öznitelik salt okunur.

uzak kuyruk yöneticisi adı

Uzak kuyruk yöneticisinin adı. Bu öznitelik salt okunur.

uzak kuyruk adı

Uzak kuyruk yöneticisinde bilinen uzak kuyruğun adı. Bu öznitelik salt okunur.

çözülmüş kuyruk yöneticisi adı

Çözülmüş kuyruk yöneticisi adı. Bu öznitelik salt okunur.

çözülmüş kuyruk adı

Çözülmüş kuyruk adı. Bu öznitelik salt okunur.

alıkoyma aralığı

Kuyruk alıkoyma aralığı. Bu öznitelik salt okunur.

kapsam

Kuyruk tanımlamasının kapsamı. Bu öznitelik salt okunur.

hizmet aralığı

Hizmet aralığı. Bu öznitelik salt okunur.

hizmet aralığı olayı

Hizmet aralığı olayları için denetim özniteliği. Bu öznitelik salt okunur.

paylaşılabilir

Kuyruğun paylaşılıp paylaşılmayacağını belirler. Bu öznitelik salt okunur.

depolama sınıfı

Depolama sınıfı. Bu öznitelik salt okunur.

iletim kuyruğu adı

İletim kuyruğunun adı. Bu öznitelik salt okunur.

tetikleme denetimi

Tetikleme kontrolü. İlk değer, kuyruk tanımlamasına bağlıdır. Bu öznitelik yalnızca yerel kuyruk için geçerlidir.

tetikleyici verileri

Tetikleyici veri. İlk değer, kuyruk tanımlamasına bağlıdır. Bu öznitelik yalnızca yerel kuyruk için geçerlidir.

tetikleyici derinliği

Tetikleme derinliği. İlk değer, kuyruk tanımlamasına bağlıdır. Bu öznitelik yalnızca yerel kuyruk için geçerlidir.

tetikleyici ileti önceliği

Tetikleyiciler için eşik ileti önceliği. İlk değer, kuyruk tanımlamasına bağlıdır. Bu öznitelik yalnızca yerel kuyruk için geçerlidir.

Tetikleme Tipi

Tetikleyici tipi. İlk değer, kuyruk tanımlamasına bağlıdır. Bu öznitelik yalnızca yerel kuyruk için geçerlidir.

kullanıma

Kullanım. Bu öznitelik salt okunur.

Oluşturucular**ImqQueue();**

Varsayılan oluşturucu.

ImqQueue(const ImqQueue & kuyruğu);

Kopya oluşturucu. ImqObject **open status** FALSE olur.

ImqQueue(const char * ad);

ImqObject **adını**ayarlar.

Nesne yöntemleri (genel)

void operator = (const ImqQueue & *kuyruğu*);

Gerekirse bir kapatma işlemi gerçekleştirir ve eşgörünüm verilerini *kuyruktankopyalar*. ImqObject **open status** FALSE olur.

ImqBoolean backoutRequeueAd (ImqString & *ad*);

geriletme yeniden kuyruğa alma adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString backoutRequeueName ();

Olası hataları göstermeden **geriletme yeniden kuyruğa alma** adını döndürür.

ImqBoolean backoutThreshold (MQLONG & *eşik*);

Arka plan eşliğinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG backoutThreshold ();

Olası hataları göstermeden **geriletme eşliği** değerini döndürür.

ImqBoolean baseQueueAdı (ImqString & *ad*);

Temel kuyruk adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString baseQueueAdı ();

Olası hataları göstermeden **temel kuyruk adını** döndürür.

ImqBoolean clusterName(ImqString & *ad*);

Küme adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString clusterName();

Olası hataları göstermeden **küme adını** döndürür.

ImqBoolean clusterNamelistAd (ImqString & *ad*);

Küme ad listesi adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString clusterNamelistName ();

Herhangi bir hata belirtisi göstermeden **küme ad listesi adını** döndürür.

ImqBoolean clusterWorkLoadPriority (MQLONG & *priority*);

Küme iş yükü öncelik değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG clusterWorkLoadPriority ();

Olası hataları göstermeden küme iş yükü öncelik değerini döndürür.

ImqBoolean clusterWorkLoadRank (MQLONG & *rank*);

Küme iş yükü sıra değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG clusterWorkLoadRank ();

Olası hataları göstermeden küme iş yükü sıra değerini döndürür.

ImqBoolean clusterWorkLoadUseQ (MQLONG & *useq*);

Küme iş yükü kullanım kuyruğu değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG clusterWorkLoadUseQ ();

Olası hataları göstermeden küme iş yükü kullanım kuyruğu değerini döndürür.

ImqBoolean creationDate (ImqString & *date*);

Yaratma tarihi' nin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString creationDate ();

Olası hataları göstermeden **oluşturma tarihini** döndürür.

ImqBoolean creationTime (ImqString & *time*);

Yaratma zamanı' nin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString creationTime ();

Olası hataları göstermeden **oluşturma zamanını** döndürür.

ImqBoolean currentDepth (MQLONG & *derinlik*);

Yürürlükteki derinliğin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG currentDepth ();

Olası hataları göstermeden **geçerli derinliği** döndürür.

ImqBoolean defaultInputOpenOption (MQLONG & seçenek);
Varsayılan giriş açma seçeneğinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG defaultInputOpenOption ();
Olası hatalar belirtilmeden varsayılan giriş açma seçeneğini döndürür.

ImqBoolean defaultPersistence (MQLONG & persistence);
Varsayılan kalıcılığın bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG defaultPersistence ();
Olası hataları göstermeden varsayılan kalıcı saklama değerini döndürür.

ImqBoolean defaultPriority (MQLONG & öncelik);
Varsayılan önceliğin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG defaultPriority ();
Olası hataları göstermeden varsayılan önceliği döndürür.

ImqBoolean defaultBind (MQLONG & bind);
Varsayılan bağ tanımının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG defaultBind ();
Olası hatalar belirtilmeden default bind (varsayılan bağ tanımlama) değerini döndürür.

ImqBoolean definitionType (MQLONG & tip);
Tanımlama tipinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG definitionType ();
Olası hataları göstermeden tanım tipini döndürür.

ImqBoolean depthHighEvent (MQLONG & olay);
Derinlik yüksek olayının etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG depthHighEvent ();
Olası hataları göstermeden derinlik yüksek olayının etkinleştirme durumunu döndürür.

ImqBoolean depthHighLimit (MQLONG & sınır);
Derinlik üst sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG depthHighLimit ();
Olası hataları göstermeden derinlik üst sınırı değerini döndürür.

ImqBoolean depthLowEvent (MQLONG & olay);
Derinlik düşük olayının etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG depthLowEvent ();
Olası hataları göstermeden derinlik düşük olayının etkinleştirme durumunu döndürür.

ImqBoolean depthLowLimit (MQLONG & sınır);
Derinlik alt sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG depthLowLimit ();
Olası hataları göstermeden derinlik alt sınırı değerini döndürür.

ImqBoolean depthMaximumEvent (MQLONG & olay);
Derinlik üst sınırı olayının etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG depthMaximumEvent ();
Olası hataları göstermeden derinlik üst sınırı olayının etkinleştirme durumunu döndürür.

ImqDistributionList * distributionListReference () const ;
Dağıtım listesi başvurusunu döndürür.

void setDistributionListReference (ImqDistributionList & list);
Dağıtım listesi başvurusunu belirler.

void setDistributionListReference (ImqDistributionList * list = 0);
Dağıtım listesi başvurusunu belirler.

ImqBoolean distributionLists (MQLONG & desteği);

Dağıtım listeleri değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG distributionLists ();

Olası hataları göstermeden **dağıtım listeleri** değerini döndürür.

ImqBoolean setDistributionListeler (const MQLONG destek);

Dağıtım listeleri değerini belirler. Başarılı olursa TRUE değerini döndürür.

ImqString dynamicQueueAd () const ;

Dinamik kuyruk adının bir kopyasını döndürür.

ImqBoolean setDynamicQueueName (const char * name);

Dinamik kuyruk adını belirler. **Dinamik kuyruk adı** yalnızca ImqObject **açık durumu** FALSE olduğunda ayarlanabilir. Başarılı olursa TRUE değerini döndürür.

ImqBoolean get (ImqMessage & msg, ImqGetMessageOptions & options);

Belirtilen **seçenekler** kullanarak kuyruktan bir ileti alır. ImqObject **açma seçeneklerinin** , **seçeneklere** bağlı olarak MQOO_INPUT_ * değerlerinden birini ya da MQOO_BROWSE değerini içerdiğinden emin olmak için gerekirse ImqObject **openFor** yöntemini çağırır. **msg** nesnesinde bir ImqCache **otomatik arabellek** varsa, arabellek alınan tüm iletileri kapsayacak şekilde büyür. **clearMessage** yöntemi, alınmadan önce **msg** nesnesine karşı çağrılır.

Bu yöntem başarılı olursa TRUE değerini döndürür.

Not: Bu **neden kodu** uyarı olarak sınıflandırılmış olsa da, ImqObject **neden kodu**

MQRC_TRUNCATED_MSG_FAILED ise, yöntem çağrısının sonucu FALSE olur. Kesilmiş bir ileti kabul edilirse, ImqCache **ileti uzunluğu** kesilen uzunluğu yansıtır. Her iki durumda da, ImqMessage **toplam ileti uzunluğu** kullanılabilir bayt sayısını gösterir.

ImqBoolean get (ImqMessage & msg);

Önceki yöntem gelince, varsayılan ileti alma seçenekleri dışında.

ImqBoolean get (ImqMessage & msg, ImqGetMessageOptions & options, const size_t buffer-size);

Önceki iki yöntem gelince, **geçersiz kılan bir arabellek büyüklüğü** belirtilmesi dışında. **msg** nesnesi bir ImqCache **otomatik arabellek** kullanıyorsa, ileti alınmadan önce **msg** nesnesinde **resizeBuffer** yöntemi çağrılır ve arabellek daha büyük bir iletiyi barındıracak şekilde büyümür.

ImqBoolean get (ImqMessage & msg, const size_t buffer-size);

Önceki yöntem gelince, varsayılan ileti alma seçenekleri dışında.

ImqBoolean hardenGetBackout (MQLONG & harden);

Harden get backout değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG hardenGetBackout ();

Olası hataları göstermeden **harden get backout** değerini döndürür.

ImqBoolean indexType(MQLONG & tip);

Dizin tipinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG indexType();

Olası hataları göstermeden **dizin tipini** döndürür.

ImqBoolean inhibitGet (MQLONG & inhibe);

elde edilmeyi engelle değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG inhibitGet ();

Olası hataları göstermeden **inhibe get** değerini döndürür.

ImqBoolean setInhibitAI (const MQLONG engelleme);

get ögesini engelle değerini ayarlar. Başarılı olursa TRUE değerini döndürür.

ImqBoolean inhibitPut (MQLONG & inhibe);

inhibe koymayı engelle değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG inhibitPut ();

Olası hataları göstermeden **girişi engelle** değerini döndürür.

ImqBoolean setInhibitPut (const MQLONG inhibe);

girmeyi engelle değerini ayarlar. Başarılı olursa TRUE değerini döndürür.

ImqBoolean initiationQueueAd (ImqString & ad);

Başlatma kuyruğu adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString initiationQueueAd ();

Olası hataları göstermeden **başlatma kuyruğu adını** döndürür.

ImqBoolean maximumDepth (MQLONG & derinlik);

Maksimum derinliğin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumDepth ();

Olası hataları göstermeden **derinlik üst sınırını** döndürür.

ImqBoolean maximumMessageLength (MQLONG & uzunluk);

İleti uzunluğu üst sınırı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumMessageLength ();

Olası hataları göstermeden **ileti uzunluğu üst sınırını** döndürür.

ImqBoolean messageDeliverySequence (MQLONG & sequence);

İleti teslim sırasının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG messageDeliverySequence ();

Olası hataları göstermeden **ileti teslim sırası** değerini döndürür.

ImqQueue * nextDistributedQueue () const ;

Sonraki dağıtılmış kuyruğu döndürür.

ImqBoolean nonPersistentMessageClass (MQLONG & monq);

Kalıcı olmayan ileti sınıfı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG nonPersistentMessageClass ();

Olası hataları göstermeden kalıcı olmayan ileti sınıfı değerini döndürür.

ImqBoolean openInputCount (MQLONG & count);

Açık giriş sayısının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG openInputCount ();

Olası hataları göstermeden **açık giriş sayısını** döndürür.

ImqBoolean openOutputCount (MQLONG & count);

Açık çıkış sayısının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG openOutputCount ();

Olası hataları göstermeden **açık çıkış sayısını** döndürür.

ImqQueue * previousDistributedQueue () const ;

Önceki dağıtılmış kuyruğu döndürür.

ImqBoolean processName (ImqString & name);

Süreç adı' in bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString processName ();

Olası hataları göstermeden **süreç adını** döndürür.

ImqBoolean koyma (ImqMessage & msg);

Varsayılan koyma iletisi seçeneklerini kullanarak kuyruğa bir ileti yerleştirir. ImqObject **open seçeneklerinin** MQOO_OUTPUT içerdiğinden emin olmak için gerekirse ImqObject **openFor** yöntemini kullanır.

Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean Put (ImqMessage & msg, ImqPutMessageOptions & pmo);

Belirtilen *pmo*' yu kullanarak kuyruğa bir ileti yerleştirir. ImqObject **openFor** yöntemini ImqObject **open seçeneklerinin** MQOO_OUTPUT değerini içerdiğinden emin olmak için gerektiği şekilde kullanır ve (*pmo* **seçenekleri** MQPMO_PASS_IDENTITY_CONTEXT, MQPMO_PASS_ALL_CONTEXT, MQPMO_SET_IDENTITY_CONTEXT ya da MQPMO_SET_ALL_CONTEXT) ilgili MQOO_*_CONTEXT değerlerini içeriyorsa.

Bu yöntem başarılı olursa TRUE değerini döndürür.

Not: *pmo* bir **bağlam başvurusu** içeriyorsa, gerekiyorsa, bağlam sağlamak için başvuru nesne açılır.

ImqBoolean queueAccounting (MQLONG & acctq);

Kuyruk hesap deęerinin bir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

MQLONG queueAccounting ();

Olası hataları göstermeden kuyruk muhasebesi deęerini döndürür.

ImqString queueManagerAd () const ;

Kuyruk yöneticisi adınıdöndürür.

ImqBoolean setQueueManagerName (const char * name);

Kuyruk yöneticisi adınıbelirler. **Kuyruk yöneticisi adı** yalnızca ImqObject **açık durumu** FALSE olduğunda ayarlanabilir. Bu yöntem başarılı olursa TRUE deęerini döndürür.

ImqBoolean queueMonitoring (MQLONG & monq);

Kuyruk izleme deęerinin bir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

MQLONG queueMonitoring ();

Olası hataları göstermeden kuyruk izleme deęerini döndürür.

ImqBoolean queueStatistics (MQLONG & statq);

Kuyruk istatistikleri deęerinin bir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

MQLONG queueStatistics ();

Olası hataları göstermeden kuyruk istatistikleri deęerini döndürür.

ImqBoolean queueType (MQLONG & tip);

Kuyruk tipi deęerinin bir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

MQLONG queueType ();

Olası hataları göstermeden **kuyruk tipini** döndürür.

ImqBoolean remoteQueueManagerName (ImqString & name);

Uzak kuyruk yöneticisi adınınbir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

ImqString remoteQueueManagerName ();

Olası hataları göstermeden **uzak kuyruk yöneticisi adını** döndürür.

ImqBoolean remoteQueueAdı (ImqString & ad);

Uzak kuyruk adınınbir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

ImqString remoteQueueAdı ();

Olası hataları göstermeden **uzak kuyruk adını** döndürür.

ImqBoolean resolvedQueueManagerName(ImqString & ad);

Çözömlenen kuyruk yöneticisi adınınbir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

Not: MQOO_RESOLVE_NAMES, ImqObject **açma seçenekleri**arasında deęilse bu yöntem başarısız olur.

ImqString resolvedQueueManagerName();

Olası hataları göstermeden **çözölen kuyruk yöneticisi adını**döndürür.

ImqBoolean resolvedQueueName (ImqString & name);

Çözömlenen kuyruk adınınbir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

Not: MQOO_RESOLVE_NAMES, ImqObject **açma seçenekleri**arasında deęilse bu yöntem başarısız olur.

ImqString resolvedQueueName ();

Olası hataları göstermeden **çözölen kuyruk adını**döndürür.

ImqBoolean retentionInterval (MQLONG & aralık);

Saklama aralığınınbir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

MQLONG retentionInterval ();

Olası hataları göstermeden **alıkoyma aralığını** döndürür.

ImqBoolean kapsam (MQLONG & kapsam);

Kapsamınbir kopyasını saęlar. Bařarılı olursa TRUE deęerini döndürür.

MQLONG kapsam ();

Olası hataları göstermeden **kapsam** deęerini döndürür.

ImqBoolean serviceInterval (MQLONG & aralık);

Hizmet aralığı' in bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG serviceInterval ();

Olası hataları belirtmeden **hizmet aralığını** döndürür.

ImqBoolean serviceIntervalEvent (MQLONG & olay);

Hizmet aralığı olayının etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG serviceIntervalEvent ();

Olası hataları göstermeden **hizmet aralığı olayının** etkinleştirme durumunu döndürür.

ImqBoolean shareability (MQLONG & shareability);

shareability (paylaşılabilirlik) değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG paylaşılabilir ();

Olası hataları göstermeden **shareability** (paylaşılabilir) değerini döndürür.

ImqBoolean storageClass(ImqString & class);

Depolama sınıfının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString storageClass ();

Olası hataları göstermeden **depolama sınıfını** döndürür.

ImqBoolean transmissionQueueAd (ImqString & ad);

İletim kuyruğu adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString transmissionQueueAd ();

Olası hataları göstermeden **iletim kuyruğu adını** döndürür.

ImqBoolean triggerControl (MQLONG & denetim);

Tetikleme denetimi değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG triggerControl ();

Olası hataları göstermeden **tetikleyici denetimi** değerini döndürür.

ImqBoolean setTriggerControl (const MQLONG control);

Tetikleme denetimi değerini belirler. Başarılı olursa TRUE değerini döndürür.

ImqBoolean triggerData (ImqString & data);

Tetikleyici verilerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString triggerData ();

Olası hataları göstermeden **tetikleyici verilerinin** bir kopyasını döndürür.

ImqBoolean setTriggerData (const char * data);

Tetikleyici verilerini ayarlar. Başarılı olursa TRUE değerini döndürür.

ImqBoolean triggerDepth (MQLONG & derinlik);

Tetikleyici derinliğinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG triggerDepth ();

Olası hataları göstermeden **tetikleyici derinliğini** döndürür.

ImqBoolean setTriggerDerinlik (const MQLONG derinlik);

Tetikleme derinliğini ayarlar. Başarılı olursa TRUE değerini döndürür.

ImqBoolean triggerMessageÖncelik (MQLONG & öncelik);

Tetikleme iletisi önceliğinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG triggerMessageÖncelik ();

Olası hataları göstermeden **tetikleyici iletisi önceliğini** döndürür.

ImqBoolean setTriggerMessagePriority (const MQLONG priority);

Tetikleyici ileti önceliğini belirler. Başarılı olursa TRUE değerini döndürür.

ImqBoolean triggerType (MQLONG & tip);

Tetikleyici tipinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG triggerType ();

Olası hataları göstermeden **tetikleyici tipini** döndürür.

ImqBoolean setTriggerType (const MQLONG type);

Tetikleyici tipini belirler. Başarılı olursa TRUE değerini döndürür.

ImqBoolean usage (MQLONG & usage);

Kullanım değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG kullanımı ();

Olası hataları göstermeden **usage** değerini döndürür.

Nesne yöntemleri (korunmalı)

void setNextDistributedQueue (ImqQueue * kuyruk = 0);

Sonraki dağıtılmış kuyruğu belirler.

Dikkat: Bu işlevi yalnızca, dağıtılmış kuyruk listesinin bozulmayacağından eminseniz kullanın.

void setPreviousDistributedQueue (ImqQueue * kuyruk = 0);

Önceki dağıtılmış kuyruğu belirler.

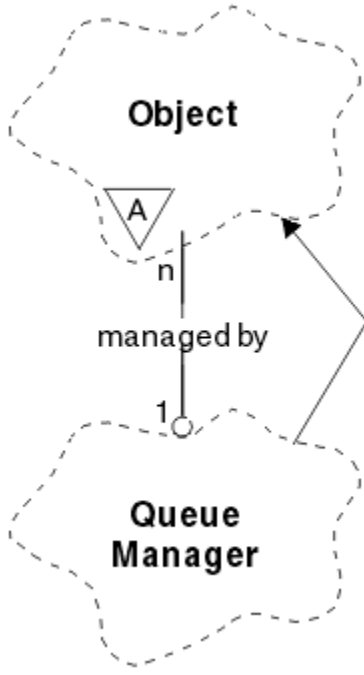
Dikkat: Bu işlevi yalnızca, dağıtılmış kuyruk listesinin bozulmayacağından eminseniz kullanın.

Neden kodları

- MQRC_ATTRIBUTE_LOCKED
- MQRC_CONTEXT_OBJECT_NOT_VALID
- MQRC_CONTEXT_OPEN_HATA
- MQRC_CURSOR_NOT_VALID
- MQRC_NO_BUFFER
- MQRC_REOPEN_EXCL_INPUT_HATA
- MQRC_REOPEN_INQUIRE_HATA
- MQRC_REOPEN_TEMPORARY_Q_ERROR
- (MQGET ' den neden kodları)
- (MQPUT ' den neden kodları)

ImqQueueManager C++ sınıfı

Bu sınıf bir kuyruk yöneticisini (MQOT_Q_MGR tipinde bir IBM MQ nesnesi) kapsıyor.



Şekil 33. *ImqQueueYönetici sınıfı*

Bu sınıf, “[ImqQueueManager çapraz başvurusu](#)” sayfa 1767’inde listelenen MQI çağrılarını ile ilgilidir. Listelenen yöntemlerin tümü tüm altyapılar için geçerli değildir; ek bilgi için [ALTER QMGR](#) başlıklı konuya bakın.

- “Sınıf öznitelikleri” sayfa 1836
- “Nesne öznitelikleri” sayfa 1837
- “Oluşturucular” sayfa 1842
- “Yok ediciler” sayfa 1842
- “Sınıf yöntemleri (genel)” sayfa 1842
- “Nesne yöntemleri (genel)” sayfa 1842
- “Nesne yöntemleri (korunmalı)” sayfa 1851
- “Nesne verileri (korunan)” sayfa 1851
- “Neden kodları” sayfa 1851

Sınıf öznitelikleri

davranış

Örtülü bağlantının ve bağlantının kesilmesinin davranışını denetler.

IMQ_EXPL_DISC_BACKOUT (0L)

Bağlantı kesme yöntemine yönelik belirtik bir çağrı, geriletme anlamına gelir. Bu öznitelik, IMQ_EXPL_DISC_COMMIT ile karşılıklı olarak birbirini dışlar.

IMQ_EXPL_DISC_COMMIT (1L)

Bağlantı kesme yöntemine belirtik bir çağrı kesinleştirme (varsayılan) anlamına gelir. Bu öznitelik, IMQ_EXPL_DISC_BACKOUT ile karşılıklı olarak birbirini dışlar.

IMQ_IMPL_CONN (2L)

Örtük bağlantıya izin verilir (varsayılan).

IMQ_IMPL_DISC_BACKOUT (0L)

Nesne yok edilirken oluşabilecek bağlantı kesme yöntemine yönelik örtük bir çağrı, gerilemeyi belirtir. Bu öznitelik, IMQ_IMPL_DISC_COMMIT ile karşılıklı olarak birbirini dışlar.

IMQ_IMPL_DISC_COMMIT (4L)

Nesne yok etme sırasında oluşabilecek bağlantı kesme yöntemine yönelik örtük bir çağrı kesinleştirme (varsayılan) anlamına gelir. Bu öznitelik, IMQ_IMPL_DISC_BACKOUT ile karşılıklı olarak birbirini dışlar.

IBM MQ V7.0 ve üstünde, örtük bir bağlantıdan yararlanan C++ uygulamalarının, ImqQueueManagersınıfındaki bir nesne için setBehavior() yönteminde sağlanan diğer seçeneklerle birlikte IMQ_IMPL_CONN belirtmesi gerekir. Uygulamanız davranış seçeneklerini belirttik olarak ayarlamak için setBehavior() yöntemini kullanmıyorsa, örneğin,

```
ImqQueueManager_object.setBehavior(IMQ_IMPL_DISC_COMMIT)
```

MQ_IMPL_CONN varsayılan olarak etkinleştirildiğinden bu değişiklik sizi etkilemez.

Uygulamanız davranış seçeneklerini belirttik olarak ayarlarsa, örneğin,

```
ImqQueueManager_object.setBehavior(IMQ_IMPL_DISC_COMMIT)
```

Uygulamanızın örtük bir bağlantıyı tamamlamasını sağlamak için, setBehavior() yöntemine IMQ_IMPL_CONN ögesini aşağıdaki gibi eklemeniz gerekir:

```
ImqQueueManager_object.setBehavior(IMQ_IMPL_CONN | IMQ_IMPL_DISC_COMMIT)
```

Nesne öznitelikleri

muhasabe bağlantıları geçersiz kılma

Uygulamaların MQI muhasebesi ve kuyruk muhasebesi values.This özniteliği salt okunurdu.

hesaplama aralığı

Ara muhasebe kayıtlarının yazılması için geçen süre (saniye). Bu öznitelik salt okunur.

etkinlik kaydı

Etkinlik raporlarının oluşturulmasını denetler. Bu öznitelik salt okunur.

yeni mca denetimini kabul et

Zaten etkin olan bir MCA ile aynı ada sahip yeni bir gelen kanal algılandığında MCA 'nın benimsenip benimsenmeyeceğini belirlemek için denetlenen öğeler. Bu öznitelik salt okunur.

yeni mca tipini benimse

Yeni mca denetimi parametreleriyle eşleşen yeni bir gelen kanal isteği saptandığında, belirli bir kanal tipindeki bir MCA 'nın artık kalmış bir örneğinin otomatik olarak yeniden başlatılıp başlatılmayacağı. Bu öznitelik salt okunur.

Kimlik Doğrulama Tipi

Gerçekleştirilmekte olan kimlik doğrulaması tipini gösterir.

yetki olayı

Yetki olaylarını denetler. Bu öznitelik salt okunur.

başlangıç seçenekleri

Başlangıç yöntemi için geçerli olan seçenekler. İlk değer MQBO_NONE.

köprü olayı

IMS köprü olaylarının oluşturulup oluşturulmadığını belirler. Bu öznitelik salt okunur.

kanal otomatik tanımı

Kanal otomatik tanımlama değeri. Bu öznitelik salt okunur.

kanal otomatik tanımlama olayı

Kanal otomatik tanımlama olay değeri. Bu öznitelik salt okunur.

kanal otomatik tanımlama çıkışı

Kanal otomatik tanımlama çıkış adı. Bu öznitelik salt okunur.

kanal olayı

Kanal olaylarının oluşturulup oluşturulmayacağını belirler. Bu öznitelik salt okunur.

kanal başlatıcı bağdaştırıcıları

IBM MQ çağrılarını işlemek için kullanılacak bağdaştırıcı alt görevlerinin sayısı. Bu öznitelik salt okunur.

kanal başlatıcı denetimi

Kuyruk Yöneticisi başlatıldığında Kanal Başlatıcısının otomatik olarak başlatılıp başlatılmayacağını belirler. Bu öznitelik salt okunur.

kanal başlatıcı dağıtıcıları

Kanal başlatıcısı için kullanılacak dağıtıcıların sayısı. Bu öznitelik salt okunur.

kanal başlatıcı izleme otomatik başlatma

Kanal başlatıcı izlemesinin otomatik olarak başlatılıp başlatılmayacağı. Bu öznitelik salt okunur.

kanal başlatıcı izleme çizelgesi büyüklüğü

Kanal başlatıcısının izleme veri alanının boyutu (MB). Bu öznitelik salt okunur.

kanal izleme

Kanallar için çevrimiçi izleme verilerinin toplanmasını denetler. Bu öznitelik salt okunur.

kanal başvurusu

İstemci bağlantısı sırasında kullanılacak bir kanal tanımına ilişkin başvuru. Bağlıyken bu öznitelik boş değere ayarlanabilir, ancak başka bir değere değiştirilemez. İlk değer boş.

kanal istatistikleri

Kanallar için istatistik verilerinin toplanmasını denetler. Bu öznitelik salt okunur.

karakter kümesi

Kodlanmış karakter takımı tanıtcısı (CCSID). Bu öznitelik salt okunur.

küme gönderen izlemesi

Otomatik olarak tanımlanan küme gönderen kanalları için çevrimiçi izleme verilerinin toplanmasını denetler. Bu öznitelik salt okunur.

küme gönderen istatistikleri

Otomatik olarak tanımlanan küme gönderen kanalları için istatistik verilerinin toplanmasını denetler. Bu öznitelik salt okunur.

küme iş yükü verileri

Küme iş yükü çıkış verileri. Bu öznitelik salt okunur.

küme iş yükü çıkışı

Küme iş yükü çıkış adı. Bu öznitelik salt okunur.

küme iş yükü uzunluğu

Küme iş yükü uzunluğu. Bu öznitelik salt okunur.

küme iş yükü mru

Küme iş yükü en son kullanılan kanal değeri. Bu öznitelik salt okunur.

küme iş yükü kullanım kuyruğu

Küme iş yükü kullanım kuyruğu değeri. Bu öznitelik salt okunur.

komut olayı

Komut olaylarının oluşturulup oluşturulmayacağını belirler. Bu öznitelik salt okunur.

komut giriş kuyruğu adı

Sistem komutu giriş kuyruğu adı. Bu öznitelik salt okunur.

komut düzeyi

Kuyruk yöneticisi tarafından desteklenen komut düzeyi. Bu öznitelik salt okunur.

komut sunucusu denetimi

Kuyruk Yöneticisi başlatıldığında Komut Sunucusunun otomatik olarak başlatılıp başlatılmayacağını belirler. Bu öznitelik salt okunur.

bağlantı seçenekleri

Bağlanma yöntemine uygulanan seçenekler. İlk değer MQCNO_NONE. Aşağıdaki ek değerler, platforma bağlı olarak mümkün olabilir:

- 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

Bağlantı Tanıtıcısı

MQ ' nun bir uygulamayı güvenilir bir şekilde tanımlamasına olanak sağlayan benzersiz tanıtıcı.

Bağlantı durumu

Kuyruk yöneticisine bağlanıldığında TRUE (Doğru) değerini verir. Bu öznitelik salt okunur.

bağlantı etiketi

Bir bağlantıyla ilişkilendirilecek etiket. Bu öznitelik yalnızca bağlı olmadığında ayarlanabilir. İlk değer boş.

şifreleme donanımı

Şifreleme donanımına ilişkin yapılandırma ayrıntıları. MQ MQI istemci bağlantıları için.

ileti kuyruğu adı

Gönderilmeyen iletiler kuyruğunun adı. Bu öznitelik salt okunur.

varsayılan iletim kuyruğu adı

Varsayılan iletim kuyruğu adı. Bu öznitelik salt okunur.

dağıtım listeleri

Kuyruk yöneticisinin dağıtım listelerini destekleme yeteneği.

dns grubu

İş Yüklü Yöneticisi Dinamik Etki Alanı Ad Hizmetleri desteği kullanılırken, kuyruk paylaşım grubuna ilişkin gelen iletileri işleyen TCP dinleyicisinin katılması gereken grubun adı. Bu öznitelik salt okunur.

dns wlm

Kuyruk paylaşım grubuna ilişkin gelen iletileri işleyen TCP dinleyicisinin, Dinamik Etki Alanı Ad Hizmetleri için İş Yüklü Yöneticisi 'ne kaydolmasının gerekip gerekmediğini belirler. Bu öznitelik salt okunur.

ilk kimlik doğrulama kaydı

ImqAuthenticationsınıfındaki bir ya da daha çok nesnenin ilki, ImqAuthenticationKayıt bağlantısı başvurusunun bu nesneyi ele aldığı belirli bir sırada değildir. MQ MQI istemci bağlantıları için.

ilk yönetilen nesne

ImqObject bağlantı başvurusunun bu nesneyi ele aldığı ImqObjectsınıfındaki bir ya da daha çok nesnenin ilki. İlk değer sıfırdır.

etkinliği engelle

Engelleme olaylarını denetler. Bu öznitelik salt okunur.

ip adresi sürümü

Kanal bağlantısı için kullanılacak IP iletişim kuralı (IPv4 ya da IPv6). Bu öznitelik salt okunur.

anahtar havuzu

Anahtarların ve sertifikaların saklandığı anahtar veritabanı dosyasının konumu. IBM MQ MQI client bağlantıları için.

anahtar sıfırlama sayısı

Gizli anahtar yeniden anlaşılmadan önce bir TLS etkileşimi içinde gönderilen ve alınan şifrelenmemiş bayt sayısı. Bu öznitelik yalnızca MQCONNX kullanan istemci bağlantıları için geçerlidir. Ayrıca bkz. [ssl tuşu sıfırlama sayısı](#).

dinleyici süreölçeri

Bir APPC ya da TCP/IP hatası oluştuysa, IBM MQ ' in dinleyiciyi yeniden başlatma girişimleri arasındaki zaman aralığı (saniye). Bu öznitelik salt okunur.

yerel olay

Yerel olayları denetler. Bu öznitelik salt okunur.

günlüğe kaydedici olayı

Kurtarma günlüğü olaylarının oluşturulup oluşturulmayacağını denetler. Bu öznitelik salt okunur.

lu grup adı

Kuyruk paylaşım grubuna ilişkin gelen iletimleri işleyen LU 6.2 dinleyicisinin kullanması gereken soysal LU adı. Bu öznitelik salt okunur.

lu adı

Giden LU 6.2 iletimleri için kullanılacak LU ' nun adı. Bu öznitelik salt okunur.

lu62 kol soneki

SYS1.PARMLIB üyesi APPCPMxx. Bu öznitelik salt okunur.

lu62 kanalları

LU 6.2 iletim protokolünü kullanan, yürürlükteki ya da bağlanabilen istemci sayısı üst sınırı. Bu öznitelik salt okunur.

etkin kanal sayısı üst sınırı

Herhangi bir zamanda etkin olabilecek kanal sayısı üst sınırı. Bu öznitelik salt okunur.

kanal üst sınırı

Geçerli olabilecek kanal sayısı üst sınırı (bağlı istemcilerle sunucu bağlantısı kanalları da içinde olmak üzere). Bu öznitelik salt okunur.

çekme noktası üst sınırı

Tarıtıcı sayısı üst sınırı. Bu öznitelik salt okunur.

ileti uzunluğu üst sınırı

Bu kuyruk yöneticisi tarafından yönetilen herhangi bir kuyruktaki herhangi bir ileti için olası uzunluk üst sınırı. Bu öznitelik salt okunur.

öncelik üst sınırı

İleti önceliği üst sınırı. Bu öznitelik salt okunur.

kesinleştirilmemiş ileti sayısı üst sınırı

Bir birim ya da iş içindeki kesinleştirilmemiş ileti sayısı üst sınırı. Bu öznitelik salt okunur.

mqi muhasebesi

MQI verileri için muhasebe bilgileri toplamasını denetler. Bu öznitelik salt okunur.

mqi istatistikleri

Kuyruk yöneticisine ilişkin istatistik izleme bilgilerinin toplanmasını denetler. Bu öznitelik salt okunur.

giden kapı üst sınırı

Giden kanallar için bağ tanımlanırken kullanılacak kapı numaraları aralığının üst ucu. Bu öznitelik salt okunur.

giden kapı alt sınırı

Giden kanallar bağlanırken kullanılacak kapı numaraları aralığının alt ucu. Bu öznitelik salt okunur.

parola

kullanıcı kimliğiyle ilişkili parola

performans olayı

Performans olaylarını denetler. Bu öznitelik salt okunur.

platformu

Kuyruk yöneticisinin bulunduğu platform. Bu öznitelik salt okunur.

kuyruk muhasebesi

Kuyruklara ilişkin hesap bilgilerinin toplanmasını denetler. Bu öznitelik salt okunur.

kuyruk izleme

Kuyruklara ilişkin çevrimiçi izleme verilerinin toplanmasını denetler. Bu öznitelik salt okunur.

kuyruk istatistikleri

Kuyruklara ilişkin istatistik verilerinin toplanmasını denetler. Bu öznitelik salt okunur.

alma zaman aşımı

Bir TCP/IP ileti kanalının, iş ortağından etkin değil durumuna dönmeden önce, sağlıklı işletim bildirimleri de içinde olmak üzere, verileri almak için bekleyeceği süre. Bu öznitelik salt okunur.

alma zaman aşımı alt sınırı

Etkin değil durumuna dönmeden önce, TCP/IP kanalının ortağından sağlıklı işletim bildirimleri de içinde olmak üzere veri almak için bekleyeceği süre alt sınırı. Bu öznitelik salt okunur.

alma zaman aşımı tipi

Alma zaman aşımı için bir niteleyici uygulandı. Bu öznitelik salt okunur.

uzak olay

Uzak olayları denetler. Bu öznitelik salt okunur.

Havuz adı

Havuz adı. Bu öznitelik salt okunur.

havuz ad listesi

Havuz ad listesi adı. Bu öznitelik salt okunur.

paylaşılan kuyruk yöneticisi adı

ObjectQMgrAdının kuyruk paylaşım grubundaki başka bir kuyruk yöneticisi olduğu paylaşılan bir kuyruğun MQOPEN'lerinin, yerel kuyruk yöneticisinde paylaşılan kuyruğun açılmasına çözülmesi gerekip gerekmediğini belirler. Bu öznitelik salt okunur.

ssl olayı

SSL olaylarının oluşturulup oluşturulmayacağını belirler. Bu öznitelik salt okunur.

ssl FIPS gerekli

Şifreleme IBM MQ yazılımında yürütüldüğünde yalnızca FIPS sertifikalı algoritmaların kullanılıp kullanılmayacağı. Bu öznitelik salt okunur.

ssl anahtar sıfırlama sayısı

Gizli anahtar yeniden anlaşılmadan önce SSL etkileşimi içinde gönderilen ve alınan şifrelenmemiş bayt sayısı. Bu öznitelik salt okunur.

start-stop olayı


Başlatma-durdurma olaylarını denetler. Bu öznitelik salt okunur.

istatistik aralığı

İstatistik izleme verilerinin izleme kuyruğuna ne sıklıkta yazıldığı. Bu öznitelik salt okunur.

eşitleme noktası kullanılabilirliği

Syncpoint katılımının kullanılabilirliği. Bu öznitelik salt okunur.

Not: Kuyruk yöneticisi-eşgüdümlü genel iş birimleri IBM i altyapısında desteklenmez. 

Bir iş birimini, _Rcommit ve _Rback yerel sistem çağrılarını kullanarak IBM itarafından dışarıdan eşgüdümlü olarak programlayabilir. IBM MQ uygulamasını iş düzeyi kesinleştirme denetimi altında STRCMTCTL komutunu kullanarak başlatarak bu iş birimi tipini başlatın. Daha fazla ayrıntı için bkz. [IBM i dış syncpoint yöneticisine yönelik arabirimler](#) . IBM i altyapısında, bir kuyruk yöneticisi tarafından eşgüdümlü yerel iş birimleri için geriletme ve kesinleştirme desteklenir.

tcp kanalları

TCP/IP iletim protokolünü kullanan, yürürlükteki ya da bağlanabilen istemci sayısı üst sınırı. Bu öznitelik salt okunur.

tcp canlı tutma

Bağlantının diğer ucunun hala kullanılabilir olup olmadığını denetlemek için TCP KEEPALIVE olanağının kullanılıp kullanılmayacağını belirler. Bu öznitelik salt okunur.

tcp adı

tcp yığın tipinin değerine bağlı olarak, kullanılacak tek ya da varsayılan TCP/IP sisteminin adı. Bu öznitelik salt okunur.

tcp yığın tipi

Kanal başlatıcısının yalnızca tcp adında belirlenen TCP/IP adres alanını kullanmasına izin verilip verilmediğini ya da seçilen herhangi bir TCP/IP adresine bağlanabilip bağlanamayacağını belirler. Bu öznitelik salt okunur.

rota kaydını izle

Rota izleme bilgilerinin kaydedilmesini denetler. Bu öznitelik salt okunur.

tetikleme aralığı

Tetikleme aralığı. Bu öznitelik salt okunur.

kullanıcı kimliği

AIX and Linux platformlarında, uygulamanın gerçek kullanıcı kimliği. Windows platformlarında, uygulamanın kullanıcı kimliği.

Oluşturucular

ImqQueueManager ();

Varsayılan oluşturucu.

ImqQueueManager (const ImqQueueManager & manager);

Kopya oluşturucu. Bağlantı durumu FALSE olacak.

ImqQueueManager (const char * ad);

ImqObject adını *ad* olarak ayarlar.

Yok ediciler

Bir ImqQueueManager nesnesi yok edildiğinde, otomatik olarak bağlantısı kesilir.

Sınıf yöntemleri (genel)

statik MQLONG davranışı ();

Davranışı döndürür.

void setBehavior(const MQLONG davranış = 0);

Davranışı ayarlar.

Nesne yöntemleri (genel)

void işleci = (const ImqQueueManager & mgr);

Gerekirse bağlantıyı keser ve eşgörünüm verilerini *mgr*' dan kopyalar. Bağlantı durumu FALSE.

ImqBoolean accountingConnOverride (MQLONG & statint);

Muhasebe bağlantıları geçersiz kılma değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG accountingConnOverride ();

Olası hataları göstermeden muhasebe bağlantıları geçersiz kılma değerini döndürür.

ImqBoolean accountingInterval (MQLONG & statint);

Hesap aralığı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG accountingInterval ();

Olası hataları göstermeden hesap aralığı değerini döndürür.

ImqBoolean activityRecording (MQLONG & rec);

Etkinlik kaydı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG activityRecording ();

Olası hataları göstermeden etkinlik kaydı değerini döndürür.

ImqBoolean adoptNewMCACheck (MQLONG & check);

Yeni MCA denetim değerinin benimsenmesi için bir kopya sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG adoptNewMCACheck ();

Olası hataları göstermeden yeni MCA denetim değerini benimse değerini döndürür.

ImqBoolean adoptNewMCAType (MQLONG & type);

Yeni MCA tipinin benimsenmesi için bir kopya sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG adoptNewMCAType ();

Olası hataları göstermeden yeni MCA tipini benimse tipini döndürür.

QLONG authenticationType () const;

Kimlik doğrulama tipini döndürür.

void setAuthenticationType (const MQLONG type = MQCSP_AUTH_NONE);

Kimlik doğrulama tipini ayarlar.

ImqBoolean authorityEvent(MQLONG & olayı);

Yetki olayının etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG authorityEvent();

Olası hataları göstermeden yetki olayının etkinleştirme durumunu döndürür.

ImqBoolean backout ();

Kesinleştirilmemiş değişiklikleri geri çeviriyor. Başarılı olursa TRUE değerini döndürür.

ImqBoolean begin ();

Bir iş birimi başlatır. Başlangıç seçenekleri bu yöntemin davranışını etkiler. Başarılı olursa TRUE değerini döndürür, ancak temeldeki MQBEGIN çağrısı MQRC_NO_EXTERNAL_KATILIMCILAR ya da MQRC_PARTICIPANT_NOT_VAR (ikisi de MQCC_WARNING ile ilişkilidir) değerini döndürse de TRUE değerini döndürür.

MQLONG beginOptions() const;

Başlangıç seçeneklerini döndürür.

void setBeginOptions (const MQLONG options = MQBO_NONE);

Başlangıç seçeneklerini belirler.

ImqBoolean bridgeEvent (MQLONG & event);

Köprü olayı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG bridgeEvent ();

Olası hataları göstermeden köprü olayı değerini döndürür.

ImqBoolean channelAutoDefinition (MQLONG & değeri);

Kanal otomatik tanımlama değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelAutoTanımlaması ();

Olası hataları göstermeden kanal otomatik tanımlama değerini döndürür.

ImqBoolean channelAutoDefinitionEvent(MQLONG & değeri);

Kanal otomatik tanımlama olay değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelAutoDefinitionEvent();

Olası hataları göstermeden kanal otomatik tanımlama olay değerini döndürür.

ImqBoolean channelAutoDefinitionExit(ImqString & ad);

Kanal otomatik tanımlama çıkış adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString channelAutoDefinitionExit();

Olası hataları göstermeden kanal otomatik tanımlama çıkış adını döndürür.

ImqBoolean channelEvent (MQLONG & event);

Kanal olay değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelEvent();

Olası hataları göstermeden kanal olayı değerini döndürür.

MQLONG channelInitiatorAdapters ();

Olası hataları göstermeden kanal başlatıcı bağdaştırıcıları değerini döndürür.

ImqBoolean channelInitiatorAdapters (MQLONG & adapters);

Kanal başlatıcı bağdaştırıcıları değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelInitiatorControl ();

Olası hataları göstermeden kanal başlatıcı başlatma değerini döndürür.

ImqBoolean channelInitiatorControl (MQLONG & init);

Kanal başlatıcı denetimi başlatma değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelInitiatorDağıtıcılar ();

Olası hataları göstermeden kanal başlatıcı dağıtıcıları değerini döndürür.

ImqBoolean channelInitiatorDağıtıcılar (MQLONG & dispatchers);

Kanal başlatıcı dağıtıcıları değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelInitiatorTraceAutoStart ();

Olası hataları göstermeden kanal başlatıcı izleme otomatik başlatma değerini döndürür.

ImqBoolean channelInitiatorTraceAutoStart (MQLONG & auto);

Kanal başlatıcı izleme otomatik başlatma değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelInitiatorTraceTableSize ();

Olası hataları göstermeden kanal başlatıcı izleme çizelgesi büyüklüğü değerini döndürür.

ImqBoolean channelInitiatorTraceTableSize (MQLONG & size);

Kanal başlatıcı izleme çizelgesi büyüklüğü değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqBoolean channelMonitoring (MQLONG & monchl);

Kanal izleme değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelMonitoring ();

Olası hataları göstermeden kanal izleme değerini döndürür.

ImqBoolean channelReference(ImqChannel * & pchannel);

Kanal başvurusunun bir kopyasını sağlar. Kanal başvurusu geçersizse, *pchannel* 'ı boş değere ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqChannel * channelReference();

Olası hataları göstermeden kanal başvurusunu döndürür.

ImqBoolean setChannelReference (ImqChannel & channel);

Kanal başvurusunu ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setChannelReference (ImqChannel * kanal = 0);

Kanal başvurusunu ayarlar ya da ilk durumuna getirir. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean channelStatistics (MQLONG & statchl);

Kanal istatistikleri değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG channelStatistics ();

Olası hataları göstermeden kanal istatistikleri değerini döndürür.

ImqBoolean characterSet(MQLONG & ccsid);

Karakter kümesinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG characterSet();

Olası hataları göstermeden karakter kümesinin bir kopyasını döndürür.

MQLONG clientSslKeyResetCount () const;

İstemci bağlantılarında kullanılan SSL anahtar sıfırlama sayısı değerini döndürür.

void setClientSslKeyResetCount(const MQLONG sayısı);

İstemci bağlantılarında kullanılan SSL anahtar sıfırlama sayısını belirler.

ImqBoolean clusterSenderMonitoring (MQLONG & monacl);

Küme gönderen izleme varsayılan değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG clusterSenderMonitoring ();

Olası hataları göstermeden küme gönderen izleme varsayılan değerini döndürür.

ImqBoolean clusterSenderStatistics (MQLONG & stacals);

Küme gönderen istatistikleri değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG clusterSenderStatistics ();

Olası hataları göstermeden küme gönderen istatistikleri değerini döndürür.

ImqBoolean clusterWorkloadData (ImqString & data);

Küme iş yükü çıkış verilerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString clusterWorkloadData ();

Olası hataları göstermeden küme iş yükü çıkış verilerini döndürür.

ImqBoolean clusterWorkloadExit (ImqString & ad);

Küme iş yükü çıkış adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString clusterWorkloadExit ();

Olası hataları göstermeden küme iş yükü çıkış adını döndürür.

ImqBoolean clusterWorkloadLength (MQLONG & uzunluk);

Küme iş yükü uzunluğunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG clusterWorkloadLength ();

Olası hataları göstermeden küme iş yükü uzunluğunu döndürür.

ImqBoolean clusterWorkLoadMRU (MQLONG & mru);

Küme iş yükünün en son kullanılan kanal değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG clusterWorkLoadMRU ();

Olası hataları göstermeden, en son kullanılan kanalların küme iş yükü değerini döndürür.

ImqBoolean clusterWorkLoadUseQ (MQLONG & useq);

Küme iş yükü kullanım kuyruğu değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG clusterWorkLoadUseQ ();

Olası hataları göstermeden küme iş yükü kullanım kuyruğu değerini döndürür.

ImqBoolean commandEvent (MQLONG & event);

Komut olayı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG commandEvent ();

Olası hataları göstermeden komut olayı değerini döndürür.

ImqBoolean commandInputQueueName(ImqString & ad);

Komut giriş kuyruğu adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString commandInputQueueName();

Olası hataları göstermeden komut giriş kuyruğu adını döndürür.

ImqBoolean commandLevel(MQLONG & düzey);

Komut düzeyinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG commandLevel();

Olası hataları göstermeden komut düzeyini döndürür.

MQLONG commandServerControl ();

Olası hataları göstermeden komut sunucusu başlatma değerini döndürür.

ImqBoolean commandServerControl (MQLONG & server);

Komut sunucusu denetim başlatma değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqBoolean commit ();

Kesinleştirilmemiş değişiklikleri kesinleştirir. Başarılı olursa TRUE değerini döndürür.

ImqBoolean connect ();

Belirtilen ImqObject adıyla kuyruk yöneticisine bağlanır; varsayılan değer, yerel kuyruk yöneticisidir. Belirli bir kuyruk yöneticisine bağlanmak istiyorsanız, bağlantıdan önce ImqObject setName yöntemini kullanın. Bir kanal başvurusu varsa, kanal tanımlamasıyla ilgili bilgileri MQCD ' deki MQCONNX ' e geçirmek için kullanılır. MQCD ' deki ChannelType MQCHT_CLNTCONN olarak ayarlandı. Yalnızca istemci bağlantıları için anlamlı olan kanal başvurusu bilgileri, sunucu bağlantıları için dikkate alınmaz.

Bağlanma seçenekleri bu yöntemin davranışını etkiler. Bu yöntem, başarılı olursa bağlantı durumunu TRUE olarak ayarlar. Yeni bağlantı durumunu döndürür.

İlk kimlik doğrulama kaydı varsa, güvenli istemci kanallarına ilişkin dijital sertifikaların kimliğini doğrulamak için kimlik doğrulama kayıtları zinciri kullanılır.

Aynı kuyruk yöneticisine birden çok ImqQueue nesnesi bağlayabilirsiniz. Tümü, iş parçacığıyla ilişkilendirilmiş bağlantı için aynı MQHCONN bağlantı tanıtıcısını kullanır ve UOW işlevselliğini paylaşır. Bağlanacak ilk ImqQueue yöneticisi MQHCONN tanıtıcısını alır. Bağlantıyı kesmek için son ImqQueueManager, MQDISC 'yi gerçekleştirir.

Çok iş parçacıklı bir program için, her iş parçacığı için ayrı bir ImqQueueManager nesnesi kullanılması önerilir.

ImqBinary connectionId () const;

Bağlantı tanıtıcısını döndürür.

ImqBinary connectionTag () const;

Bağlantı etiketini döndürür.

ImqBoolean setConnectionTag (const MQBYTE128 etiketi = 0);

Bağlantı etiketini ayarlar. *etiket* sıfırsa, bağlantı etiketini temizler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setConnectionTag (const ImqBinary & etiketi);

Bağlantı etiketini ayarlar. *tag* veri uzunluğu sıfır olmalıdır (bağlantı etiketini temizlemek için) ya da MQ_CONN_TAG_LENGTH. Bu yöntem başarılı olursa TRUE değerini döndürür.

MQLONG connectOptions() const;

Bağlantı seçeneklerini döndürür.

void setConnectOptions (const MQLONG options = MQCNO_NONE);

Bağlantı seçeneklerini belirler.

ImqBoolean connectionStatus() const;

Bağlantı durumunu döndürür.

ImqString cryptographicHardware ();

Şifreleme donanımını döndürür.

ImqBoolean setCryptographicDonanım (const char * hardware = 0);

Şifreleme donanımını ayarlar. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean deadLetterQueueName(ImqString & ad);

İleti gönderilmeyen ileti kuyruğu adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString deadLetterQueueName();

Olası hataları göstermeden, gönderilmeyen ileti kuyruğu adının bir kopyasını döndürür.

ImqBoolean defaultTransmissionQueueName(ImqString & ad);

Varsayılan iletim kuyruğu adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString defaultTransmissionQueueName();

Olası hataları göstermeden varsayılan iletim kuyruğu adını döndürür.

ImqBoolean disconnect ();

Kuyruk yöneticisiyle bağlantıyı keser ve bağlantı durumunu FALSE olarak ayarlar. Bu nesneyle ilişkili tüm ImqProcess ve ImqQueue nesnelerini kapatır ve bağlantı kesilmeden önce bağlantı başvurularını ayarlar. Birden çok ImqQueueManager nesnesi aynı kuyruk yöneticisine bağlıysa, yalnızca son bağlantı kesilen nesne fiziksel bağlantıyı keser; diğerleri mantıksal bağlantıyı keser. Kesinleştirilmemiş değişiklikler yalnızca fiziksel bağlantı kesilirse kesinleştirilir.

Bu yöntem başarılı olursa TRUE değerini döndürür. Bağlantı olmadığına çağrılırsa, dönüş kodu da true olur.

ImqBoolean distributionLists(MQLONG & desteği);

Dağıtım listesi değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG distributionLists();

Olası hataları göstermeden dağıtım listesi değerini döndürür.

ImqBoolean dnsGroup (ImqString & group);

DNS grup adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString dnsGroup ();

Olası hataları göstermeden DNS grubu adını döndürür.

ImqBoolean dnsWlm (MQLONG & wlm);

DNS WLM değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG dnsWlm ();

Olası hataları göstermeden DNS WLM değerini döndürür.

ImqAuthenticationRecord * firstAuthenticationRecord () const;

İlk kimlik doğrulama kaydını döndürür.

void setFirstAuthenticationRecord (const ImqAuthenticationRecord * air = 0);

İlk kimlik doğrulama kaydını ayarlar.

ImqObject * firstManagedObject () const;

İlk yönetilen nesneyi döndürür.

ImqBoolean inhibitEvent(MQLONG & olay);

Engelleme olayının etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG inhibitEvent();

Olası hataları göstermeden, engelleme olayının etkinleştirme durumunu döndürür.

ImqBoolean ipAddressSürüm (MQLONG & version);

IP adresi sürüm değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG ipAddressSürüm ();

Olası hataları göstermeden IP adresi sürüm değerini döndürür.

ImqBoolean keepAlive (MQLONG & keepalive);

Canlı tutma değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG keepAlive ();

Olası hataları göstermeden canlı tutma değerini döndürür.

ImqString keyRepository ();

Anahtar havuzunu döndürür.

ImqBoolean setKeyRepository (const char * repository = 0);

Anahtar havuzunu ayarlar. Başarılı olursa TRUE değerini döndürür.

ImqBoolean listenerTimer (MQLONG & timer);

Dinleyici süreölçeri değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG listenerTimer ();

Olası hataları göstermeden dinleyici süreölçeri değerini döndürür.

ImqBoolean localEvent(MQLONG & olay);

Yerel olayın etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG localEvent();

Olası hataları göstermeden yerel olayın etkinleştirme durumunu döndürür.

ImqBoolean loggerEvent (MQLONG & count);

Günlüğe kaydedici olay değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG loggerEvent ();

Olası hataları göstermeden günlüğe kaydedici olay değerini döndürür.

ImqBoolean luGroupAdı (ImqString & name);

LU grup adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür

ImqString luGroupName ();

Olası hataları göstermeden LU grup adını döndürür.

ImqBoolean lu62ARMSuffix (ImqString & suffix);

LU62 ARM son ekinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString lu62ARMSuffix ();

Olası hataları göstermeden LU62 ARM son ekini döndürür.

ImqBoolean luName (ImqString & name);

LU adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString luName ();

Olası hataları göstermeden LU adını döndürür.

ImqBoolean maximumActiveKanalları (MQLONG & kanal);

Etkin kanal sayısı üst sınırı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumActiveKanalları ();

Olası hataları göstermeden etkin kanal sayısı üst sınırını döndürür.

ImqBoolean maximumCurrentKanalları (MQLONG & kanal);

Yürürlükteki kanal sayısı üst sınırı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumCurrentKanalları ();

Olası hataları göstermeden geçerli kanal değeri üst sınırını döndürür.

ImqBoolean maximumHandles(MQLONG & sayı);

Tanıtıcı üst sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumHandles();

Olası hataları göstermeden tanıtıcı sayısı üst sınırını döndürür.

ImqBoolean maximumLu62Channels (MQLONG & kanal);

LU62 kanal değeri üst sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumLu62Channels ();

Olası hataları göstermeden LU62 kanal değeri üst sınırını döndürür.

ImqBoolean maximumMessageLength (MQLONG & length);

İleti uzunluğu üst sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumMessageLength ();

Olası hataları göstermeden ileti uzunluğu üst sınırını döndürür.

ImqBoolean maximumPriority(MQLONG & öncelik);

Öncelik üst sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumPriority();

Olası hataları göstermeden, öncelik üst sınırının bir kopyasını döndürür.

ImqBoolean maximumTcpKanalları (MQLONG & kanal);

TCP kanalı üst sınırı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumTcpKanalları ();

Olası hataları göstermeden TCP kanalı sayısı üst sınırını döndürür.

ImqBoolean maximumUncommittedİletiler (MQLONG & sayı);

Kesinleştirilmemiş ileti sayısı üst sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG maximumUncommittedMessages ();

Olası hataları göstermeden kesinleştirilmemiş ileti sayısı üst sınırını döndürür.

ImqBoolean mqiAccounting (MQLONG & statint);

MQI muhasebe değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG mqiAccounting ();

Olası hataları göstermeden MQI muhasebe değerini döndürür.

ImqBoolean mqiStatistics (MQLONG & statmqi);

MQI istatistik değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG mqiStatistics ();

Olası hataları göstermeden MQI istatistikleri değerini döndürür.

ImqBoolean outboundPortMaks (MQLONG & max);

Giden kapı değeri üst sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG outboundPortMaks ();

Olası hataları göstermeden giden kapı değeri üst sınırını döndürür.

ImqBoolean outboundPortMin (MQLONG & min);

Minimum giden kapı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG outboundPortMin ();

Olası hataları göstermeden giden kapı değeri alt sınırını döndürür.

ImqBinary password () const;

İstemci bağlantılarında kullanılan parolayı döndürür.

ImqBoolean setPassword (const ImqString & password);

İstemci bağlantılarında kullanılan parolayı belirler.

ImqBoolean setPassword (const char * = 0 parola);

İstemci bağlantılarında kullanılan parolayı belirler.

ImqBoolean setPassword (const ImqBinary & password);

İstemci bağlantılarında kullanılan parolayı belirler.

ImqBoolean performanceEvent(MQLONG & olay);

Performans olayının etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG performanceEvent();

Olası hataları göstermeden performans olayının etkinleştirme durumunu döndürür.

ImqBoolean altyapısı (MQLONG & platform);

Platformun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG platform ();

Olası hataları göstermeden platformu döndürür.

ImqBoolean queueAccounting (MQLONG & acctq);

Kuyruk hesap değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG queueAccounting ();

Olası hataları göstermeden kuyruk muhasebesi değerini döndürür.

ImqBoolean queueMonitoring (MQLONG & monq);

Kuyruk izleme değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG queueMonitoring ();

Olası hataları göstermeden kuyruk izleme değerini döndürür.

ImqBoolean queueStatistics (MQLONG & statq);

Kuyruk istatistikleri değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG queueStatistics ();

Olası hataları göstermeden kuyruk istatistikleri değerini döndürür.

ImqBoolean receiveTimeout (MQLONG & timeout);

Alma zamaşımı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG receiveTimeout ();

Olası hataları göstermeden alma zamaşımı değerini döndürür.

ImqBoolean receiveTimeoutMin (MQLONG & min);

Alma zamaşımı değeri alt sınırının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG receiveTimeoutMin ();

Olası hataları göstermeden alma zamaşımı değeri alt sınırını döndürür.

ImqBoolean receiveTimeoutTipi (MQLONG & type);

Alma zamaşımı tipinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG receiveTimeoutTip ();

Olası hataları göstermeden alma zamaşımı tipini döndürür.

ImqBoolean remoteEvent(MQLONG & olay);

Uzak olayın etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG remoteEvent();

Olası hataları göstermeden uzak olayın etkinleştirme durumunu döndürür.

ImqBoolean repositoryName(ImqString & name);

Havuz adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString repositoryName();

Olası hataları göstermeden havuz adını döndürür.

ImqBoolean repositoryNameListAd (ImqString & ad);

Havuz ad listesi adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString repositoryNameListName ();

Olası hataları göstermeden havuz ad listesi adının bir kopyasını döndürür.

ImqBoolean sharedQueueQueueManagerAdı (MQLONG & name);

Paylaşılan kuyruk yöneticisi adı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG sharedQueueQueueManagerName ();

Olası hataları göstermeden, paylaşılan kuyruk yöneticisi adı değerini döndürür.

ImqBoolean sslEvent (MQLONG & event);

SSL olay değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG sslEvent ();

Olası hataları göstermeden SSL olay değerini döndürür.

ImqBoolean sslFips (MQLONG & sslfips);

SSL FIPS değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG sslFips ();

Olası hataları göstermeden SSL FIPS değerini döndürür.

ImqBoolean sslKeyResetCount (MQLONG & count);

SSL anahtar sıfırlama sayısı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG sslKeyResetCount ();

Olası hataları göstermeden SSL anahtar sıfırlama sayısı değerini döndürür.

ImqBoolean startStopEvent (MQLONG & olayı);

Start-stop olayının etkinleştirme durumunun bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG startStopEvent ();

Olası hataları göstermeden, start-stop olayının etkinleştirme durumunu döndürür.

ImqBoolean statisticsInterval (MQLONG & statint);

İstatistik aralığı değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG statisticsInterval ();

Olası hataları göstermeden istatistik aralığı değerini döndürür.

ImqBoolean syncPointKullanılabilirliği (MQLONG & eşitleme);

Eşitleme noktası kullanılabilirlik değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG syncPointKullanılabilirliği ();

Olası hataları göstermeden, syncpoint kullanılabilirlik değerinin bir kopyasını döndürür.

ImqBoolean tcpName (ImqString & name);

TCP sistem adının bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

ImqString tcpName ();

Olası hataları göstermeden TCP sistem adını döndürür.

ImqBoolean tcpStackTipi (MQLONG & type);

TCP yığın tipinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG tcpStackTipi ();

Olası hataları göstermeden TCP yığın tipini döndürür.

ImqBoolean traceRouteKaydı (MQLONG & routerec);

İzleme rotası kayıt değerinin bir kopyasını sağlar. Başarılı olursa TRUE değerini döndürür.

MQLONG traceRouteRecording ();

Olası hataları göstermeden izleme rotası kayıt deęerini döndürür.

ImqBoolean triggerInterval(MQLONG & interval);

Tetikleme aralığının bir kopyasını sağlar. Başarılı olursa TRUE deęerini döndürür.

MQLONG triggerInterval();

Olası hataları göstermeden tetikleme aralığını döndürür.

ImqBinary userId () const;

İstemci bağlantılarında kullanılan kullanıcı kimliğini döndürür.

ImqBoolean setUserId (const ImqString & id);

İstemci bağlantılarında kullanılan kullanıcı kimliğini belirler.

ImqBoolean setUserId (const char * = 0 id);

İstemci bağlantılarında kullanılan kullanıcı kimliğini belirler.

ImqBoolean setUserId (const ImqBinary & id);

İstemci bağlantılarında kullanılan kullanıcı kimliğini belirler.

Nesne yöntemleri (korunmalı)**void setFirstManagedObject (const ImqObject * object = 0);**

İlk yönetilen nesneyi ayarlar.

Nesne verileri (korunan)**MQHCONN ohconn**

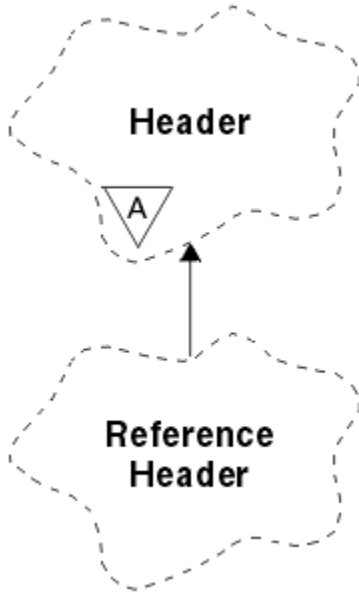
IBM MQ bağlantı tanıtıcısı (yalnızca bağlantı durumu TRUE olduğunda anlamlıdır).

Neden kodları

- MQRC_ATTRIBUTE_LOCKED
- MQRC_ENVIRONMENT_ERROR
- MQRC_FUNCTION_NOT_SUPPORTED
- MQRC_REFERENCE_HATA
- (MQBACK için neden kodları)
- (MQBEGIN için neden kodları)
- (MQCMIT için neden kodları)
- (MQCONNX için neden kodları)
- (MQDISC için neden kodları)
- (MQCONN için neden kodları)

ImqReferenceHeader C++ sınıfı

Bu sınıf, MQRMH veri yapısının özelliklerini içerir.



Şekil 34. *ImqReferenceÜstbilgi sınıfı*

Bu sınıf, [“ImqReferenceÜstbilgi çapraz başvurusu” sayfa 1771](#) içinde listelenen MQI çağrılarını ile ilgilidir.

- [“Nesne öznitelikleri” sayfa 1852](#)
- [“Oluşturucular” sayfa 1853](#)
- [“ImqItem yöntemleri aşırı yüklendi” sayfa 1853](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1853](#)
- [“Nesne verileri \(korunan\)” sayfa 1854](#)
- [“Neden kodları” sayfa 1854](#)

Nesne öznitelikleri

hedef ortam

Hedef için ortam. Başlangıç değeri boş değerli bir dizgi.

Hedef Adı

Veri hedefinin adı. Başlangıç değeri boş değerli bir dizgi.

Eşgörünüm Tanıtıcısı

Eşgörünüm tanıtıcısı. MQ_OBJECT_INSTANCE_ID_LENGTH uzunluklu bir ikili değer (MQBYTE24). İlk değer MQOII_NONE.

mantıksal uzunluk

Bu üstbilgiyi izleyen ileti verilerinin mantıksal ya da amaçlanan uzunluğu. İlk değer sıfırdır.

mantıksal görelî konum

İzleyen ileti verilerinin mantıksal görelî konumu, en son hedefte, veriler bağlamında bir bütün olarak yorumlanacaktır. İlk değer sıfırdır.

mantıksal görelî konum 2

Mantıksal görelî konuma yüksek sıralı uzantı. İlk değer sıfırdır.

Başvuru tipi

Başvuru tipi. Başlangıç değeri boş değerli bir dizgi.

Kaynak Ortamı

Kaynak için ortam. Başlangıç değeri boş değerli bir dizgi.

KAYNAK ADI

Veri kaynağının adı. Başlangıç değeri boş değerli bir dizgi.

Olusturucular

ImqReferenceUstbilgisi ();

Varsayılan olusturucu.

ImqReferenceUstbilgisi (const ImqReferenceHeader & header);

Kopya olusturucu.

ImqItem yontemleri asiri yuklendi

sanal ImqBoolean copyOut (ImqMessage & msg);

Baslangicta ileti arabelleğine bir MQRMH veri yapısı ekler, var olan ileti verilerini daha ileriye taşıyarak *msg* biçimini MQFMT_REF_MSG_HEADER olarak ayarlar.

Daha fazla ayrıntı için "[ImqHeader C++ sınıfı](#)" sayfa 1799 üzerindeki ImqHeader sınıf yöntemi açıklamasına bakın.

sanal ImqBoolean pasteIn (ImqMessage & msg);

İleti arabelleğindeki bir MQRMH veri yapısını okur.

Başarılı olmak için ImqMessage biçimi MQFMT_REF_MSG_HEADER olmalıdır.

Daha fazla ayrıntı için "[ImqHeader C++ sınıfı](#)" sayfa 1799 üzerindeki ImqHeader sınıf yöntemi açıklamasına bakın.

Nesne yontemleri (genel)

void işleci = (const ImqReferenceHeader & header);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *üstbilgiseçeneğinden* kopyalar.

ImqString destinationEnvironment () const;

Hedef ortamın bir kopyasını döndürür.

void setDestinationEnvironment (const char * environment = 0);

Hedef ortamı belirler.

ImqString destinationName () const;

Hedef adının bir kopyasını döndürür.

void setDestinationAd (const char * ad = 0);

Hedef adını belirler.

ImqBinary instanceId () const;

Eşgörünüm tanıtıcısının bir kopyasını döndürür.

ImqBoolean setInstanceId (const ImqBinary & id);

Eşgörünüm tanıtıcısını ayarlar. *simge* veri uzunluğu 0 ya da MQ_OBJECT_INSTANCE_ID_LENGTH olmalıdır. Bu yöntem başarılı olursa TRUE değerini döndürür.

void setInstanceId (const MQBYTE24 id = 0);

Eşgörünüm tanıtıcısını ayarlar. *tanıtıcı* sıfır olabilir; bu, MQOII_NONE ile aynıdır. *id* sıfır değilse, ikili verilerin MQ_OBJECT_INSTANCE_ID_LENGTH byte 'larına yanıt vermelidir. MQOII_NONE gibi önceden tanımlanmış değerleri kullanırken, imza eşleşmesini sağlamak için bir tip dönüşümü yapmanız gerekebilir; örneğin, (MQBYTE *) MQOII_NONE.

MQLONG logicalLength () const;

Mantıksal uzunluğu döndürür.

void setLogicalLength (const MQLONG length);

Mantıksal uzunluğu belirler.

MQLONG logicalOffset () const;

Mantıksal görel konumu döndürür.

void setLogicalOffset (const MQLONG offset);

Mantıksal görel konumu belirler.

MQLONG logicalOffset2 () const;

Mantıksal görel konum 2 'yi döndürür.

void setLogicalOffset2 (const MQLONG görelî konumu);

Mantıksal görelî konum 2 'yi ayarlar.

ImqString referenceType () const;

Başvuru tipinin bir kopyasını döndürür.

void setReferenceType (const char * name = 0);

Başvuru tipini ayarlar.

ImqString sourceEnvironment () const;

Kaynak ortamın bir kopyasını döndürür.

void setSourceEnvironment (const char * environment = 0);

Kaynak ortamı ayarlar.

ImqString sourceName () const;

Kaynak adının bir kopyasını döndürür.

void setSourceAd (const char * ad = 0);

Kaynak adını belirler.

Nesne verileri (korunan)

MQRMH omqrmh

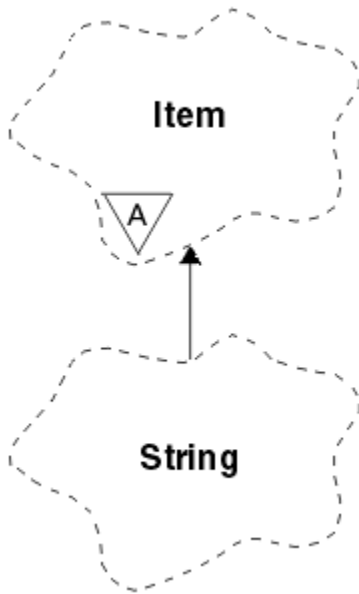
MQRMH veri yapısı.

Neden kodları

- MQR_C_BINARY_DATA_LENGTH_HATA
- MQR_C_STRUC_LENGTH_HATA
- MQR_C_STRUC_HATA
- MQR_C_VERIMSIZ_VERI
- MQR_C_TUTARSIZ_BIÇIMI
- MQR_C_ENCODING_HATA

ImqString C++ sınıfı

Bu sınıf, boş sonlandırılmış dizgiler için karakter dizgisi saklama ve işleme olanağı sağlar.



Şekil 35. ImqString sınıfı

Bir parametrenin **char *** için çağrıda bulunduğu çoğu durumda **char *** yerine `ImqString` kullanın.

- [“Nesne öznitelikleri” sayfa 1855](#)
- [“Oluşturucular” sayfa 1855](#)
- [“Sınıf yöntemleri \(genel\)” sayfa 1855](#)
- [“ImqItem yöntemleri aşırı yüklendi” sayfa 1856](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1856](#)
- [“Nesne yöntemleri \(korumalı\)” sayfa 1859](#)
- [“Neden kodları” sayfa 1859](#)

Nesne öznitelikleri

karakter

Saklama alanındaki karakterler, sondaki boş değerden önce gelir.

uzunluk

Karakteriçindeki bayt sayısı. **Saklama alanı**yoksa, **uzunluk** sıfır olur. İlk değer sıfırdır.

depolama

Rasgele büyüklükte geçici bir bayt dizisi. **karakterlerin** sonunun saptanabilmesi için, **saklama alanında karakterlerden sonra** her zaman bir sonda boş değer bulunmalıdır. Yöntemler bu durumun korunmasını sağlar, ancak dizideki byte 'ları doğrudan ayarlarken, değişiklikten sonra sonda boş değer bulunduğundan emin olun. Başlangıçta, **storage** özneliği yoktur.

Oluşturucular

`ImqString();`

Varsayılan oluşturucu.

`ImqString(const ImqString & dizgi);`

Kopya oluşturucu.

`ImqString(const char c);`

karakterleri *c* 'yi oluşturur.

`ImqString(const char * text);`

karakterleri *metinden* kopyalanır.

`ImqString(const void * buffer, const size_t length);`

length byte 'ı *buffer* tipinden başlayarak kopyalar ve bunları **karakterlere**atar. Kopyalanan boş değerli karakterler için yerine koyma değeri yapılıdır. Yerine koyma karakteri bir nokta (.). Kopyalanan diğer yazdırılamayan ya da görüntülenemez karakterlere özel önem verilmez.

Sınıf yöntemleri (genel)

statik `ImqBoolean` kopyası (`char * hedef-arabellek, const size_t uzunluk, const char * kaynak arabellek, const char pad = 0`);

kaynak arabellekten en çok *uzunluk* byte 'ı *hedef arabellek* e kopyalar. *kaynak-arabellek* içindeki karakter sayısı yetersizse, *hedef-arabellek* alanında kalan boşluğu *doldurma* karakterleriyle doldurur. *kaynak-arabellek* sıfır olabilir. *hedef-arabellek* , *uzunluk* değeri de sıfır ise sıfır olabilir. Hata kodları kaybolur. Bu yöntem başarılı olursa `TRUE` değerini döndürür.

statik `ImqBoolean` kopyası (`char * hedef-arabellek, const size_t uzunluk, const char * kaynak arabellek, ImqError & error-object, const char pad = 0`);

kaynak arabellekten en çok *uzunluk* byte 'ı *hedef arabellek* e kopyalar. *kaynak-arabellek* içindeki karakter sayısı yetersizse, *hedef-arabellek* alanında kalan boşluğu *doldurma* karakterleriyle doldurur. *kaynak-arabellek* sıfır olabilir. *hedef-arabellek* , *uzunluk* değeri de sıfır ise sıfır olabilir. Hata kodları *hata-nesnesi* içinde ayarlanır. Bu yöntem başarılı olursa `TRUE` değerini döndürür.

ImqItem yöntemleri aşırı yüklendi

sanal ImqBoolean copyOut (ImqMessage & msg);

karakterleri ileti arabelleğine kopyalar ve var olan içeriği değiştirir. *msg biçimini* MQFMT_STRING olarak ayarlar.

Daha fazla ayrıntı için üst sınıf yöntemi tanımına bakın.

sanal ImqBoolean pasteIn (ImqMessage & msg);

İleti arabelleğindeki kalan verileri, var olan **karakterleri**değiştirerek aktararak **karakterleri** belirler.

Başarılı olmak için *msg* nesnesinin **kodlaması** MQENC_NATIVE olmalıdır. MQGMO_CONVERT ile MQENC_NATIVE iletilerini alın.

Başarılı olmak için ImqMessage **biçimi** MQFMT_STRING olmalıdır.

Daha fazla ayrıntı için üst sınıf yöntemi tanımına bakın.

Nesne yöntemleri (genel)

char & operator [] (konst size_t göreli konum) const;

storageiçindeki *offset* göreli konumundaki karaktere başvuruda bulunur. İlgili baytın var olduğundan ve adreslenebilir olduğundan emin olun.

ImqString işleci () (konst size_t offset, konst size_t length = 1) const;

offset konumundan başlayarak **karakterlerinden** bayt kopyalayarak bir alt dizgi döndürür. *length* sıfır ise, **karakterlerini** geri kalanını döndürür. *offset* ve *length* birleşimi **karakterleri** içinde bir başvuru üretmezse, boş bir ImqString döndürür.

void işleci = (konst ImqString & dizgi);

Var olan eşgörünüm verilerini değiştirerek *dizgi*' den eşgörünüm verilerini kopyalar.

ImqString işleci + (konst char c) const;

karakterlerin sonuna *c* eklenmesinin sonucunu döndürür.

ImqString işleç + (konst char * metin) const;

karakterlerin sonuna *metin* eklenmesinin sonucunu döndürür. Bu da tersine çevrilebiliyor. Örneğin:

```
strOne + "string two" ;  
"string one" + strTwo ;
```

Not: Çoğu derleyici **strOne + "string two"**; Microsoft Visual C++ için **strOne + (char *) "string two"**;

ImqString işleci + (konst ImqString & string1) const;

karakterlerin sonuna *string1* eklenmesinin sonucunu döndürür.

ImqString işleç + (konst double sayı) const;

Metne dönüştürmeden sonra **karakterlere** *sayı* eklenmesinin sonucunu döndürür.

ImqString işleci + (konst long sayı) const;

Metne dönüştürmeden sonra **karakterlere** *sayı* eklenmesinin sonucunu döndürür.

void işleci + = (konst char c);

karakterlerin sonuna *c* ekler.

void işleci + = (konst char * text);

karakterlerin sonuna *metin* ekler.

void işleci + = (konst ImqString & dizgi);

karakterlerin sonuna *dizgi* ekler.

void işleci + = (konst double sayı);

Metne dönüştürmeden sonra **karakterlerin** sonuna *sayı* ekler.

void işleci + = (konst long sayı);

Metne dönüştürmeden sonra **karakterlerin** sonuna *sayı* ekler.

işletmen char * () const;

Saklama alanındaki ilk baytın adresini döndürür. Bu değer sıfır olabilir ve geçici olabilir. Bu yöntemi yalnızca salt okunur amaçlarla kullanın.

ImqBoolean işleci < (const ImqString & string) const;

karakterleri , **karşılaştırma** yöntemini kullanarak *dizgi* ile karşılaştırır. Sonuç küçükse TRUE, büyükse ya da eşitse FALSE olur.

ImqBoolean işleci > (const ImqString & dizgi) const;

karakterleri , **karşılaştırma** yöntemini kullanarak *dizgi* ile karşılaştırır. Bundan büyükse sonuç TRUE, küçükse ya da eşitse FALSE olur.

ImqBoolean işleci < = (const ImqString & dizgi) const;

karakterleri , **karşılaştırma** yöntemini kullanarak *dizgi* ile karşılaştırır. Bundan küçük ya da büyükse sonuç TRUE (Doğru), büyükse FALSE (Yanlış) olur.

ImqBoolean işleci > = (const ImqString & dizgi) const;

karakterleri , **karşılaştırma** yöntemini kullanarak *dizgi* ile karşılaştırır. Bundan büyük ya da ona eşitse sonuç TRUE (Doğru), küçükse FALSE (Yanlış) olur.

ImqBoolean işleci == (const ImqString & dizgi) const;

karakterleri , **karşılaştırma** yöntemini kullanarak *dizgi* ile karşılaştırır. TRUE ya da FALSE değerini döndürür.

ImqBoolean işleci != (const ImqString & dizgi) const;

karakterleri , **karşılaştırma** yöntemini kullanarak *dizgi* ile karşılaştırır. TRUE ya da FALSE değerini döndürür.

kısa karşılaştırma (const ImqString & dizgi) const;

karakterleri *dizgi* ile karşılaştırır. **Karakterler** eşitse sonuç sıfır, küçükse eksi ve büyüktür ise pozitif olur. Karşılaştırma büyük ve küçük harfe duyarlıdır. Boş bir ImqString boş olmayan bir ImqString değerinden küçük olarak kabul edilir.

ImqBoolean copyOut(char * buffer, const size_t length, const char pad = 0);

karakterlerden en çok *uzunluk* baytı *arabelleğe* kopyalar. **Karakter** sayısı yetersizse, *arabellek* alanında kalan boşluğu *doldurma* karakterleriyle doldurur. *buffer* , *length* değeri de sıfır ise sıfır olabilir. Başarılı olursa TRUE değerini döndürür.

size_t copyOut(uzun & sayı) const;

Metinden dönüştürmeden sonra **karakterlerden** *sayı* değerini ayarlar ve dönüştürmede yer alan karakterlerin sayısını döndürür. Bu değer sıfırsa, dönüştürme gerçekleştirilmez ve *sayı* ayarlanmaz. Dönüştürülebilir bir karakter sırası aşağıdaki değerlerle başlamalıdır:

```
<blank(s)>  
<+|->  
digit(s)
```

size_t copyOut(ImqString & token, const char c = ") const;

Karakterler , *c* den farklı bir ya da daha fazla karakter içeriyorsa, bir simgeyi bu karakterlerin ilk bitişik sırası olarak tanımlar. Bu durumda *simge* bu sıraya ayarlanır ve döndürülen değer, baştaki karakterlerin sayısı *c* ve sıradaki bayt sayısı toplamıdır. Ters durumda sıfır döndürülür ve *simge* ayarlanmaz.

size_t cutOut(uzun & sayı);

copy yönteminde *number* değerini ayarlar, ancak dönüş değeriyle gösterilen bayt sayısını **karakterlerden** de kaldırır. Örneğin, aşağıdaki örnekte gösterilen *dizgi* **cutOut (sayı)** kullanılarak üç sayıya kesilebilir. Üç kez:

```
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 = " "

token 'ı **copyOut** yöntemi için ayarlar ve **karakterlerinden** *strToken* karakterlerini ve *token* karakterlerinden önce gelen *c* karakterlerini kaldırır. *c* boşluk değilse, *simge* karakterlerinin doğrudan yerine gelen *c* karakterlerini kaldırır. Kaldırılan karakterlerin sayısını döndürür. Örneğin, aşağıdaki örnekte gösterilen dizgi **cutOut** (*simge*) kullanılarak üç simgeye kesilebiliyor. üç kez:

```
strText = " Program Version 1.1 "  
while ( strText.cutOut( token ) );  
  
// token becomes "Program", then "Version",  
// then "1.1" leaving strText == " "
```

Aşağıdaki örnek, bir DOS yol adının nasıl ayrıştırılacağını göstermektedir:

```
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 find(const ImqString & string);

Karakter içinde herhangi bir yerde *dizgi* için tam eşleşme arar. Eşleşme bulunamazsa, FALSE değerini döndürür. Ters durumda TRUE değerini döndürür. *dizgi* boş değerliyse, TRUE değerini döndürür.

ImqBoolean find(const ImqString & string, size_t & offset);

dizgi için, **karakterlerin** göreli konumdan *görelî konumdan* itibaren bir yerde tam eşleşme arar. *dizgi* boşsa, *görelî konum* güncellenmeden TRUE değerini döndürür. Eşleşme bulunamazsa, FALSE değerini döndürür (*offset* değeri artırılmış olabilir). Bir eşleşme bulunursa TRUE değerini döndürür ve *kayma* değerini **karakterleri** içindeki *dizgi* göreli konumuna günceller.

size_t length () const;

uzunluk değerini döndürür.

ImqBoolean pasteIn(const double number, const char * format = "%f");

Metne dönüştürmeden sonra **karakterlerin** sonuna *sayı* ekler. Başarılı olursa TRUE değerini döndürür.

Kayan nokta dönüşümünü biçimlendirmek için *format* belirtimi kullanılır. Belirtilirse, **printf** ve kayan noktalı sayılarla (örneğin, **%3f**) kullanılmak üzere uygun bir *sayı* olmalıdır.

ImqBoolean pasteIn(const long number);

Metne dönüştürmeden sonra **karakterlerin** sonuna *sayı* ekler. Başarılı olursa TRUE değerini döndürür.

ImqBoolean pasteIn(const void * buffer, const size_t length);

buffer arabelleğindeki *uzunluk* byte 'ı **karakterlerin** sonuna ekler ve sondaki son boş değeri ekler. Kopyalanan boş karakterlerin yerini alır. Yerine koyma karakteri bir nokta (.). Kopyalanan diğer yazdırılmayan ya da görüntülenmeyen karakterlere özel önem verilmez. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean set(const char * buffer, const size_t length);

karakterleri , boş değer içerebilecek bir değişmez uzunluklu karakter alanından ayarlar. Gerekirse, değişmez uzunluklu alandaki karakterlere boş değer ekler. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqBoolean setStorage(const size_t length);

Saklama alanını ayırır (ya da yeniden ayırır). Sondaki boş değer de içinde olmak üzere, özgün **karakterleri** korur (bunlar için hala yer varsa), ancak ek saklama alanı başlatmaz.

Bu yöntem başarılı olursa TRUE değerini döndürür.

size_t storage () const;

storage(depolama) içindeki bayt sayısını döndürür.

size_t stripLeading(const char c = " ");

karakterlerin başındaki c karakterlerini temizler ve kaldırılan sayıyı döndürür.

size_t stripTrailing(const char c = " ");

karakterlerin sonundaki c karakterlerini temizler ve kaldırılan sayıyı döndürür.

ImqString upperCase() const;

karakterlerin büyük bir kopyasını döndürür.

Nesne yöntemleri (korumalı)

ImqBoolean assign (const ImqString & dizgi);

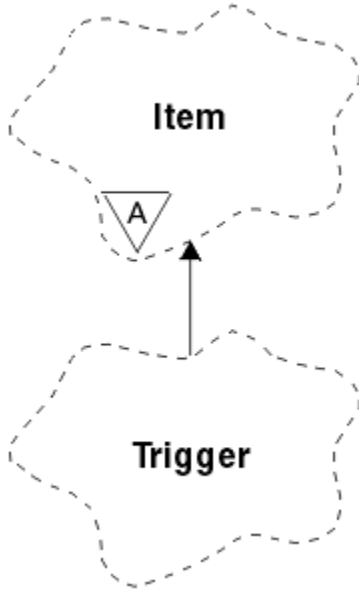
Eşdeğer **operator** = yöntemine eşdeğerdir, ancak sanal değildir. Başarılı olursa TRUE değerini döndürür.

Neden kodları

- MQRC_DATA_TRUNCATED
- MQRC_NULL_POINTER
- MQRC_STORAGE_KULLANILAMIYOR
- MQRC_BUFFER_ERROR
- MQRC_TUTARSIZ_BIÇIMI

ImqTrigger C++ sınıfı

Bu sınıf MQTM (tetikleyici iletisi) veri yapısını kapsıyor.



Şekil 36. ImqTrigger sınıfı

Bu sınıftaki nesnelere genellikle bir tetikleyici izleme programı tarafından kullanılır. Tetikleyici izleme programının görevi, bu belirli iletileri beklemek ve diğer IBM MQ uygulamalarının ileti beklerken başlatılmalarını sağlamak için bu iletiler üzerinde işlem yapmaktır.

Kullanım örneği için IMQSTRG örnek programına bakın.

- [“Nesne öznitelikleri” sayfa 1860](#)
- [“Oluşturucular” sayfa 1860](#)

- [“ImqItem yöntemleri aşırı yüklendi” sayfa 1860](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1861](#)
- [“Nesne verileri \(korunan\)” sayfa 1861](#)
- [“Neden kodları” sayfa 1862](#)

Nesne öznitelikleri

Uygulama tanıtıcısı

İletiyi gönderen uygulamanın tanıtıcısı. Başlangıç değeri boş değerli bir dizgi.

Uygulama Tipi

İletiyi gönderen uygulamanın tipi. İlk değer sıfırdır. Aşağıdaki ek değerler kullanılabilir:

- 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

ortam verileri

Sürece ilişkin ortam verileri. Başlangıç değeri boş değerli bir dizgi.

İşlem adı

Süreç adı. Başlangıç değeri boş değerli bir dizgi.

Kuyruk adı

Başlatılacak kuyruğun adı. Başlangıç değeri boş değerli bir dizgi.

tetikleyici verileri

Sürece ilişkin verileri tetikleyin. Başlangıç değeri boş değerli bir dizgi.

Kullanıcı verileri

Sürece ilişkin kullanıcı verileri. Başlangıç değeri boş değerli bir dizgi.

Oluşturucular

ImqTrigger();

Varsayılan oluşturucu.

ImqTrigger(const ImqTrigger & tetikleyici);

Kopya oluşturucu.

ImqItem yöntemleri aşırı yüklendi

sanal ImqBoolean copyOut (ImqMessage & msg);

İleti arabelleğine bir MQTM veri yapısı yazar ve var olan içeriği değiştirir. msg biçimini MQFMT_TRIGGER olarak ayarlar.

Daha fazla ayrıntı için [“ImqItem C++ sınıfı” sayfa 1803](#) adresindeki ImqItem sınıf yöntemi açıklamasına bakın.

sanal ImqBoolean pasteIn (ImqMessage & msg);

İleti arabelleğindeki bir MQTM veri yapısını okur.

Başarılı olmak için ImqMessage biçimi MQFMT_TRIGGER olmalıdır.

Daha fazla ayrıntı için [“ImqItem C++ sınıfı” sayfa 1803](#) adresindeki ImqItem sınıf yöntemi açıklamasına bakın.

Nesne yöntemleri (genel)

void işleci = (const ImqTrigger & tetikleyici);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *tetikleyiciden* kopyalar.

ImqString applicationId () const;

Uygulama tanıtıcısının bir kopyasını döndürür.

void setApplicationId (const char * id);

Uygulama tanıtıcısını ayarlar.

MQLONG applicationType () const;

Uygulama tipini döndürür.

void setApplicationType (const MQLONG tip);

Uygulama tipini ayarlar.

ImqBoolean copyOut (MQTMC2 * ptmc2);

Başlatma kuyruklarında alınan MQTM veri yapısını kapsüller. Çağırان tarafından sağlanan eşdeğer bir MQTMC2 veri yapısını doldurur ve QMgrName alanını (MQTM veri yapısında olmayan) tüm boşluklara ayarlar. MQTMC2 veri yapısı, geleneksel olarak bir tetikleyici izleyicisi tarafından başlatılan uygulamalar için bir parametre olarak kullanılır. Bu yöntem başarılı olursa TRUE değerini döndürür.

ImqString environmentData () const;

Ortam verilerinin bir kopyasını döndürür.

void setEnvironmentData (const char * data);

Ortam verilerini ayarlar.

ImqString processName () const;

Süreç adının bir kopyasını döndürür.

void setProcessAd (const char * ad);

İşlem adını, boşluklarla doldurarak 48 karaktere ayarlar.

ImqString queueName () const;

Kuyruk adının bir kopyasını döndürür.

void setQueueAd (const char * ad);

Kuyruk adını, boşluklarla doldurarak 48 karaktere ayarlar.

ImqString triggerData () const;

Tetikleyici verilerinin bir kopyasını döndürür.

void setTriggerData (const char * data);

Tetikleyici verilerini ayarlar.

ImqString userData () const;

Kullanıcı verilerinin bir kopyasını döndürür.

void setUserData (const char * data);

Kullanıcı verilerini ayarlar.

Nesne verileri (korunan)

MQTM omqtm

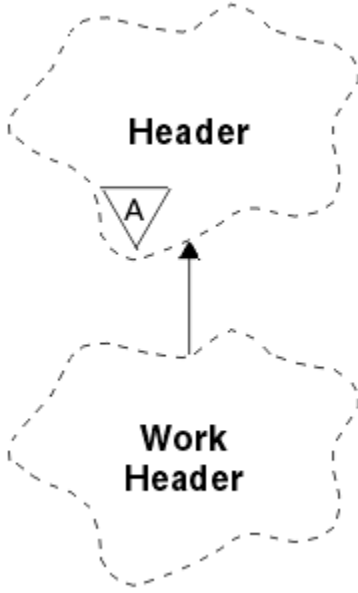
MQTM veri yapısı.

Neden kodları

- MQRC_NULL_POINTER
- MQRC_TUTARSIZ_BIÇIMI
- MQRC_ENCODING_HATA
- MQRC_STRUC_HATA

ImqWorkÜstbilgi C++ sınıfı

Bu sınıf, MQWIH veri yapısının belirli özelliklerini içerir.



Şekil 37. ImqWorkÜstbilgi sınıfı

Bu sınıftaki nesnelere, iletileri z/OS Workload Manager tarafından yönetilen kuyruğa yerleşen uygulamalar tarafından kullanılır.

- [“Nesne öznitelikleri” sayfa 1862](#)
- [“Oluşturucular” sayfa 1862](#)
- [“ImqItem yöntemleri aşırı yüklendi” sayfa 1863](#)
- [“Nesne yöntemleri \(genel\)” sayfa 1863](#)
- [“Nesne verileri \(korunan\)” sayfa 1863](#)
- [“Neden kodları” sayfa 1863](#)

Nesne öznitelikleri

ileti simgesi

MQ_MSG_TOKEN_LENGTH uzunluğundaki z/OS İş Yüğü Yöneticisi için ileti simgesi. İlk değer MQMTOK_NONE.

Hizmet Adı

Bir işlemin 32 karakterlik adı. Ad başlangıçta boşluktur.

hizmet adımı

İşlem içindeki bir adımın 8 karakterlik adı. Ad başlangıçta boşluktur.

Oluşturucular

ImqWorkÜstbilgisi ();

Varsayılan oluşturucu.

ImqWorkÜstbilgisi (const ImqWorkÜstbilgi & üstbilgi);

Kopya oluşturucu.

ImqItem yöntemleri aşırı yüklendi

sanal ImqBoolean copyOut(ImqMessage & msg);

İleti arabelleğinin başına bir MQWIH veri yapısı ekler, var olan ileti verilerini daha ileriye taşıyarak *msg* biçimini MQFMT_WORK_INFO_HEADER olarak ayarlar.

Daha fazla ayrıntı için üst sınıf yöntemi tanımına bakın.

sanal ImqBoolean pasteIn(ImqMessage & msg);

İleti arabelleğindeki bir MQWIH veri yapısını okur.

Başarılı olmak için *msg* nesnesinin kodlaması MQENC_NATIVE olmalıdır. MQGMO_CONVERT ile MQENC_NATIVE iletilerini alın.

ImqMessage biçimi MQFMT_WORK_INFO_HEADER olmalıdır.

Daha fazla ayrıntı için üst sınıf yöntemi tanımına bakın.

Nesne yöntemleri (genel)

void işleci = (const ImqWorkÜstbilgi & üstbilgi);

Var olan eşgörünüm verilerini değiştirerek eşgörünüm verilerini *üstbilgiseçeneğinden* kopyalar.

ImqBinary messageToken () const;

ileti simgesinidöndürür.

ImqBoolean setMessageToken (const ImqBinary & token);

ileti simgesiniayarlar. *simge* veri uzunluğu sıfır ya da MQ_MSG_TOKEN_LENGTH olmalıdır. Başarılı olursa TRUE değerini döndürür.

void setMessageToken (const MQBYTE16 token = 0);

ileti simgesiniayarlar. *simge* sıfır olabilir; bu, MQMTOK_NONE ile aynıdır. *simge* sıfır değilse, ikili verilerin MQ_MSG_TOKEN_LENGTH byte 'larına yanıt vermelidir.

MQMTOK_NONE gibi önceden tanımlanmış değerleri kullanırken, imza eşleşmesini sağlamak için bir tip dönüşümü yapmanız gerekebilir; örneğin, (MQBYTE *) MQMTOK_NONE.

ImqString serviceName () const;

Sondaki boşluklar da içinde olmak üzere **hizmet adını**döndürür.

void setServiceAd (const char * ad);

hizmet adınıayarlar.

ImqString serviceStep () const;

Sondaki boşluklar da içinde olmak üzere **hizmet adımını**döndürür.

void setServiceAdım (const char * adım);

Hizmet adımı' nı ayarlar.

Nesne verileri (korunan)

MQWIH omqwih

MQWIH veri yapısı.

Neden kodları

- MQRC_BINARY_DATA_LENGTH_HATA

IBM MQ classes for JMS nesnelere özellikleri

IBM MQ classes for JMS içindeki tüm nesnelere özellikleri vardır. Farklı özellikler farklı nesne tipleri için geçerlidir. Farklı özelliklerin izin verilen farklı değerleri vardır ve simgesel özellik değerleri, yönetim aracı ile program kodu arasında farklılık gösterir.

IBM MQ classes for JMS , IBM MQ JMS yönetim aracını, IBM MQ Gezgini 'ni ya da bir uygulamada kullanarak nesnelere özelliklerini ayarlamak ve sorgulamak için olanaklar sağlar. Özelliklerin çoğu yalnızca nesne tiplerinin belirli bir alt kümesiyle ilgilidir.

IBM MQ JMS yönetim aracını nasıl kullandığınıza ilişkin bilgi için [Yönetim aracını kullanarak JMS nesnelere yapılandırma](#) başlıklı konuya bakın.

Çizelge 868 sayfa 1864 , her özelliğin kısa bir açıklamasını verir ve her özellik için geçerli olduğu nesne tiplerini gösterir. Nesne tipleri anahtar sözcükler kullanılarak tanımlanır; bu nesnelere ilişkin açıklamalar için [Denetim aracını kullanarak JMS nesnelere yapılandırma](#) başlıklı konuya bakın.

Numaralar, tablonun sonundaki notlara başvurmalıdır. Ayrıca bkz. [“IBM MQ classes for JMS nesnelere özellikleri arasındaki bağımlılıklar” sayfa 1867.](#)

Bir özellik, aşağıdaki biçimde bir ad-değer çiftinden oluşur:

```
PROPERTY_NAME(property_value)
```

Bu bölümdeki konular, her özellik için, özelliğin adı ve kısa bir tanımı listeler ve yönetim aracında kullanılan geçerli özellik değerlerini gösterir. ve bir uygulamadaki özelliğin değerini ayarlamak için kullanılan set yöntemi. Bu konular ayrıca, her özellik için geçerli özellik değerlerini ve araçta kullanılan simgesel özellik değerleri ile bunların programlanabilir eşdeğerleri arasındaki eşlemeleri de gösterir.

Özellik adları büyük ve küçük harfe duyarlı değildir ve bu konularda gösterilen tanınan adlar kümesiyle sınırlıdır.

Özellik	Kısa Biçim	Nesne tipi							
		CF	QCF	TCF.	Q	T	XACF	XAQCF	XATCF
“UYGULAMAADI” sayfa 1869	appName	Y	Y	Y			Y	Y	Y
“ZAMANUYUMSUZ KURAL Dışı DURUMU” sayfa 1869	AEX	Y	Y	Y			Y	Y	Y
“BALOPTIONS” sayfa 1870	OPTIONS	Y	Y	Y			Y	Y	Y
“BALTIP TIPI” sayfa 1871	TİP	Y	Y	Y			Y	Y	Y
“BALTIMEOUT” sayfa 1872	TIMEOUT	Y	Y	Y			Y	Y	Y
“BROKERCDURSUBQ” sayfa 1872 ¹	CCDSUB					Y			
“BROKERCCSUBQ” sayfa 1873 ¹	CCSID	Y		Y			Y		Y
“BROKERCONQ” sayfa 1873 ¹	BCON	Y		Y			Y		Y
“BROKERDURSUBQ” sayfa 1873 ¹	BDSUB					Y			
“BROKERPUBQ” sayfa 1874 ¹	BPUB	Y		Y		Y	Y		Y
“BROKERPUBQMGR” sayfa 1874 ¹	BPQM (Kalite Yönetimi)					Y			
“BROKERQMGR” sayfa 1875 ¹	BQM	Y		Y			Y		Y
“BROKERSUBQ” sayfa 1875 ¹	BSUB	Y		Y			Y		Y

Çizelge 868. Özellik adları ve uygulanabilir nesne tipleri (devamı var)

Özellik	Kısa Biçim	Nesne tipi								
		CF	QCF	TCF.	Q	T	XACF	XAQCF	XATCF	
“ARALIKÇI” sayfa 1876 ¹	BVER	E ²		E ²		Y	Y		Y	
“CCDTURL” sayfa 1876 ³	CCDT	Y	Y	Y			Y	Y	Y	
“CCSID” sayfa 1877	CCS	Y	Y	Y	Y	Y	Y	Y	Y	
“Kanal” sayfa 1877 ³	Chan	Y	Y	Y			Y	Y	Y	
“Temizle” sayfa 1878 ¹	CL	Y		Y			Y		Y	
“TEMİZLEME” sayfa 1878 ¹	KESIM	Y		Y			Y		Y	
“BAĞLANTI ADAMI” sayfa 1879	CNLIST	Y	Y	Y						
“CLIENTRECONNECTOPTIONS” sayfa 1879	KIRPMA	Y	Y	Y						
“CLIENTRECONNECTTIMEOUT” sayfa 1880	CRT	Y	Y	Y						
“İSTEMCIID” sayfa 1881	CID	E ²	Y	E ²			Y	Y	Y	
“KLONESUPP” sayfa 1881	CLS	Y		Y			Y		Y	
“TAMAMLANDI” sayfa 1881	HC	Y		Y			Y		Y	
“ŞİRKET” sayfa 1882	MC	Y	Y	Y			Y	Y	Y	
“BİRLEŞTİR” sayfa 1882	CNOPT	Y	Y	Y			Y	Y	Y	
“CONNTAG (ETIKET)” sayfa 1883	CNTAG	Y	Y	Y			Y	Y	Y	
“TANIM” sayfa 1884	ALÇ	E ²	Y	E ²	Y	Y	Y	Y	Y	
“DIRECTAUTH” sayfa 1884	DEUTH.	E ²		E ²						
“Kodlama” sayfa 1885	ENC				Y	Y				
“Son kullanma tarihi” sayfa 1886	ÜS				Y	Y				
“FAILIFQUIESCE” sayfa 1886	FIQ	Y	Y	Y	Y	Y	Y	Y	Y	
“ANASİSTEM ADI” sayfa 1887	Anasistem	E ²	Y	E ²			Y	Y	Y	
“YERELADRES” sayfa 1887	LA	E ²	Y	E ²			Y	Y	Y	
“MAPNAMESTYLE” sayfa 1888	MNST	Y	Y	Y			Y	Y	Y	
“MAXBUFFSIZE” sayfa 1889	MBSZ	E ²		E ²						
“MDREAD” sayfa 1889	MDR				Y	Y				
“MDYAZMA” sayfa 1890	MDW				Y	Y				
“MDMSGCTX” sayfa 1890	MDCTX				Y	Y				
“MSGBATCHSZ” sayfa 1891 ¹	MBS	Y	Y	Y			Y	Y	Y	
“MSGBODY (DOSYA)” sayfa 1891	MBODY				Y	Y				
“MSGREÇEKİŞME” sayfa 1892	MRET	Y	Y				Y	Y		
“MSGSELECTION” sayfa 1892 ¹	MSEL	Y		Y			Y		Y	
“ÇOKLU YAYIN” sayfa 1893	MCAST	E ²		E ²		Y				

Çizelge 868. Özellik adları ve uygulanabilir nesne tipleri (devamı var)

Özellik	Kısa Biçim	Nesne tipi							
		CF	QCF	TCF.	Q	T	XACF	XAQCF	XATCF
“OPTIMIST YAYIN” sayfa 1893 ¹	OPTPUB	Y		Y					
“OUTCOMENOTIFICATION” sayfa 1894 ¹	Bildirim Gönder	Y		Y					
“Kalıcılık” sayfa 1894	HER BİRİ İÇİN				Y	Y			
“POLLINGİT” sayfa 1895 ¹	NOKTA	Y	Y	Y			Y	Y	Y
“PORT” sayfa 1895	PORT	E ²	Y	E ²			Y	Y	Y
“Öncelik” sayfa 1896	PRI				Y	Y			
“SÜREÇ SÜRESİ” sayfa 1896 ¹	PROCDUR	Y		Y					
“PROVIDERVERSION” sayfa 1897	PVER	Y	Y	Y			Y	Y	Y
“PROXYHOSTNAME” sayfa 1899	PHOST	E ²		E ²					
“PROXYPORT” sayfa 1899	SINIR	E ²		E ²					
“GENEL” sayfa 1900 ¹	PAI	Y		Y			Y		Y
“PUTASYNCALKUTSAL” sayfa 1900	PAALD				Y	Y			
“YöNETİCİ” sayfa 1901	QMGR	Y	Y	Y	Y		Y	Y	Y
“kuyruk” sayfa 1901	QU				Y				
“READAHEADALLOWED” sayfa 1902	RAALD				Y	Y			
“READAHEADCLOSEPOLICY” sayfa 1902	RACP				Y	Y			
“RECEIVECCSID” sayfa 1903	Uzak denetleyici sistem				Y	Y			
“ALAVECONVERSION” sayfa 1903	RCNV				Y	Y			
“ALMA YALITMA” sayfa 1904 ¹	RCVISOL	Y		Y					
“YENİDEN ÇIK” sayfa 1904	RCX	Y	Y	Y			Y	Y	Y
“RECEXITINIT” sayfa 1905	RCXI	Y	Y	Y			Y	Y	Y
“REPLYTOSTYLE” sayfa 1905	RTOST				Y	Y			
“İLK DURUMUNA GEÇİR” sayfa 1906 ¹	NOKTA	Y	Y				Y	Y	
“SECEXIT” sayfa 1906	SCX	Y	Y	Y			Y	Y	Y
“SECEXITINIT” sayfa 1907	SCXI	Y	Y	Y			Y	Y	Y
“SENDCHECKCOUNT” sayfa 1907	SCC.	Y	Y	Y			Y	Y	Y
“SENDEXIT” sayfa 1908	SDX	Y	Y	Y			Y	Y	Y
“SENDEXITINIT” sayfa 1908	SDXI	Y	Y	Y			Y	Y	Y
“FARKLILIKLI” sayfa 1909	SKALD	Y	Y	Y			Y	Y	Y
“SPARSESUB” sayfa 1909 ¹	SSUBS	Y		Y					

Çizelge 868. Özellik adları ve uygulanabilir nesne tipleri (devamı var)

Özellik	Kısa Biçim	Nesne tipi							
		CF	QCF	TCF.	Q	T	XACF	XAQCF	XATCF
“ŞİFRELEME TAKIMI” sayfa 1910	SCPHS	Y	Y	Y			Y	Y	Y
“SSLCRL” sayfa 1910	SCRL	Y	Y	Y			Y	Y	Y
“SSLFIPSREQUIRED” sayfa 1911	SFIPS	Y	Y	Y			Y	Y	Y
“SSLPEERNAME” sayfa 1911	SPEER	Y	Y	Y			Y	Y	Y
“SSLRESETCOUNT” sayfa 1912	src	Y	Y	Y			Y	Y	Y
“STATREFRESHINT” sayfa 1912 ¹	SRU	Y		Y			Y		Y
“ALT MAĞAZA” sayfa 1913 ¹	SS	Y		Y			Y		Y
“SYNCPOINTALLYAKALAR” sayfa 1913	SPAG.	Y	Y	Y			Y	Y	Y
“HEDEF” sayfa 1914	TC				Y	Y			
“TARGCLIENTEŞLEŞME” sayfa 1914	TCM	Y	Y				Y	Y	
“TEMPMODEL” sayfa 1915	TM	Y	Y				Y	Y	
“TEMPQPREFIX” sayfa 1915	TQP	Y	Y				Y	Y	
“TEMPTOPICPREFIX” sayfa 1916	TTP	Y		Y			Y		Y
“Konu” sayfa 1916	TOP					Y			
“Aktarım” sayfa 1917	TRAN	E ²	Y	E ²			Y	Y	Y
“WILDCARDFORMAT” sayfa 1917	WCFMT	Y		Y			Y		Y

Not:

1. Bu özellik, IBM MQ classes for JMS ' un 70 sürümüyle kullanılabilir; ancak, bağlantı üreticisinin PROVIDERVERSION özelliği 7 'den küçük bir sürüm numarasına ayarlanmadıkça, IBM WebSphere MQ 7.0 kuyruk yöneticisine bağlı bir uygulama için herhangi bir etkisi yoktur.
2. Bir aracıya gerçek zamanlı bağlantı kullanıldığında ConnectionFactory ya da TopicConnectionFactory nesnesi için yalnızca BROKERVER, CLIENTID, DESCRIPTION, DIRECTAUTH, HOSTNAME, LOCALADDRESS, MAXBUFFSIZE, MULTICAST, PORT, PROXYHOSTNAME, PROXYPORT ve TRANSPORT özellikleri desteklenir.
3. Bir nesnenin CCDURL ve CHANNEL özelliklerinin her ikisi de aynı anda ayarlanmamalıdır.

IBM MQ classes for JMS nesnelerinin özellikleri arasındaki bağımlılıklar

Bazı özelliklerin geçerliliği, diğer özelliklerin belirli değerlerine bağlıdır.

Bu bağımlılık aşağıdaki özellik gruplarında oluşabilir:

- İstemci Özellikleri
- Bir aracıya gerçek zamanlı bağlantı için özellikler
- Kullanıma hazırlama dizgilerinden çık

İstemci Özellikleri

Kuyruk yöneticisine yönelik bir bağlantı için, aşağıdaki özellikler yalnızca TRANSPORT, CLIENT değerine sahipse geçerlidir:

- ANASİSTEM ADI

- PORT
- Kanal
- YERELADRES
- CCDTURL
- CCSID
- TAMAMLANDI
- ŞİRKET
- YENIDEN ÇIK
- REEXITINIT
- SEEXIT
- SEEXITINIT
- SENDEXIT
- SENDEXITINIT
- FARKLILIKLI
- ŞİFRELEME TAKIMI
- SSLCRL
- SSLFIPSREQUIRED
- SSLPEERNAME
- SSLRESETCOUNT
- UYGULAMAADI

TRANSPORT, BIND değerine sahipse, denetim aracını kullanarak bu özellikler için değer ayarlayamazsınız.

TRANSPORT işlevi CLIENT değerini içeriyorsa, BROKERVER özelliğinin varsayılan değeri V1 ve PORT özelliğinin varsayılan değeri 1414 'tür. BROKERVER ya da PORT değerini belirttik olarak ayarlarsanız, TRANSPORT değerinde daha sonra yapılan bir değişiklik seçimlerinizi geçersiz kılmaz.

Bir aracıya gerçek zamanlı bağlantı için özellikler

TRANSPORT, DIRECT ya da DIRECTHTTP değerine sahipse yalnızca aşağıdaki özellikler geçerlidir:

- ARALIKÇI
- İSTEMCIID
- TANIM
- DIRECTAUTH
- ANASİSTEM ADI
- YERELADRES
- MAXBUFFSIZE
- MULTICAST (yalnızca DIRECT için desteklenir)
- PORT
- PROXYHOSTNAME (yalnızca DIRECT için desteklenir)
- PROXYPORT (yalnızca DIRECT için desteklenir)

DIRECT ya da DIRECTHTTP değerine sahipse, BROKERVER özelliğinin varsayılan değeri V2ve PORT özelliğinin varsayılan değeri 1506 'dır. BROKERVER ya da PORT değerini belirttik olarak ayarlarsanız, TRANSPORT değerinde daha sonra yapılan bir değişiklik seçimlerinizi geçersiz kılmaz.

Kullanıma hazırlama dizgilerinden çık

İlgili çıkış adını belirtmeden çıkış kullanıma hazırlama dizgilerinin hiçbirini ayarlamayın. Çıkış kullanıma hazırlama özellikleri şunlardır:

- RECEXITINIT
- SECEXITINIT
- SENDEXITINIT

Örneğin, RECEXIT(some.exit.classname) belirtilmeden RECEXITINIT(myString) belirtilmesi bir hataya neden olur.

İlgili başvurular

[“Aktarım” sayfa 1917](#)

Bir kuyruk yöneticisine ya da aracıya yönelik bağlantının türü.

UYGULAMAADI

Bir uygulama, kuyruk yöneticisiyle bağlantısını tanıtan bir ad ayarlayabilir. Bu uygulama adı **DISPLAY CONN MQSC/PCF** komutuyla gösterilir (burada alan **APPLTAG** olarak adlandırılır) ya da IBM MQ Gezgin **Uygulama Bağlantıları** ekranında (alanın **App name** olarak adlandırıldığı yerde).

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: APPLICATIONNAME

JMS yönetim aracı kısa adı: APPNAME

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setAppAdı ()
- MQConnectionFactory.getAppAdı ()

Değerler

28 karakterden uzun olmayan herhangi bir geçerli dize. Daha uzun adlar, gerekirse, baştaki paket adları kaldırılarak sığacak şekilde ayarlanır. Örneğin, çağıran sınıf com.example.MainApp ise, tam ad kullanılır, ancak çağıran sınıf com.example.dictionaryAndThesaurus.multilingual.mainApp ise, kullanılabilir uzunluğa uyan en uzun sınıf adı ve en sağdaki paket adı birleşimi olduğundan multilingual.mainApp adı kullanılır.

Sınıf adının kendisi 28 karakterden uzunsa, sığacak şekilde kesilir. Örneğin, com.example.mainApplicationForSecondTestCase, mainApplicationForSecondTest olur.

 z/OS üzerinde, APPNAME:

- Ayarlanırsa bağ tanımlama kipi dikkate alınmaz ve ayarlanırsa, yalnızca boşluk olarak ayarlanabilir.
- İstemci kipi ayarlanabilir ve kullanılabilir.

ZAMANUYUMSUZ KURAL DIŞI DURUMU

Bu özellik, IBM MQ classes for JMS 'un bir ExceptionListener ' a yalnızca bir bağlantı kesildiğinde mi, yoksa bir JMS API çağrısına zamanuyumsuz olarak herhangi bir kural dışı durum oluştuğunda mı bilgi verdiğini belirler. Bu, ExceptionListener kaydı olan bu ConnectionFactory ' den oluşturulan tüm Connections için geçerlidir.

Uygulanabilir nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: ASYNCEXCEPTION

JMS yönetim aracı kısa adı: AEX

Programlı erişim

Ayarlayıcılar/Alıcılar

- MQConnectionFactory.setAsyncKural Dışı Durumlar ()
- MQConnectionFactory.getAsyncKural Dışı Durumlar ()

Değerler

ASYNC_EXCEPTIONS_ALL

Zamanuyumlu bir API çağrısının kapsamı dışında zamanuyumsuz olarak saptanan herhangi bir kural dışı durum ve tüm bağlantı bozuk kural dışı durumlar ExceptionListener' a gönderilir.

<i>Çizelge 869. Tüm zamanuyumsuz kural dışı durumlar: ortamlar ve ilgili sabit adlar</i>	
Çevre	Değer
JMS Denetim Aracı	TÜMÜ
Programlı	WMQCONSTANTS.ASYNC_EXCEPTIONS_ALL = -1
IBM MQ Explorer	Tümü

ASYNC_EXCEPTIONS_CONNECTIONBROKEN

ExceptionListenerolanağına yalnızca bozuk bir bağlantıyı gösteren kural dışı durumlar gönderilir. Zamanuyumsuz işleme sırasında oluşan diğer kural dışı durumlar ExceptionListener' a raporlanmaz ve bu nedenle uygulamaya bu kural dışı durumlar bildirilmez. Bu, IBM MQ 8.0.0 Fix Pack 2' in varsayılan değeridir. Bkz. [JMS: IBM MQ 8.0](#) içinde özel durum dinleyici değişiklikleri.

<i>Çizelge 870. Bozuk bir bağlantıyı gösteren kural dışı durumlar: ortamlar ve ilgili değişmez adlar</i>	
Çevre	Değer
JMS Denetim Aracı	BAĞLANTı KESİLDİ
Programlı	WMQCONSTANTS.ASYNC_EXCEPTIONS_CONNECTIONBROKEN = 1
IBM MQ Explorer	Bağlantı Kesildi

Aşağıdaki ek değişmez tanımlandı:

- Kaynak IBM MQ 8.0.0 Fix Pack 2: WMQCONSTANTS.ASYNC_EXCEPTIONS_DEFAULT = ASYNC_EXCEPTIONS_CONNECTIONBROKEN
- IBM MQ 8.0.0 Fix Pack 2öncesinde: WMQCONSTANTS.ASYNC_EXCEPTIONS_DEFAULT = ASYNC_EXCEPTIONS_ALL

İlgili kavramlar

[IBM MQ classes for JMS içindeki kural dışı durumlar](#)

V 9.3.4 BALOPTIONS

İstemci iletimini kullanan IBM MQ classes for JMS uygulamalarının tek tip kümelerde nasıl yeniden dengelendiğini denetler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

JMS yönetim aracı uzun adı: **BALOPTIONS**

JMS yönetim aracı kısa adı: **OPTIONS**

Programlı erişim

Ayarlayıcılar/alıcılar

- `MQConnectionFactory.setBalancingOptions()`
- `MQConnectionFactory.getBalancingOptions()`

Değerler

IGNNONE

Hareketlerin olağan işleme uygulanır ve uygulamaların hareket sırasında taşınması istenmez.

Bu değer IBM MQ *BalancingOption* `MQBNO_OPTIONS_NONE` ile eşlenir.

IGNTRANS

Bir işlem sırasında uygulamaların taşınması istenebilir.

Bu değer, IBM MQ *BalancingOption* `MQBNO_OPTIONS_IGNORE_TRANS` ile eşlenir.

V 9.3.4 BALTIP TIPI

İstemci iletimini kullanan IBM MQ classes for JMS uygulamalarının tek tip bir kümede nasıl yeniden dengelenebileceğini denetler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

JMS yönetim aracı uzun adı: **BALTYPE**

JMS yönetim aracı kısa adı: **TYPE**

Programlı erişim

Ayarlayıcılar/alıcılar

- `MQConnectionFactory.setBalancingApplicationType()`
- `MQConnectionFactory.getBalancingApplicationType()`

Değerler

basit

Tek biçimli bir kümedeki uygulamaların varsayılan işleme geçerlidir.

Bu değer IBM MQ *BalancingOption* `MQBNO_BALTYPE_SIMPLE` ile eşlenir.

İSTEK_YANIT

Zamanaşımı süresi geçmedikçe, **MQPUT** bir **MQGET** tarafından dengelenmediyse, uygulamanın yeniden bağlanması istenmez.

Bu değer, IBM MQ *BalancingOption* `MQBNO_BALTYPE_REQREP` ile eşlenir.

V 9.3.4 BALTIMEOUT

İstemci iletimini kullanan IBM MQ classes for JMS uygulamalarının tek bir kümede nasıl yeniden dengeleneceğini denetler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

JMS yönetim aracı uzun adı: **BALTIMEOUT**

JMS yönetim aracı kısa adı: **TIMEOUT**

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setBalancingTimeout()
- MQConnectionFactory.getBalancingTimeout()

Değerler

hiçbir zaman

Uygulama, tek tip bir kümede yeniden dengeleme amacıyla hiçbir zaman zamanaşımına uğramaz.

Bu değer, IBM MQ *BalancingOption* MQBNO_TIMEOUT_NOTile eşlenir.

immediate

Uygulama, tek tip bir kümede yeniden dengeleme amacıyla hemen zaman aşımına uğratır.

Bu değer, IBM MQ *BalancingOption* MQBNO_TIMEOUT_IMMEDIATEile eşlenir.

VARSAYILAN

Uygulama, 10 saniyelik varsayılan süreden sonra tek biçimli bir kümede yeniden dengeleme amacıyla zamanaşımına uğramıştır.

Bu değer, IBM MQ *BalancingOption* MQBNO_TIMEOUT_AS_DEFAULTile eşlenir.

nn

Uygulama, *nn* saniyelik bir süre sonra tek tip bir kümede yeniden dengeleme amacıyla zamanaşımına uğramıştır.

nn , 1 ile 999999999999 arasında olabilir.

BROKERCDURSUBQ

ConnectionConsumer için sürekli abonelik iletilerinin alındığı kuyruğun adı.

Uygulanabilir nesnelere

Konu

JMS yönetim aracı uzun adı: BROKERCCDURSUBQ

JMS yönetim aracı kısa adı: CCDSUB

Programlı erişim

Ayarlayıcılar/alıcılar

- MQTopic.setBrokerCCDurSubQueue()
- MQTopic.getBrokerCCDurSubQueue()

Değerler

SYSTEM.JMS.D.CC.SUBSCRIBER.QUEUE

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

BROKERCCSUBQ

ConnectionFactoryConsumer için sürekli olmayan abonelik iletilerinin alındığı kuyruğun adı.

Uygulanabilir nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: BROKERCCSUBQ

JMS yönetim aracı kısa adı: CCSUB

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setBrokerCCSubQueue()
- MQConnectionFactory.getBrokerCCSubQueue()

Değerler

SYSTEM.JMS.ND.CC.SUBSCRIBER.QUEUE

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

BROKERCONQ

Aracının denetim kuyruğu adı.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: BROKERCONQ

JMS yönetim aracı kısa adı: BCON

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setBrokerControlQueue()
- MQConnectionFactory.getBrokerControlQueue()

Değerler

SYSTEM.BROKER.CONTROL.QUEUE

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

BROKERDURSUBQ

IBM MQ classes for JMS , IBM MQ ileti alışverişi sağlayıcısı geçiş kipinde kullanılırken, bu özellik sürekli abonelik iletilerinin alındığı kuyruğun adını belirtir.

Uygulanabilir nesnelere

Konu

JMS yönetim aracı uzun adı: BROKERDURSUBQ

JMS yönetim aracı kısa adı: BDSUB

Programlı erişim

Ayarlayıcılar/alıcılar

- MQTopic.setBrokerDurSubQueue()
- MQTopic.getBrokerDurSubQueue()

Değerler

SYSTEM.JMS.D.SUBSCRIBER.QUEUE

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

SYSTEM.JMS.D

İlgili görevler

JMS **PROVIDERVERSION** özelliğinin yapılandırılması

BROKERPUBQ

Yayınlanan iletilerin gönderildiği kuyruğun adı (akış kuyruğu).

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionÜreticisi, Konu, XAConnectionFactory, XATopicConnectionÜreticisi

JMS yönetim aracı uzun adı: BROKERPUBQ

JMS yönetim aracı kısa adı: BPUB

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setBrokerPubQueue
- MQConnectionFactory.getBrokerPubQueue

Değerler

SYSTEM.BROKER.DEFAULT.STREAM

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

BROKERPUBQMGR

Konu üzerinde yayınlanan iletilerin gönderildiği kuyruğun iyesi olan kuyruk yöneticisinin adı.

Uygulanabilir Nesnelere

Konu

JMS yönetim aracı uzun adı: BROKERPUBQMGR

JMS yönetim aracı kısa adı: BPQM

Programlı erişim

Ayarlayıcılar/alıcılar

- MQTopic.setBrokerPubQueueManager()
- MQTopic.getBrokerPubQueueManager()

Değerler

boş değerli

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

BROKERQMGR

Aracının çalıştığı kuyruk yöneticisinin adı.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: BROKERQMGR

JMS yönetim aracı kısa adı: BQM

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setBrokerQueueManager()
- MQConnectionFactory.getBrokerQueueManager()

Değerler

boş değerli

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

BROKERSUBQ

IBM MQ classes for JMS , IBM MQ ileti alışverişi sağlayıcısı geçiş kipinde kullanıldığında bu özellik, kalıcı olmayan abonelik iletilerinin alındığı kuyruğun adını belirtir.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: BROKERSUBQ

JMS yönetim aracı kısa adı: BSUB

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setBrokerSubQueue()
- MQConnectionFactory.getBrokerSubQueue()

Değerler

SYSTEM.JMS.ND.SUBSCRIBER.QUEUE

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

SYSTEM.JMS.ND

İlgili görevler

JMS **PROVIDERVERSION** özelliğinin yapılandırılması

ARALIKÇI

Kullanılmakta olan aracının sürümü.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: BROKERVER

JMS yönetim aracı kısa adı: BVER

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setBrokerSürüm ()
- MQConnectionFactory.getBrokerSürüm ()

Değerler

V1

IBM MQ Yayınlama/Abone Olma aracısını kullanmak ya da uyumluluk kipinde bir IBM MQ Integrator, WebSphere Event Broker, WebSphere Business Integration Event Broker ya da WebSphere Business Integration Message Broker aracısını kullanmak için. TRANSPORT, BIND ya da CLIENT olarak ayarlandıysa, varsayılan değer budur.

V2

Yerli kipte bir IBM MQ Integrator, WebSphere Event Broker, WebSphere Business Integration Event Broker ya da WebSphere Business Integration Message Broker aracısını kullanmak için. TRANSPORT değeri DIRECT ya da DIRECTHTTP olarak ayarlandıysa, varsayılan değer budur.

belirtilmedi

Aracı V6 sürümünden V7sürümüne geçtikten sonra, bu özelliği RFH2 üstbilgilerinin artık kullanılmayacak şekilde ayarlayın. Geçişten sonra bu özellik artık ilgili değildir.

CCDTURL

İstemci kanal tanımlama çizelgesini içeren dosyanın adını ve yerini tanımlayan ve dosyaya nasıl erişilebileceğini belirten Tek Tip Kaynak Konum Belirleyici (URL).

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: CCDTURL

JMS yönetim aracı kısa adı: CCDT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setCCDTURL()
- MQConnectionFactory.getCCDTURL()

Değerler

boş değerli

Bu varsayılan değerdir.

Birörnek Kaynak Yeri Belirleyici (URL)

CCSID

Bağlantı üreticileri için bu özellik, kuyruk yöneticisiyle iç veri akışları için kullanılacak kodlanmış karakter takımı tanıtıcısını (CCSID) belirler. Hedefler için bu özellik, MapMessages, StreamMessagesve TextMessages içindeki dizgi verilerini bu hedefe kodlamak için kullanılacak CCSID 'yi tanımlar.

Not: Normalde bağlantı üreticileri için bu özelliğin değiştirilmesi gerekmez.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, Queue, Topic, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS denetim aracı uzun adı: CCSID

JMS yönetim aracı kısa adı: CCS

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setCCSID()
- MQConnectionFactory.getCCSID()

Değerler

819

Bir bağlantı üreticisinin varsayılan değeri.

1208

Bir hedefe ilişkin varsayılan değer.

Pozitif bir tamsayı

İlgili kavramlar

[JMS ileti dönüştürme](#)

Kanal

Kullanılmakta olan istemci bağlantı kanalının adı.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: CHANNEL

JMS yönetim aracı kısa adı: CHAN

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setChannel()
- MQConnectionFactory.getChannel()

Değerler

SYSTEM.DEF.SVRCONN

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

Temizle

BROKER ya da MIGRATE Abonelik Depoları için Temizleme Düzeyi.

Uygulanabilir Nesneler

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: CLEANUP

JMS yönetim aracı kısa adı: CL

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setCleanupLevel ()
- MQConnectionFactory.getCleanupLevel ()

Değerler

GÜVENLİ

Güvenli temizlik kullan. Bu varsayılan değerdir.

ÖSÖZ PROP

Java komut satırında ayarlanan bir özelliğe göre güvenli, güçlü ya da temizleme işlemi yapma.

YOK

Temizlik yapma.

güçlü

Güçlü bir temizlik kullanın.

TEMİZLEME

Yayınlama/abone olma temizleme yardımcı programının arka plan yürütmeleri arasındaki milisaniye cinsinden aralık.

Uygulanabilir Nesneler

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: CLEANUPINT

JMS yönetim aracı kısa adı: CLINT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setCleanupInterval ()
- MQConnectionFactory.getCleanupInterval ()

Değerler

3600000

Bu varsayılan değerdir.

Pozitif bir tamsayı

BAĞLANTI ADAMI

TCP/IP bağlantı adlarının listesi. Liste, her yeniden bağlanma yeniden deneme girişiminde bir kez sırayla denenir.

Uygulanabilir Nesneler

ConnectionFactory, QueueConnectionÜreticisi, TopicConnectionÜreticisi

JMS yönetim aracı uzun adı: CONNECTIONNAMELIST

JMS denetim aracı kısa adı: CNLIST

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setconnectionNameList ()
- MQConnectionFactory.getconnectionNameList ()

Değerler

HOSTNAME (PORT) virgülle ayrılmış listesi. HOSTNAME bir DNS adı ya da IP adresi olabilir.

PORT varsayılan değeri 1414 'dür.

CLIENTRECONNECTOPTIONS

Yeniden bağlantıyı yöneten seçenekler.

Uygulanabilir Nesneler

ConnectionFactory, QueueConnectionÜreticisi, TopicConnectionÜreticisi

JMS yönetim aracı uzun adı: CLIENTRECONNECTOPTIONS

JMS yönetim aracı kısa adı: CROPT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setClientReconnectOptions()
- MQConnectionFactory.getClientReconnectOptions()

Değerler

QMGR

Uygulama, başlangıçta bağlı olduğu aynı kuyruk yöneticisine yeniden bağlanabilir.

Uygulamanın bağlanmayı denediği kuyruk yöneticisi, bağlantı adı listesinde belirtildiği gibi, ilk bağlandığı kuyruk yöneticisine farklı bir QMID 'si varsa, MQRC_RECONNECT_QMID_MISMATCH neden koduyla bir hata döndürülür.

Bir uygulama yeniden bağlanabiliyorsa, ancak IBM MQ classes for JMS uygulaması ile ilk bağlantı kurduğu kuyruk yöneticisi arasında bir benzerlik varsa bu değeri kullanın.

Bir uygulamanın yüksek kullanılabilirlikli bir kuyruk yöneticisinin yedek yönetim ortamına otomatik olarak yeniden bağlanmasını istiyorsanız bu değeri seçin.

Bu değeri programlı olarak kullanmak için WMQConstants.WMQ_CLIENT_RECONNECT_Q_MGRdeğişmezini kullanın.

Fark Etmez

Uygulama, bağlantı adı listesinde belirtilen kuyruk yöneticilerinden herhangi birine yeniden bağlanabilir.

Yeniden bağlanma seçeneğini yalnızca, JMS uygulaması için IBM MQ sınıfları ile başlangıçta bağlantı kurduğu kuyruk yöneticisi arasında benzerlik yoksa kullanın.

Bir programdaki bu değeri kullanmak için WMQConstants.WMQ_CLIENT_RECONNECTdeğişmezini kullanın.

DEVRE DIŞI

Uygulama yeniden bağlanmayacak.

Bu değeri programlı olarak kullanmak için WMQConstants.WMQ_CLIENT_RECONNECT_DISABLEDdeğişmezini kullanın.

VARLIK

Uygulamanın otomatik olarak yeniden bağlanıp bağlanmayacağı, IBM MQ kanal özneliği DefReconnect' in değerine bağlıdır.

Bu varsayılan değerdir.

Bir programdaki bu değeri kullanmak için WMQConstants.WMQ_CLIENT_RECONNECT_AS_DEFdeğişmezini kullanın.

CLIENTRECONNECTTIMEOUT

Yeniden bağlantı yeniden denemeleri durmadan önce geçilir.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionÜreticisi, TopicConnectionÜreticisi

JMS yönetim aracı uzun adı: CLIENTRECONNECTTIMEOUT

JMS yönetim aracı kısa adı: CRT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setClientReconnectTimeout()
- MQConnectionFactory.setClientReconnectTimeout()

Değerler

Saniye cinsinden aralık. Varsayılan 1800 (30 dakika).

İSTEMCIID

İstemci tanıtıcısı, sürekli abonelikler için uygulama bağlantısını benzersiz olarak tanımlamak için kullanılır.

Uygulanabilir Nesneler

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: CLIENTID

JMS yönetim aracı kısa adı: CID

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setClientId ()
- MQConnectionFactory.getClientId ()

Değerler

boş değerli

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

KLONESUPP

Aynı sürekli konu abonesinin iki ya da daha fazla eşgörünümünün aynı anda çalışıp çalışamayacağını belirler.

Uygulanabilir Nesneler

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: KLONESUPP

JMS yönetim aracı kısa adı: CLS

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setCloneSupport ()
- MQConnectionFactory.getCloneSupport ()

Değerler

DEVRE DIŞI

Sürekli bir konu abonesinin aynı anda yalnızca bir örneği çalıştırılabilir. Bu varsayılan değerdir.

Etkinleştirildi

Aynı sürekli konu abonesinin iki ya da daha fazla eşgörünümü aynı anda çalışabilir, ancak her yönetim ortamının ayrı bir Java sanal makinesinde (JVM) çalışması gerekir.

TAMAMLANDI

Bir bağlantıdaki üstbilgi verilerinin sıkıştırılması için kullanılacak tekniklerin listesi.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: COMPHDR

JMS yönetim aracı kısa adı: HC

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setHdrCompList()
- MQConnectionFactory.getHdrCompList()

Değerler

YOK

Bu varsayılan değerdir.

SYSTEM

RLE ileti üstbilgisi sıkıştırması gerçekleştirilir.

ŞİRKET

Bir bağlantıdaki ileti verilerini sıkıştırmada kullanılacak tekniklerin listesi.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: COMPMMSG

JMS yönetim aracı kısa adı: MC

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMsgCompList()
- MQConnectionFactory.getMsgCompList()

Değerler

YOK

Bu varsayılan değerdir.

Boş karakterlerle ayrılmış olarak aşağıdaki değerlerden birinin ya da daha fazlasının listesi:

RLE ZLIBFAST ZLIBHIGH

BİRLEŞTİR

Bağ tanımlarını kullanan IBM MQ classes for JMS uygulamalarının kuyruk yöneticisine nasıl bağlanacağını denetler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory.

JMS yönetim aracı uzun adı: CONNOPT

JMS yönetim aracı kısa adı: CNOPT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMQConnectionFactoryOptions ()
- MQConnectionFactory.getMQConnectionFactoryOptions ()

Değerler

Standart

Uygulama ile kuyruk yöneticisi arasındaki bağ tanımının türü, kuyruk yöneticisinin *DefaultBindType* özneliğinin değerine bağlıdır. STANDARD değeri, IBM MQ *ConnectOption* MQCNO_STANDARD_BINDING ile eşlenir.

PAYLAŞILAN

Uygulama ve yerel kuyruk yöneticisi aracı ayrı yürütme birimlerinde çalışır, ancak bazı kaynakları paylaşır. Bu değer, IBM MQ *ConnectOption* MQCNO_SHARED_BINDING ile eşlenir.

Yalıtılmış

Uygulama ve yerel kuyruk yöneticisi aracı ayrı yürütme birimlerinde çalışır ve kaynak paylaşmaz. YALITILMIŞ değer, IBM MQ *ConnectOption* MQCNO_ISOLATED_BINDING ile eşlenir.

FastPath

Uygulama ve yerel kuyruk yöneticisi aracı aynı yürütme biriminde çalışır. Bu değer, IBM MQ *ConnectOption* MQCNO_FASTPATH_BINDING ile eşlenir.

SERIALQM

Uygulama, kuyruk yöneticisi kapsamında bağlantı etiketinin dışlayıcı olarak kullanılmasını ister. Bu değer, IBM MQ *ConnectOption* MQCNO_SERIALIZE_CONN_TAG_Q_MGR ile eşlenir.

SERIALQSG

Uygulama, kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubu kapsamında bağlantı etiketinin özel olarak kullanılmasını ister. SERIALQSG değeri, IBM MQ *ConnectOption* MQCNO_SERIALIZE_CONN_TAG_QSG ile eşlenir.

İLK DURUMA GEÇİRİCTQM

Uygulama, bağlantı etiketinin paylaşılan kullanımını ister; ancak, kuyruk yöneticisi kapsamındaki bağlantı etiketinin paylaşılan kullanımıyla ilgili kısıtlamalar vardır. Bu değer, IBM MQ *ConnectOption* MQCNO_RESTRICT_CONN_TAG_Q_MGR ile eşlenir.

RESTRICTQSG

Uygulama, bağlantı etiketinin paylaşılan kullanımını ister, ancak kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubu kapsamında bağlantı etiketinin paylaşılan kullanımıyla ilgili kısıtlamalar vardır. Bu değer, IBM MQ *ConnectOption* MQCNO_RESTRICT_CONN_TAG_QSG ile eşlenir.

IBM MQ bağlantı seçenekleriyle ilgili daha fazla bilgi için [MQCONNX çağrısıyla kuyruk yöneticisine bağlanmabaşlıklı](#) konuya bakın.

CONNTAG (ETIKET)

Uygulama kuyruk yöneticisine bağlıyken, kuyruk yöneticisinin bir iş birimi içinde uygulama tarafından güncellenen kaynaklarla ilişkilendirdiği bir etiket.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: CONNTAG

JMS yönetim aracı kısa adı: CNTAG

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setConnTag ()
- MQConnectionFactory.getConnTag ()

Değerler

Her öğenin 0 olduğu 128 öğeden oluşan bir bayt dizisi

Bu varsayılan değerdir.

Herhangi bir dizgi

Değer 128 byte 'tan uzunsa kesilir.

TANIM

Saklanan nesnenin tanımlaması.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, Queue, Topic, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: DESCRIPTION

JMS yönetim aracı kısa adı: DESC

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setDescription()
- MQConnectionFactory.getDescription()

Değerler

boş değerli

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

DIRECTAUTH

Bir aracıya gerçek zamanlı bağlantıda TLS kimlik doğrulamasının kullanılıp kullanılmayacağını belirler.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory

JMS yönetim aracı uzun adı: DIRECTAUTH

JMS yönetim aracı kısa adı: DAUTH

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setDirectAuth ()
- MQConnectionFactory.getDirectAuth ()

Değerler

Temel Düzey

Kimlik doğrulama, kullanıcı adı kimlik doğrulaması ya da parola kimlik doğrulaması yok. Bu varsayılan değerdir.

Sertifika

Genel anahtar sertifikası kimlik doğrulaması.

Kodlama

İleti bu hedefe gönderildiğinde, iletinin gövdesindeki sayısal verilerin nasıl temsil edileceği. Özellik, ikili tamsayıların, paketlenmiş ondalık tamsayıların ve kayan noktalı sayıların gösterimini belirtir.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS yönetim aracı uzun adı: ENCODING

JMS yönetim aracı kısa adı: ENC

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setEncoding()
- MQDestination.getEncoding()

Değerler

ENCODING özelliği

ENCODING özelliğinin alabileceği geçerli değerler üç alt özellikten oluşturulur:

tamsayı kodlaması

Normal ya da ters çevrilmiş

ondalık kodlama

Normal ya da ters çevrilmiş

kayan nokta kodlaması

IEEE normal, IEEE ters çevrilmiş ya da z/OS

ENCODING özelliği, aşağıdaki sözdizimiyle üç karakterli bir dizgi olarak ifade edilir:

```
{N|R}{N|R}{N|R|3}
```

Bu dizgide:

- N olağan durumu gösterir
- R , tersine çevrildiğini gösterir
- 3 , z/OS ögesini belirtir.
- İlk karakter *tamsayı kodlamasını* temsil eder.
- İkinci karakter *ondalık kodlamayı* temsil eder.
- Üçüncü karakter *kayan nokta kodlamasını* temsil eder

Bu, ENCODING özelliği için on iki olası değer kümesi sağlar.

Java platformu için uygun kodlama değerlerini ayarlayan NATIVEDizgisi olan ek bir değer vardır.

Aşağıdaki örneklerde ENCODING için geçerli birleşimler gösterilmektedir:

```
ENCODING (NNR)
ENCODING (NATIVE)
ENCODING (RR3)
```

Son kullanma tarihi

Bir hedefteki iletilerin süre bitimine kadar olan süre.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS yönetim aracı uzun adı: EXPIRY

JMS yönetim aracı kısa adı: EXP

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setExpiry()
- MQDestination.getExpiry()

Değerler

Uygulama

Süre bitimi JMS uygulaması tarafından tanımlanabilir. Bu varsayılan değerdir.

YALIN

Süre bitimi gerçekleşmez.

0

Süre bitimi gerçekleşmez.

Milisaniye cinsinden süre bitimini gösteren herhangi bir pozitif tamsayı.

FAILIFQUIESCE

Bu özellik, kuyruk yöneticisi susturma durumundaysa ya da bir uygulama CLIENT iletimi kullanılarak bir kuyruk yöneticisine bağlanıyorsa ve uygulamanın kullandığı kanal susturma durumuna getirildiyse (örneğin, **STOP CHANNEL** ya da **STOP CHANNEL MODE (QUIESCE)** MQSC komutu kullanılarak), belirli yöntemlere yönelik çağrılarının başarısız olup olmayacağını belirler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, Queue, Topic, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: FAILIFQUIESCE

JMS yönetim aracı kısa adı: FIQ

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setFailIfQuiesce()
- MQConnectionFactory.getFailIfQuiesce()

Değerler

EVET

Kuyruk yöneticisi susturma durumundaysa ya da bir kuyruk yöneticisine bağlanmak için kullanılan kanal susturuluyorsa, belirli yöntemlere yönelik çağrılar başarısız olur. Bir uygulama bu koşullardan birini saptarsa, uygulama hemen görevini tamamlayabilir ve bağlantıyı kapatarak kuyruk yöneticisinin ya da kanal örneğinin durmasını sağlar. Bu varsayılan değerdir.

HAYIR

Kuyruk yöneticisi ya da bir kuyruk yöneticisine bağlanmak için kullanılan kanal susturma durumunda olduğundan, yöntem çağrısı başarısız olmaz. Bu değeri belirtirseniz, bir uygulama kuyruk yöneticisinin ya da kanalın susturulduğunu saptayamaz. Uygulama kuyruk yöneticisine ilişkin işlemleri gerçekleştirmeye devam edebilir ve kuyruk yöneticisinin durmasını önleyebilir.

ANASİSTEM ADI

Bir kuyruk yöneticisine bağlantı için, kuyruk yöneticisinin çalıştığı sistemin anasistem adı ya da IP adresi ya da bir aracıya gerçek zamanlı bağlantı için, aracının çalıştığı sistemin anasistem adı ya da IP adresi.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: HOSTNAME

JMS yönetim aracı kısa adı: HOST

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setHostName ()
- MQConnectionFactory.getHostName ()

Değerler

localhost

Bu varsayılan değerdir.

Geçerli herhangi bir dizgi

YERELADRES

Bir kuyruk yöneticisine yönelik bağlantı için bu özellik, kullanılacak yerel ağ arabirimini ya da yerel kapıyı ya da yerel kapı aralığını belirtir.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: LOCALADDRESS

JMS yönetim aracı kısa adı: LA

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setLocalAddress ()
- MQConnectionFactory.getLocalAddress ()

Değerler

"" (boş dizgi)

Bu varsayılan değerdir.

[ip-addr] [(düşük kapı [, yüksek kapı])] biçiminde bir dizgi

Bazı örnekler:

192.0.2.0

Kanal, yerel olarak 192.0.2.0 adresine bağlanır.

192.0.2.0(1000)

Kanal, yerel olarak 192.0.2.0 adresine bağlanır ve 1000 numaralı kapıyı kullanır.

192.0.2.0(1000,2000)

Kanal, yerel olarak 192.0.2.0 adresine bağlanır ve 1000-2000 aralığında bir kapı kullanır.

(1000)

Kanal, yerel olarak 1000 numaralı kapıya bağlanır.

(1000,2000)

Kanal, yerel olarak 1000-2000 aralığındaki bir kapıya bağlanır.

IP adresi yerine bir anasistem adı belirleyebilirsiniz. Bir aracıya gerçek zamanlı bağlantı için, bu özellik yalnızca çoklu yayın kullanıldığında anlamlıdır ve özelliğin değeri bir kapı numarası ya da bir kapı numarası aralığı içermemelidir. Bu durumda özelliğin geçerli değerleri yalnızca boş değer, IP adresi ya da anasistem adıdır.

MAPNAMESTYLE

Uyumluluk biçiminin MapMessage öge adları için kullanılmasına izin verir.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: MAPNAMESTYLE

JMS yönetim aracı kısa adı: MNST

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMapNameStyle()
- MQConnectionFactory.getMapNameStyle()

Değerler

Standart

Standart com.ibm.jms.JMSMapMessage öge adlandırma biçimi kullanılır. Bu varsayılan değerdir ve geçerli olmayan Java tanıtıcılarının öge adı olarak kullanılmasına izin verir.

Uyumlu

Daha eski com.ibm.jms.JMSMapMessage öge adlandırma biçimi kullanılır. Öge adı olarak yalnızca geçerli Java tanıtıcıları kullanılabilir. Bu yalnızca, eşlem iletileri 5.3 sürümünden önceki bir IBM MQ classes for JMS sürümünü kullanan bir uygulamaya gönderiliyorsa gereklidir.

MAXBUFFSIZE

Uygulama tarafından işlenmeyi beklerken iç ileti arabelleğinde saklanabilecek alınan ileti sayısı üst sınırı. Bu özellik yalnızca TRANSPORT, DIRECT ya da DIRECTHTTP değerine sahip olduğunda geçerlidir.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionÜreticisi

JMS yönetim aracı uzun adı: MAXBUFFSIZE

JMS yönetim aracı kısa adı: MBSZ

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMaxBufferSize()
- MQConnectionFactory.getMaxBufferSize()

Değerler

1000

Bu varsayılan değerdir.

Pozitif bir tamsayı

MDREAD

Bu özellik, bir JMS uygulamasının MQMD alanlarının değerlerini alıp almayacağını belirler.

Uygulanabilir Nesnelere

JMS yönetim aracı uzun adı: MDREAD

JMS yönetim aracı kısa adı: MDR

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setMQMDReadEnabled()
- MQDestination.getMQMDReadEnabled()

Değerler

HAYIR

İleti gönderilirken, gönderilen bir iletideki JMS_IBM_MQMD* özellikleri MQMD 'deki güncellenen alan değerlerini yansıtacak şekilde güncellenmez. İleti alınırken, JMS_IBM_MQMD* özelliklerinin hiçbiri, gönderen bunların bazılarını ya da tümünü ayarlamış olsa da, alınan bir iletide kullanılamaz. Bu, yönetim araçları için varsayılan değerdir.

Programlar için False (Yanlış) değerini kullanın.

Evet

İleti gönderilirken, gönderilen bir iletideki tüm JMS_IBM_MQMD* özellikleri, gönderenin belirttik olarak ayarlamadığı özellikler de içinde olmak üzere, MQMD 'deki güncellenen alan değerlerini yansıtacak şekilde güncellenir. İleti alınırken, gönderenin belirttik olarak ayarlamadığı özellikler de içinde olmak üzere, alınan bir iletide tüm JMS_IBM_MQMD* özellikleri kullanılabilir.

Programlar için True değerini kullanın.

MDYAZMA

Bu özellik, bir JMS uygulamasının MQMD alanlarının değerlerini ayarlayıp ayarlayamayacağını belirler.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS denetim aracı uzun adı: MDWRITE

JMS yönetim aracı kısa adı: MDR

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setMQMDWriteEnabled()
- MQDestination.getMQMDWriteEnabled()

Değerler

HAYIR

Tüm JMS_IBM_MQMD* özellikleri yoksayılar ve değerleri temel MQMD yapısına kopyalanmaz. Bu, yönetim araçları için varsayılan değerdir.

Programlar için False (Yanlış) değerini kullanın.

EVET

JMS_IBM_MQMD* özellikleri işlendi. Değerleri temeldeki MQMD yapısına kopyalanır.

Programlar için True değerini kullanın.

MDMSGCTX

JMS uygulaması tarafından ayarlanacak ileti bağlamı düzeyi. Bu özelliğin yürürlüğe girmesi için uygulamanın uygun bağlam yetkisiyle çalışması gerekir.

Uygulanabilir Nesnelere

JMS yönetim aracı uzun adı: MDMSGCTX

JMS yönetim aracı kısa adı: MDCTX

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setMQMDMessageContext()
- MQDestination.getMQMDMessageContext()

Değerler

VARSAYILAN

MQOPEN API çağrısı ve MQPMO yapısı belirttik ileti bağlamı seçenekleri belirtmiyor. Bu, yönetim araçları için varsayılan değerdir.

Programlar için WMQ_MDCTX_DEFAULT kullanın.

SET_IDENTITY_CONTEXT

MQOPEN API çağrısı MQOO_SET_IDENTITY_CONTEXT ileti bağlamı seçeneğini belirtiyor ve MQPMO yapısı MQPMO_SET_IDENTITY_CONTEXT ögesini belirtiyor.

Programlar için WMQ_MDCTX_SET_IDENTITY_CONTEXT değerini kullanın.

SET_ALL_CONTEXT

MQOPEN API çağırısı MQOO_SET_ALL_CONTEXT ileti bağlamı seçeneğini belirtiyor ve MQPMO yapısı MQPMO_SET_ALL_CONTEXT ögesini belirtiyor.

Programlar için WMQ_MDCTX_SET_ALL_CONTEXT kullanın.

MSGBATCHSZ

Zamanuyumsuz ileti teslimi kullanılırken bir paketteki bir kuyruktan alınacak ileti sayısı üst sınırı.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: MAXBUFFSIZE

JMS yönetim aracı kısa adı: MBSZ

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMsgBatchSize()
- MQConnectionFactory.getMsgBatchSize()

Değerler

10

Bu varsayılan değerdir.

Pozitif bir tamsayı

MSGBODY (DOSYA)

Bir JMS uygulamasının, ileti bilgi yükünün bir parçası olarak IBM MQ iletisinin MQRFH2 ' ye erişip erişmediğini belirler.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS yönetim aracı uzun adı: WMQ_MESSAGE_BODY

JMS yönetim aracı kısa adı: MBODY

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMessageBodyStyle()
- MQConnectionFactory.getMessageBodyStyle()

Değerler

Belirtilmedi

Gönderirken IBM MQ classes for JMS , WMQ_TARGET_CLIENT değerine bağlı olarak bir MQRFH2 üstbilgisi oluşturur ya da içermez. Alma sırasında JMSdeğeri gibi davranır.

JMS

Gönderirken IBM MQ classes for JMS otomatik olarak bir MQRFH2 üstbilgisi oluşturur ve bunu IBM MQ iletisine ekler.

IBM MQ classes for JMS alırken, JMS ileti özelliklerini MQRFH2 'deki değerlere göre ayarlayın (varsa); MQRFH2 ' yi JMS ileti gövdesinin bir parçası olarak sunmaz.

MQ

Gönderirken IBM MQ classes for JMS , bir MQRFH2oluşturmaz.

IBM MQ classes for JMS , alırken JMS ileti gövdesinin bir parçası olarak MQRFH2 ' yi sunar.

MSGREÇEKİŞME

Bağlantı tüketicisinin teslim edilmemiş iletileri giriş kuyruğunda tutup tutmadığını belirler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory,

JMS denetim aracı uzun adı: MSGREÇEKİŞME

JMS yönetim aracı kısa adı: MRET

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMessageRetention ()
- MQConnectionFactory.getMessageRetention ()

Değerler

Evet

Teslim edilmeyen iletiler giriş kuyruğunda kalır. Bu varsayılan değerdir.

Hayır

Teslim edilmemiş iletiler, yok etme seçeneklerine göre ele alınmaktadır.

MSGSELECTION

İleti seçiminin IBM MQ classes for JMS tarafından mı, yoksa aracı tarafından mı yapılacağını belirler. TRANSPORT değeri DIRECT ise, ileti seçimi her zaman aracı tarafından yapılır ve MSGSELECTION değeri yoksayıdır. BROKERVER V1değerine sahip olduğunda aracı tarafından ileti seçimi desteklenmez.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: MSGSELECTION

JMS yönetim aracı kısa adı: MSEL

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMessageSelection ()
- MQConnectionFactory.getMessageSelection ()

Değerler

CLIENT

İleti seçimi IBM MQ classes for JMS tarafından yapılır. Bu varsayılan değerdir.

Aracı

İleti seçimi aracı tarafından yapılır.

ÇOKLU YAYIN

Bir aracıya gerçek zamanlı bir bağlantıda çoklu yayını etkinleştirmek ve etkinleştirildiyse, aracından ileti tüketicisine ileti teslim etmek için çoklu hedefin nasıl kullanılacağını kesin olarak belirtmek için. Özellik, ileti üreticisinin aracıya ileti gönderme şeklinden etkilenmez.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, Konu

JMS yönetim aracı uzun adı: MULTICAST

JMS yönetim aracı kısa adı: MCAST

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setMulticast()
- MQConnectionFactory.getMulticast()

Değerler

DEVRE DIŞI

İletiler, çoklu yayın iletimi kullanılarak bir ileti tüketicisine teslim edilmez. Bu, ConnectionFactory ve TopicConnectionFactory nesnelere için varsayılan değerdir.

ASCF

İletiler, ileti tüketicisiyle ilişkili bağlantı üreticisine ilişkin çoklu yayın ayarına göre bir ileti tüketicisine teslim edilir. Bağlantı üreticisine ilişkin çoklu yayın ayarı, ileti tüketicisinin oluşturulduğu sırada belirtilir. Bu değer yalnızca Konu nesnelere için geçerlidir ve Konu nesnelere için varsayılan değerdir.

Etkinleştirildi

Konu aracında çoklu yayın için yapılandırıldıysa, iletiler çok hedefli iletim kullanılarak bir ileti tüketicisine teslim edilir. Konu güvenilir çoklu yayın için yapılandırıldıysa, güvenilir bir hizmet kalitesi kullanılır.

GÜVENİLİR

Konu, aracında güvenilir çoklu yayın için yapılandırıldıysa, iletiler, güvenilir bir hizmet kalitesine sahip çoklu yayın iletimi kullanılarak ileti tüketicisine teslim edilir. Konu güvenilir çoklu yayın için yapılandırılmadıysa, konu için bir ileti tüketicisi oluşturamazsınız.

HAYIR.

Konu aracında çoklu yayın için yapılandırıldıysa, iletiler çok hedefli iletim kullanılarak ileti tüketicisine teslim edilir. Konu güvenilir çoklu yayın için yapılandırılmış olsa da güvenilir bir hizmet kalitesi kullanılmaz.

OPTİMİST YAYIN

Bu özellik, IBM MQ classes for JMS 'un denetimi bir ileti yayınlayan bir yayıncıya hemen döndürüp döndürmeyeceğini ya da denetimi yalnızca çağrıyla ilişkili tüm işlemleri tamamladıktan sonra mı döndüreceğini ve sonucu yayıncıya bildirebileceğini belirler.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, Konu

JMS yönetim aracı uzun adı: OPTIYAYIN

JMS yönetim aracı kısa adı: OPTPUB

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setOptimisticPublication ()
- MQConnectionFactory.getOptimisticPublication ()

Değerler

HAYIR

Bir yayınlıyıcı bir ileti yayınladığında, IBM MQ classes for JMS denetimi çağrıyla ilişkili tüm işlemleri tamamlamadan yayınlıyıcıya döndürmez ve sonucu yayınlıyıcıya bildirebilir. Bu varsayılan değerdir.

EVET

Bir yayınlıyıcı bir ileti yayınladığında, IBM MQ classes for JMS denetimi, çağrıyla ilişkili tüm işlemleri tamamlamadan hemen önce yayınlıyıcıya döndürür ve sonucu yayınlıyıcıya bildirebilir. IBM MQ classes for JMS , sonucu yalnızca yayınlıyıcı iletiyi kesinleştirdiğinde bildirir.

OUTCOMENOTIFICATION

Bu özellik, IBM MQ classes for JMS ' un denetimi yalnızca bir iletiyi kabul eden ya da kesinleştirmiş olan bir aboneye hemen döndürüp döndürmeyeceğini ya da denetimi yalnızca çağrıyla ilişkili tüm işlemleri tamamladıktan sonra mı döndürdüğünü ve sonucu aboneye bildirebileceğini belirler.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionÜreticisi

JMS yönetim aracı uzun adı: OUTCOMENOTIFICATION

JMS yönetim aracı kısa adı: NOTIFY

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setOutcomeNotification ()
- MQConnectionFactory.getOutcomeNotification ()

Değerler

EVET

Bir abone bir iletiyi onayladığında ya da kesinleştirdiğinde, IBM MQ classes for JMS , aramayla ilişkili tüm işlemleri tamamlamadan ve sonucu aboneye bildirmeden denetimi aboneye döndürmez. Bu varsayılan değerdir.

HAYIR

Bir abone bir iletiyi onayladığında ya da kesinleştirdiğinde, IBM MQ classes for JMS denetimi çağrıyla ilişkili tüm işlemleri tamamlamadan hemen önce aboneye döndürür ve sonucu aboneye bildirebilir.

Kalıcılık

Bir hedefe gönderilen iletilerin kalıcılığı.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS yönetim aracı uzun adı: PERSISTENCE

JMS yönetim aracı kısa adı: PER

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setPersistence()
- MQDestination.getPersistence()

Değerler

Uygulama

Kalıcılık, JMS uygulaması tarafından tanımlanır. Bu varsayılan değerdir.

QDEF

Kalıcılık, kuyruk varsayılanı değerini alır.

PERS

İletiler kalıcı.

YANLIK

İletiler kalıcı değil.

YÜKSEK

Bu değer kullanımıyla ilgili daha fazla bilgi için bkz. [JMS kalıcı iletiler](#) .

POLLINGİT

Bir oturumdaki her ileti dinleyicinin kuyruğunda uygun bir ileti yoksa, bu, her ileti dinleyicisinin kuyruğundan bir ileti almaya yeniden çalışmasından önce geçmesi için milisaniye cinsinden bu aralığın üst sınırıdır. Sık sık oturumdaki ileti dinleyicilerden herhangi biri için uygun bir ileti yoksa, bu özelliğin değerini artırmayı düşünebilirsiniz. Bu özellik yalnızca TRANSPORT, BIND ya da CLIENT değerine sahipse geçerlidir.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: POLLINGINT

JMS yönetim aracı kısa adı: PINT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setPollingInterval ()
- MQConnectionFactory.getPollingInterval ()

Değerler

5000

Bu varsayılan değerdir.

Pozitif bir tamsayı

PORT

Bir kuyruk yöneticisine yönelik bağlantı için, kuyruk yöneticisinin dinlediği kapının numarası ya da bir aracıya gerçek zamanlı bağlantı için, aracının gerçek zamanlı bağlantıları dinlediği kapının numarası.

Uygulanabilir Nesneler

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: PORT

JMS yönetim aracı kısa adı: PORT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setPort()
- MQConnectionFactory.getPort()

Değerler

1414

TRANSPORT için CLIENT değeri belirlendiyse, varsayılan değer budur.

1506

TRANSPORT değeri DIRECT ya da DIRECTHTTP olarak ayarlandıysa, varsayılan değer budur.

Pozitif bir tamsayı

Öncelik

Bir hedefe gönderilen iletilerin önceliği.

Uygulanabilir Nesneler

Kuyruk, Konu

JMS yönetim aracı uzun adı: PRIORITY

JMS yönetim aracı kısa adı: PRI

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setPriority()
- MQDestination.getPriority()

Değerler

Uygulama

Öncelik, JMS uygulaması tarafından tanımlanır. Bu varsayılan değerdir.

QDEF

Öncelik, kuyruk varsayılan değerini alır.

0-9 aralığındaki herhangi bir tamsayı

En yüksekte en yükseğe.

SÜREÇ SÜRESİ

Bu özellik, bir abonenin denetimi IBM MQ classes for JMS' e döndürmeden önce aldığı herhangi bir iletiyi hızlı bir şekilde işlemlerini garanti edip etmeyeceğini belirler.

Uygulanabilir Nesneler

ConnectionFactory, TopicConnectionÜreticisi

JMS yönetim aracı uzun adı: PROCESSDURATION

JMS denetim aracı kısa adı: PROCDUR

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setProcessDuration ()
- MQConnectionFactory.getProcessSüre ()

Değerler

BİLİNMIYOR

Bir abone, aldığı herhangi bir iletiyi ne kadar hızlı işleyebileceğini garanti edemez. Bu varsayılan değerdir.

short

Bir abone, denetimi IBM MQ classes for JMS' e döndürmeden önce aldığı herhangi bir iletiyi hızlı bir şekilde işleyeceğini garanti eder.

PROVIDERVERSION

Bu özellik, üç IBM MQ ileti alışverişi işletim kipi arasında ayırım yapar: IBM MQ ileti alışverişi sağlayıcısı normal kipi, IBM MQ ileti alışverişi sağlayıcısı normal kipi, kısıtlamalar ve IBM MQ ileti alışverişi sağlayıcısı geçiş kipi.

IBM MQ ileti alışverişi sağlayıcısı normal kipi, JMS' yi gerçekleştirmek için IBM MQ kuyruk yöneticisinin tüm özelliklerini kullanır. Bu kip, JMS 2.0 API 'sini ve işlevselliğini kullanmak üzere eniyilenmiştir. Kısıtlamalar içeren IBM MQ ileti alışverişi sağlayıcısı normal kipi JMS 2.0 API 'sini kullanır, ancak paylaşılan abonelikler, gecikmeli teslim ya da zamanuyumsuz gönderme gibi yeni özellikleri kullanmaz.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: PROVIDERVERSION

JMS yönetim aracı kısa adı: PVER

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setProviderSürüm ()
- MQConnectionFactory.getProviderSürüm ()

Değerler

PROVIDERVERSION özelliğini 8 (normal kip), 7 (kısıtlamalarla normal kip), 6 (geçiş kipi) ya da belirlenmemiş (varsayılan değer) değerlerinden herhangi birine ayarlayabilirsiniz. **PROVIDERVERSION** özelliği için belirttiğiniz değer bir dizgi olmalıdır. 8, 7 ya da 6 seçeneklerinden birini belirtiyorsanız, bunu aşağıdaki biçimlerden herhangi birinde yapabilirsiniz:

- V.R.M.F
- V.R.M
- V.R
- V

Burada V, R, M ve F sıfırdan büyük ya da sıfıra eşit tamsayı değerleridir. Ek R, M ve F değerleri isteğe bağlıdır ve ayrıntılı denetim gerektiğinde kullanmanız! (isteğe bağlı) için kullanılabilir. Örneğin, **PROVIDERVERSION** 7düzeyini kullanmak istiyorsanız, **PROVIDERVERSION**=7, 7.0, 7.0.0 ya da 7.0.0.0değerini ayarlayabilirsiniz.

8 -Normal kip

JMS uygulaması, IBM MQ ileti alışverişi sağlayıcısının normal kipini kullanır. Normal kip, JMS' i gerçekleştirmek için IBM MQ kuyruk yöneticisinin tüm özelliklerini kullanır. Bu kip, JMS 2.0 API ve işlevselliğini kullanmak için eniyilenmiştir.

800 ya da sonraki bir komut düzeyiyle bir kuyruk yöneticisine bağlanıyorsanız, zamanuyumsuz gönderme, gecikmeli teslim ya da paylaşılan abonelik gibi tüm JMS 2.0 API ve özellikleri kullanılabilir.

Bağlantı üreticisi ayarlarında belirtilen kuyruk yöneticisi bir IBM MQ 8.0.0 ya da sonraki bir kuyruk yöneticisi değilse, `createConnection` yöntemi bir kural dışı durumla başarısız olur JMSFMQ0003.

IBM MQ ileti alışverişi sağlayıcısı normal kipi, paylaşım etkileşimleri özelliğini kullanır ve paylaşılabilen etkileşimlerin sayısı, sunucu bağlantısı kanalındaki **SHARECNV()** özelliği tarafından denetlenir. Bu özellik 0olarak ayarlanırsa, IBM MQ ileti alışverişi sağlayıcısı normal kipini kullanamazsınız ve `createConnection` yöntemi bir kural dışı durumla başarısız olur JM5CC5007.

7 -Kısıtlamalarla olağan kip

JMS uygulaması, IBM MQ ileti alışverişi sağlayıcısının normal kipini kısıtlamalarla kullanır. Bu kip, paylaşılan abonelikler, gecikmeli teslim ya da zamanuyumsuz gönderme gibi yeni özellikleri değil, JMS 2.0 API 'sini kullanır.

PROVIDERVERSION ayarını 7 olarak ayarlarsanız, yalnızca IBM MQ ileti alışverişi sağlayıcısı olağan ve kısıtlamalar kipiyle kurun " bu seçenek kullanılabilir.

Normal kipi kullanarak sınırlamalarla bağlanıyorsanız, 800 'den düşük bir komut düzeyine sahip bir kuyruk yöneticisine bağlandıysanız, JMS 2.0 API 'sini kullanabilirsiniz, ancak zamanuyumsuz gönderme, gecikmeli teslim ya da paylaşılan abonelik özelliklerini kullanamazsınız.

IBM MQ ileti alışverişi sağlayıcısı normal kipi, kısıtlamalar ile paylaşım etkileşimleri özelliğini kullanır ve paylaşılabilen etkileşimlerin sayısı, sunucu bağlantısı kanalındaki **SHARECNV()** özelliği tarafından denetlenir. Bu özellik 0olarak ayarlanırsa, kısıtlamalarla IBM MQ ileti alışverişi sağlayıcısı normal kipini kullanamazsınız ve `createConnection` yöntemi bir kural dışı durumla başarısız olur JM5CC5007.

6 -Geçiş kipi

JMS uygulaması IBM MQ ileti alışverişi sağlayıcısı geçiş kipini kullanır.

Bu kipi kullanarak bir IBM MQ 8.0ya da daha sonraki bir kuyruk yöneticisine bağlanabilirsiniz, ancak IBM MQ classes for JMS kuyruk yöneticisinin yeni özelliklerinin hiçbiri kullanılmaz; örneğin, önden okuma ya da akış.

IBM MQ 8.0 ya da sonraki bir kuyruk yöneticisine bağlanan IBM MQ 8.0 ya da sonraki bir istemciniz varsa, ileti seçimi istemci sisteminde değil, kuyruk yöneticisi tarafından yapılır.

IBM MQ ileti alışverişi sağlayıcısı geçiş kipi belirtilirse ve herhangi-gereken sa: geçerli seni JMS 2.0 API, API yöntemi çağırısı JM5CC5007kural dışı durumuyla başarısız olur.

belirlenmedi (varsayılan)

PROVIDERVERSION özelliği varsayılan olarak *belirlenmemiş* olarak ayarlanır.

JNDI içinde önceki bir IBM MQ classes for JMS sürümüyle oluşturulan bir bağlantı üreticisi, bağlantı üreticisi IBM MQ classes for JMS' nin yeni sürümüyle kullanıldığında bu değeri alır. Hangi işlem kipinin kullanıldığını saptamak için aşağıdaki algoritma kullanılır. Bu algoritma, `createConnection` yöntemi çağrıldığında kullanılır ve IBM MQ ileti alışverişi sağlayıcısı normal kipinin, kısıtlamalarla normal kipinin ya da IBM MQ ileti alışverişi sağlayıcısı geçiş kipinin gerekli olup olmadığını saptamak için bağlantı üreticisinin diğer yönlerini kullanır.

1. Önce, IBM MQ ileti alışverişi sağlayıcısının normal kipini kullanma girişiminde bulunuldu.
2. Bağlı kuyruk yöneticisi IBM MQ 8.0 ya da üstü değilse, IBM MQ ileti alışverişi sağlayıcısını normal kipte kullanma girişiminde bulunma kısıtlamaları vardır.

3. Bağlı kuyruk yöneticisi IBM WebSphere MQ 7.0.1'ya da üstü değilse, bağlantı kapanır ve bunun yerine IBM MQ ileti alışverişi sağlayıcısı geçiş kipi kullanılır.
4. Sunucu bağlantı kanalındaki **SHARECNV** özelliği 0 olarak ayarlanırsa, bağlantı kapanır ve bunun yerine IBM MQ ileti alışverişi sağlayıcısı geçiş kipi kullanılır.
5. **BROKERVER** V1 olarak ya da varsayılan *belirlenmemiş* değere ayarlanırsa, IBM MQ ileti alışverişi sağlayıcısı normal kipi kullanılmaya devam eder.

Uyumluluğu ilgili ek bilgi için ALTER QMGR komutunun PSMODE parametresine ilişkin bilgi için [ALTER QMGR](#) başlıklı konuya bakın.

6. **BROKERVER** V2 olarak ayarlanırsa, yapılan işlem **BROKERQMGR** değerine bağlıdır:

- **BROKERQMGR** boşluk ise:

BROKERCONQ özelliği tarafından belirtilen kuyruk çıkış için açılabilir (çıkış başarılı olduğunda MQOPEN) ve kuyruk yöneticisindeki **PSMODE** COMPAT ya da DISABLED olarak ayarlandıysa, IBM MQ ileti alışverişi sağlayıcısı geçiş kipi kullanılır.

- **BROKERCONQ** özelliği tarafından belirtilen kuyruk çıkış için açılmıyorsa ya da **PSMODE** özneliği **ENABLED**: olarak ayarlandıysa:

IBM MQ ileti alışverişi sağlayıcısı normal kipi kullanılır.

- **BROKERQMGR** boşluk değilse :

IBM MQ ileti alışverişi sağlayıcısı geçiş kipi kullanılır.

Kullandığınız bağlantı üreticisini değiştiremezseniz, bağlantı üreticisinin herhangi bir ayarını geçersiz kılmak için `com.ibm.msg.client.wmq.overrideProviderVersion` özelliğini kullanabilirsiniz. Bu geçersiz kılma, JVM ' deki tüm bağlantı üreticileri için geçerlidir, ancak gerçek bağlantı üreticisi nesnelere değiştirilmez.

İlgili görevler

JMS **PROVIDERVERSION** özelliğinin yapılandırılması

PROXYHOSTNAME

Yetkili sunucu aracılığıyla bir aracıya gerçek zamanlı bağlantı kullanılırken yetkili sunucunun çalıştığı sistemin anasistem adı ya da IP adresi.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory Üreticisi

JMS yönetim aracı uzun adı: PROXYHOSTNAME

JMS yönetim aracı kısa adı: PHOST

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setProxyHostName()
- MQConnectionFactory.getProxyHostName()

Değerler

boş değerli

Yetkili sunucunun anasistem adı. Bu varsayılan değerdir.

PROXYPORT

Yetkili sunucu aracılığıyla bir aracıya gerçek zamanlı bağlantı kullanılırken yetkili sunucunun dinlediği kapının numarası.

Uygulanabilir Nesneler

ConnectionFactory, TopicConnectionFactory

JMS yönetim aracı uzun adı: PROXYPORT

JMS yönetim aracı kısa adı: PPORT

Programlı erişim

Ayarlayıcılar/alıcılar

MQConnectionFactory.setProxyPort ()

MQConnectionFactory.getProxyPort ()

Değerler

443

Yetkili sunucunun kapı numarası. Bu varsayılan değerdir.

GENEL

IBM MQ classes for JMS aracından bir alındı bildirimini istemeden önce bir yayınlayıcı tarafından yayınlanan iletilerin sayısı.

Bu özelliğin değerini düşürdüğünüzde, IBM MQ classes for JMS onayları daha sık ister, bu nedenle yayınlayıcının performansı düşer. Değeri yükselttiğinizde, IBM MQ classes for JMS aracının başarısız olması durumunda kural dışı durum yayınlamak daha uzun sürer. Bu özellik yalnızca TRANSPORT, BIND ya da CLIENT değerine sahiptir.

Uygulanabilir Nesneler

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: PROXYPORT

JMS yönetim aracı kısa adı: PPORT

Programlı erişim

Ayarlayıcılar/alıcılar

MQConnectionFactory.setPubAckInterval()

MQConnectionFactory.getPubAckInterval()

Değerler

25

Herhangi bir pozitif tamsayı varsayılan değer olabilir.

PUTASYNCKUTSAL

Bu özellik, ileti üreticilerinin bu hedefe ileti göndermek için zamanuyumsuz girişler kullanmalarına izin verilip verilmediğini belirler.

Uygulanabilir Nesneler

Kuyruk, Konu

JMS yönetim aracı uzun adı: PUTASYNCKAL,

JMS yönetim aracı kısa adı: PAALD

Programlı erişim

Ayarlayıcılar/alıcılar

MQDestination.setPutAsyncAllowed()

MQDestination.getPutAsyncAllowed()

Değerler

AS_DEST

Kuyruğa ya da konu tanımlamasına gönderme yaparak, zamanuyumsuz girişlere izin verilip verilmediğini saptayın. Bu varsayılan değerdir.

AS_Q_DEF

Kuyruk tanımlamasına gönderme yaparak, zamanuyumsuz girişlere izin verilip verilmediğini saptayın.

AS_TOPIC_DEF

Konu tanımlamasına başvurarak zamanuyumsuz girişlere izin verilip verilmediğini belirleyin.

HAYIR

Zamanuyumsuz girişlere izin verilmez.

EVET

Zamanuyumsuz girişlere izin verilir.

YÖNETİCİ

Bağlanılacak kuyruk yöneticisinin adı.

Ancak, uygulamanız bir kuyruk yöneticisine bağlanmak için bir istemci kanal tanımlama çizelgesi kullanıyorsa, [IBM MQ classes for JMS ile istemci kanal tanımlama çizelgesinin kullanılmasında](#) başlıklı konuya bakın.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, Kuyruk, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: QMANAGER

JMS yönetim aracı kısa adı: QMGR

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setQueueManager ()
- MQConnectionFactory.getQueueManager ()

Değerler

"" (boş dizgi)

Herhangi bir dizgi varsayılan değer olabilir.

kuyruk

JMS kuyruk hedefinin adı. Bu, kuyruk yöneticisi tarafından kullanılan kuyruğun adıyla eşleşir.

Uygulanabilir Nesnelere

Kuyruk

JMS yönetim aracı uzun adı: QUEUE

JMS yönetim aracı kısa adı: QU

Değerler

Herhangi bir dizgi

Geçerli bir IBM MQ kuyruk adı.

İlgili başvurular

[IBM MQ nesnelere adlandırma kuralları >](#)

READAHEADALLOWED

Bu özellik, ileti tüketicilerinin ve kuyruk tarayıcılarının, bu hedeften kalıcı olmayan iletileri almadan önce iç arabelleğe almak için önden okuma özelliğini kullanıp kullanamayacağını belirler.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS yönetim aracı uzun adı: READAHEADALLOWED

JMS yönetim aracı kısa adı: RAALD

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setReadAheadAllowed()
- MQDestination.getReadAheadAllowed()

Değerler

AS_DEST

Kuyruğa ya da konu tanımlamasına başvurarak önden okumaya izin verilip verilmediğini belirleyin. Bu, yönetim araçlarındaki varsayılan değerdir.

Programlarda WMQConstants.WMQ_READ_AHEAD_ALLOWED_AS_DEST komutunu kullanın.

AS_Q_DEF

Kuyruk tanımlamasına başvurarak önden okumaya izin verilip verilmediğini belirleyin.

Programlarda WMQConstants.WMQ_READ_AHEAD_ALLOWED_AS_Q_DEF komutunu kullanın.

AS_TOPIC_DEF

Konu tanımlamasına başvurarak önden okumaya izin verilip verilmediğini belirleyin.

Programlarda WMQConstants.WMQ_READ_AHEAD_ALLOWED_AS_TOPIC_DEF komutunu kullanın.

HAYIR

Önden okumaya izin verilmez.

Programlarda WMQConstants.WMQ_READ_AHEAD_ALLOWED_DISABLED komutunu kullanın.

EVET

Önden okumaya izin verilir.

Programlarda WMQConstants.WMQ_READ_AHEAD_ALLOWED_ENABLED komutunu kullanın.

READAHEADCLOSEPOLICY

Zamanuyumsuz bir ileti dinleyicisine teslim edilen iletiler için, ileti tüketicisi kapatıldığında iç önden okuma arabelleğindeki iletilere ne olur?

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS yönetim aracı uzun adı: READAHEADCLOSEPOLICY

JMS yönetim aracı kısa adı: RACP

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setReadAheadClosePolicy()
- MQDestination.getReadAheadClosePolicy()

Değerler

Tümünü KALDIR

İç önden okuma arabelleğindeki tüm iletiler, geri dönmeden önce uygulamanın ileti dinleyicisine teslim edilir. Bu, yönetim araçlarındaki varsayılan değerdir.

Programlarda WMQConstants.WMQ_READ_AHEAD_DELIVERALL komutunu kullanın.

TESLIVER_CURRENT

Yalnızca yürürlükteki ileti dinleyici çağrısı dönmeden önce tamamlanır; iletiler iç önden okuma arabelleğinde bırakılır ve daha sonra atılır.

Programlarda WMQConstants.WMQ_READ_AHEAD_DELIVERCURRENT komutunu kullanın.

RECEIVECCSID

Kuyruk yöneticisi ileti dönüştürmesi için hedef CCSID ' yi belirleyen hedef özellik. RECEIVECONVERSION WMQ_RECEIVE_CONVERSION_QMGR olarak ayarlanmadıkça değer yoksayılr.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS denetim aracı uzun adı: RECEIVECCSID

JMS yönetim aracı kısa adı: RCCS

Programlı erişim

Ayarlayıcılar/Alıcılar

- MQDestination.setReceiveCCSID
- MQDestination.getReceiveCCSID

Değerler

WMQConstants.WMQ_RECEIVE_CCSID_JVM_DEFAULT

0 -JVM kullan Charset.defaultCharset

1208

UTF-8

CCSID

Desteklenen kodlanmış karakter takımı tanıtıcısı.

ALAVECONVERSION

Veri dönüştürme işleminin kuyruk yöneticisi tarafından gerçekleştirilip gerçekleştirilmeyeceğini belirleyen hedef özellik.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS yönetim aracı uzun adı: RECEIVECONVERSION

JMS yönetim aracı kısa adı: RCNV

Programlı erişim

Ayarlayıcılar/Alıcılar

- `MQDestination.setReceiveConversion`
- `MQDestination.getReceiveConversion`

Değerler

WMQConstants.WMQ_RECEIVE_CONVERSION_CLIENT_MSG

1 -Yalnızca JMS istemcisinde veri dönüştürme işlemi gerçekleştirin. 7.0.1.5dahil olmak üzere, V7.0' a kadar olan varsayılan değer.

WMQConstants.WMQ_RECEIVE_CONVERSION_QMGR

2 -İstemciye ileti göndermeden önce kuyruk yöneticisinde veri dönüştürme işlemi gerçekleştirin. APAR IC72897 uygulanmazsa, V7.0 ile V7.0.1.4 (bu değerler de içinde olmak üzere) arasındaki varsayılan (ve yalnızca) değer.

ALMA YALITMA

Bu özellik, bir abonenin abone kuyruğunda kesinleştirilmemiş iletileri alıp alamayacağını belirler.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionÜreticisi

JMS yönetim aracı uzun adı: RECEIVEISOLATION

JMS yönetim aracı kısa adı: RCVISOL

Değerler

Kesinleştirildi

Bir abone yalnızca, abone kuyruğundaki kesinleştirilmiş iletileri alır. Bu, yönetim araçlarındaki varsayılan değerdir.

Programlarda `WMQConstants.WMQ_RCVISOL_COMMITTED` komutunu kullanın.

ONAYLANMADI

Bir abone, abone kuyruğunda kesinleştirilmemiş iletileri alabilir.

Programlarda `WMQConstants.WMQ_RCVISOL_UNCOMMITTED` komutunu kullanın.

YENİDEN ÇIK

Arka arkaya çalıştırılacak bir kanal alma çıkışı ya da alma çıkışlarını tanımlar.

IBM MQ classes for JMS ' in alma çıkışlarını bulması için ek yapılandırma gerekebilir. Daha fazla bilgi için [JMS için IBM MQ sınıflarının kanal çıkışlarını kullanacak şekilde yapılandırılması](#) başlıklı konuya bakın.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: RECEXIT

JMS yönetim aracı kısa adı: RCX

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setReceiveExit ()
- MQConnectionFactory.getReceiveExit ()

Değerler

- null (boş değerli). Bu varsayılan değerdir.
- Virgüllerle ayrılmış bir ya da daha çok öğeden oluşan bir dizgi; burada her öğe aşağıdakilerden biridir:
 - WMQReceiveExit arabirimini uygulayan bir sınıfın adı (Java içinde yazılmış bir kanal alma çıkışı için).
 - *libraryName(entryPointName)* biçimindeki bir dizgi (Java içinde yazılmayan bir kanal alma çıkışı için).

RECEXITINIT

Kanala geçirilen kullanıcı verileri, çağrıldıklarında çıkış alır.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: RECEXITINIT

JMS yönetim aracı kısa adı: RCXI

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setReceiveExitInit()
- MQConnectionFactory.getReceiveExitInit()

Değerler

boş değerli

Virgüllerle ayrılmış bir ya da daha çok kullanıcı verisi ögesini içeren bir dizgi. Bu varsayılan değerdir.

REPLYTOSTYLE

Alınan bir iletideki JMSReplyTo alanının nasıl oluşturulacağını belirler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: REPLYTOSTYLE

JMS yönetim aracı kısa adı: RTOST

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setReplyToStyle()
- MQConnectionFactory.getReplyToStyle()

Değerler

VARSAYILAN

MQMD ile eşdeğerdir.

RFH2

RFH2 üstbilgisinde sağlanan değeri kullanın. Gönderen uygulamada bir JMSReplyTo değeri ayarlandıysa, bu değeri kullanın.

Mqmd

MQMD tarafından sağlanan değeri kullanın. Bu davranış, IBM WebSphere MQ 6.0.2.4 ve 6.0.2.5 varsayılan davranışına eşdeğerdir.

Gönderen uygulama tarafından ayarlanan JMSReplyTo değeri bir kuyruk yöneticisi adı içermiyorsa, alan kuyruk yöneticisi MQMD ' ye kendi adını ekler. Bu parametreyi MQMD olarak ayarlarsanız, kullandığınız yanıt kuyruğu alıcı kuyruk yöneticisinde olur. Bu parametreyi RFH2 olarak ayarlarsanız, kullandığınız yanıt kuyruğu, gönderen uygulama tarafından ilk olarak ayarlandığı şekilde, gönderilen iletinin RFH2 ' de belirtilen kuyruk yöneticisinde bulunur.

Gönderen uygulama tarafından ayarlanan JMSReplyTo değeri bir kuyruk yöneticisi adı içeriyorsa, hem MQMD hem de RFH2 aynı değeri içerdiğinden bu parametrenin değeri önemli değildir.

İLK DURUMUNA GEÇİR

Noktadan noktaya iletişim etki alanındaki bir ileti tüketicisi, hangi iletileri almak istediğini seçmek için bir ileti seçici kullandığında, IBM MQ classes for JMS kuyrukta IBM MQ kuyruğunda kuyruğun MsgDeliverySequence özneliği tarafından belirlenen sırada uygun iletileri arar.

IBM MQ classes for JMS uygun bir ileti bulup tüketicie teslim ettikten sonra IBM MQ classes for JMS , kuyruktaki geçerli konumundan sonraki uygun iletiyi aramaya devam eder. IBM MQ classes for JMS , kuyruğun sonuna ulaşınca kadar ya da bu özelliğin değerinin belirlendiği süre (milisaniye) doluncaya kadar kuyrukta bu şekilde arama yapmaya devam eder. Her durumda IBM MQ classes for JMS , aramaya devam etmek için kuyruğun başına döner ve yeni bir zaman aralığı başlar.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory

JMS yönetim aracı uzun adı: RESCANINT

JMS yönetim aracı kısa adı: RINT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setRescanInterval ()
- MQConnectionFactory.getRescanInterval ()

Değerler

5000

Herhangi bir pozitif tamsayı varsayılan değer olabilir.

SECEXIT

Bir kanal güvenlik çıkışı tanımlar.

IBM MQ classes for JMS ' in güvenlik çıkışlarını bulması için ek yapılandırma gerekebilir. Daha fazla bilgi için JMS için IBM MQ sınıflarının kanal çıkışlarını kullanacak şekilde yapılandırılması başlıklı konuya bakın.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SECEXIT

JMS yönetim aracı kısa adı: SXC

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSecurityExit ()
- MQConnectionFactory.getSecurityExit ()

Değerler

- null (boş değerli). Bu varsayılan değerdir.
- Virgüllerle ayrılmış bir ya da daha çok öğeden oluşan bir dizgi; burada her öğe aşağıdakilerden biridir:
 - WMQSecurityExit arabirimini uygulayan bir sınıfın adı (Java içinde yazılmış bir kanal güvenliği çıkışı için).
 - *libraryName(entryPointName)* biçimindeki bir dizgi (Java içinde yazılmayan bir kanal güvenlik çıkışı için).

SECEXITINIT

Çağrıldığında bir kanal güvenlik çıkışına geçirilen kullanıcı verileri.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SECEXITINIT

JMS yönetim aracı kısa adı: SCXI

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSecurityExitInit()
- MQConnectionFactory.getSecurityExitInit()

Değerler

boş değerli

Herhangi bir dizgi varsayılan değer olabilir.

SENDCHECKCOUNT

Hareket etmeyen tek bir JMS oturumunda zamanuysuz koyma hatalarının denetlenmesi arasında izin verilecek gönderme çağrılarının sayısı.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SENDCHECKCOUNT

JMS yönetim aracı kısa adı: SCC

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSendCheckCount()
- MQConnectionFactory.getSendCheckCount()

Değerler

boş değerli

Herhangi bir dizgi varsayılan değer olabilir.

SENDEXIT

Bir kanal gönderme çıkışı ya da art arda çalıştırılacak gönderme çıkışlarını tanımlar.

IBM MQ classes for JMS ' in gönderme çıkışlarını bulması için ek yapılandırma gerekebilir. Daha fazla bilgi için [JMS için IBM MQ sınıflarının kanal çıkışlarını kullanacak şekilde yapılandırılması](#) başlıklı konuya bakın.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SENDEXIT

JMS yönetim aracı kısa adı: SDX

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSendExit ()
- MQConnectionFactory.getSendExit ()

Değerler

- null (boş değerli). Bu varsayılan değerdir.
- Virgüllerle ayrılmış bir ya da daha çok öğeden oluşan bir dizgi; burada her öğe aşağıdakilerden biridir:
 - WMQSendExit arabirimini uygulayan bir sınıfın adı (Java içinde yazılmış bir kanal gönderme çıkışı için).
 - *libraryName(entryPointName)* biçimindeki bir dizgi (Java içinde yazılmayan bir kanal gönderme çıkışı için).

SENDEXITINIT

Çağrıldıklarında kanal gönderme çıkışlarına geçirilen kullanıcı verileri.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SENDEXITINIT

JMS yönetim aracı kısa adı: SDXI

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSendExitInit()
- MQConnectionFactory.getSendExitInit()

Değerler

boş değerli

Virgüllerle ayrılmış kullanıcı verilerinin bir ya da daha çok ögesini içeren dizgiler varsayılan değer olabilir.

FARKLILIKLI

IBM MQ ileti alışverişi sağlayıcısı normal kipini ya da sınırlamalarla olağan kipini kullanan uygulamalar için bu özellik, bağlantı üreticisinden yaratılan JMS bağlantıları, oturumları ve bağlamları için paylaşım etkileşimleri işlevinin kullanılıp kullanılmayacağını belirler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SHARECONVAL,

JMS yönetim aracı kısa adı: SCALD

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setShareConvAllowed()
- MQConnectionFactory.getShareConvAllowed()

Değerler

EVET

Aynı JVM içindeki bağlantı üreticisinden yaratılan JMS bağlantıları, oturumları ve bağlamları, uygun olduğunda bir kanal eşgörünümünü (TCP/IP bağlantısıyla eşlenir) paylaşabilir.

Bu, yönetim araçları için varsayılan değerdir.

Programlar için WMQConstants.WMQ_SHARE_CONV_ALLOWED_YES komutunu kullanın.

HAYIR

Bağlantı üreticisinden oluşturulan her JMS bağlantısı ve bu JMS bağlantılarından yaratılan her JMS oturumu, bir kuyruk yöneticisine kendi kanal yönetim ortamına (TCP/IP bağlantısı) sahiptir.

JMS bağlamları için, bağlantı üreticisinden yaratılan ilk bağlam iki kanal yönetim ortamı (TCP/IP bağlantıları) yaratır. İlkinden yaratılan diğer JMS bağlamlarının kendi kanal yönetim ortamı (TCP/IP bağlantısı) vardır.

Programlar için WMQConstants.WMQ_SHARE_CONV_ALLOWED_NO değerini kullanın.

İlgili kavramlar

IBM MQ ileti alışverişi sağlayıcısı işletim kipleri

JMS için IBM MQ sınıflarında TCP/IP bağlantısının paylaşılması

SPARSESUB

TopicSubscriber nesnesinin ileti alma ilkesini denetler.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory

JMS yönetim aracı uzun adı: SPARSESUB

JMS yönetim aracı kısa adı: SSUBS

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSparseAbonelikler ()
- MQConnectionFactory.getSparseAbonelikler ()

Değerler

HAYIR

Abonelikler sık sık eşleşen iletiler alır. Bu, yönetim araçları için varsayılan değerdir.

Programlar için false değerini kullanın.

EVET

Abonelikler, seyrek eşleşen iletiler alır. Bu değer, abonelik kuyruğunun göz atmak için açılabilmesini gerektirir.

Programlar için true değerini kullanın.

ŞIFRELEME TAKIMI

TLS bağlantısı için kullanılacak CipherSuite .

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SSLCIPHERSUITE

JMS yönetim aracı kısa adı: SCPHS

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSSLCipherSuite ()
- MQConnectionFactory.getSSLCipherSuite ()

Değerler

boş değerli

Bu varsayılan değerdir. Daha fazla bilgi için bkz. [JMS nesnelere TLS özellikleri](#).

SSLCRL

TLS sertifika iptali olup olmadığını denetlemek için CRL sunucuları.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SSLCRL

JMS yönetim aracı kısa adı: SCRL

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSSLCertStores ()
- MQConnectionFactory.getSSLCertStores ()

Değerler

boş değerli

LDAP URL adreslerinin boşlukla ayrılmış listesi. Bu varsayılan değerdir. Daha fazla bilgi için bkz. [JMS nesnelerinin TLS özellikleri](#).

SSLFIPSREQUIRED

Bu özellik, bir TLS bağlantısının IBM Java JSSE FIPS sağlayıcısı (IBMJSSEFIPS) tarafından desteklenen bir CipherSuite kullanması gerekip gerekmediğini belirler.

Not: AIX, Linux, and Windows işletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, [IBM Crypto for C \(ICC\) sertifikasını](#) görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu [İşlem listesindeki NIST CMVP modüllerinde](#) aranarak görüntülenebilir.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SSLFIPSREQUIRED

JMS yönetim aracı kısa adı: SFIPS

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSSLFipsRequired ()
- MQConnectionFactory.getSSLFipsGerekli ()

Değerler

HAYIR

TLS bağlantısı, IBM Java JSSE FIPS sağlayıcısı (IBMJSSEFIPS) tarafından desteklenmeyen herhangi bir CipherSuite kullanabilir.

Bu varsayılan değerdir. Programlarda false değerini kullanın.

EVET

TLS bağlantısı, IBMJSSEFIPS tarafından desteklenen bir CipherSuite kullanmalıdır.

Programlarda true değerini kullanın.

SSLPEERNAME

TLS için, kuyruk yöneticisi tarafından sağlananlarla eşleşmesi gereken bir *ayırt edici ad* iskeleti.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SSLPEERNAME

JMS yönetim aracı kısa adı: SPEER

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSSLPeerName ()
- MQConnectionFactory.getSSLPeerAd ()

Değerler

boş değerli

Bu varsayılan değerdir. Daha fazla bilgi için bkz. [JMS nesnelere TLS özellikleri](#).

SSLRESETCOUNT

TLS için, şifreleme için kullanılan gizli anahtar yeniden anlaşılmadan önce bir bağlantı tarafından gönderilen ve alınan toplam bayt sayısı.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SSLRESETCOUNT

JMS yönetim aracı kısa adı: SRC

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSSLResetCount ()
- MQConnectionFactory.getSSLResetCount ()

Değerler

0

Sıfır ya da 999, 999 ya da 999 'dan küçük ya da bu değere eşit herhangi bir pozitif tamsayı. Bu varsayılan değerdir. Daha fazla bilgi için bkz. [JMS nesnelere TLS özellikleri](#).

STATREFRESHINT

Bir abonenin kuyruk yöneticisine olan bağlantısını kaybettiği zaman algılanan, uzun süren hareketin yenilenmesi arasındaki milisaniye cinsinden aralık.

Bu özellik yalnızca SUBSTORE ' un QUEUE değeri varsa anlamlıdır.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: STATREFRESHINT

JMS yönetim aracı kısa adı: SRI

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setStatusRefreshInterval()
- MQConnectionFactory.getStatusRefreshInterval()

Değerler

60000

Herhangi bir pozitif tamsayı varsayılan değer olabilir. Daha fazla bilgi için bkz. [JMS nesnelerinin TLS özellikleri](#).

ALT MAĞAZA

Burada IBM MQ classes for JMS , etkin aboneliklerle ilgili kalıcı verileri saklar.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SUBSTORE

JMS yönetim aracı kısa adı: SS

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSubscriptionStore ()
- MQConnectionFactory.getSubscriptionStore ()

Değerler

Aracı

Aboneliklerin ayrıntılarını tutmak için aracıya dayalı abonelik deposunu kullanın. Bu, yönetim araçları için varsayılan değerdir.

Programlar için WMQConstants.WMQ_SUBSTORE_BROKERkomutunu kullanın.

GEÇİŞ YAPIN

Kuyruk tabanlı abonelik deposundan aracı tabanlı abonelik deposuna abonelik bilgilerini aktarın.

Programlar için WMQConstants.WMQ_SUBSTORE_MIGRATEkomutunu kullanın.

kuyruk

Aboneliklerin ayrıntılarını tutmak için kuyruğa dayalı abonelik deposunu kullanın.

Programlar için WMQConstants.WMQ_SUBSTORE_QUEUEkomutunu kullanın.

SYNCPPOINTALLYAKALAR

Bu özellik, tüm alımların eşitleme noktası altında gerçekleştirilip gerçekleştirilmeyeceğini belirler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: SYNCPPOINTALLTALK

JMS yönetim aracı kısa adı: SPAG

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setSyncpointAllGets()
- MQConnectionFactory.getSyncpointAllGets()

Değerler

Hayır

Bu varsayılan değerdir.

Evet

HEDEF

Bu özellik, hedef uygulamalarla bilgi alışverişi için IBM MQ RFH2 biçiminin kullanılıp kullanılmayacağını belirler.

Uygulanabilir Nesnelere

Kuyruk, Konu

JMS yönetim aracı uzun adı: TARGCLIENT

JMS yönetim aracı kısa adı: TC

Programlı erişim

Ayarlayıcılar/alıcılar

- MQDestination.setTargetClient()
- MQDestination.getTargetClient()

Değerler

JMS

İletinin hedefi bir JMS uygulamasıdır. Bu, yönetim araçları için varsayılan değerdir.

Programlar için WMQConstants.WMQ_CLIENT_JMS_COMPLIANT komutunu kullanın.

MQ

İletinin hedefi JMS IBM MQ olmayan bir uygulamadır.

Programlar için WMQConstants.WMQ_CLIENT_NONJMS_MQ komutunu kullanın.

TARGCLIENTEŞLEŞME

Bu özellik, gelen bir iletinin JMSReplyTo üstbilgi alanıyla tanımlanan kuyruğa gönderilen bir yanıt iletisinin yalnızca gelen iletinin bir MQRFH2 üstbilgisi varsa MQRFH2 üstbilgisine sahip olup olmadığını belirler.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory

JMS yönetim aracı uzun adı: TARGCLIENTEŞLEŞME

JMS yönetim aracı kısa adı: TCM

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setTargetClientMatching()

- MQConnectionFactory.getTargetClientMatching()

Değerler

EVET

Gelen bir iletinin MQRFH2 üstbilgisi yoksa, iletinin JMSReplyTo üstbilgi alanından türetilen Kuyruk nesnesinin TARGCLIENT özelliği MQ' ya gönderilir. İletinin MQRFH2 üstbilgisi varsa, TARGCLIENT özelliği JMS olarak ayarlanır. Bu, yönetim araçları için varsayılan değerdir.

Programlar için true değerini kullanın.

HAYIR

Gelen bir iletinin JMSReplyTo üstbilgi alanından türetilen Kuyruk nesnesinin TARGCLIENT özelliği her zaman JMSolarak ayarlanır.

Programlar için false değerini kullanın.

TEMPMODEL

JMS geçici kuyruklarının yaratıldığı model kuyruğunun adı.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory

JMS yönetim aracı uzun adı: TEMPMODEL

JMS yönetim aracı kısa adı: TM

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setTemporaryModel ()
- MQConnectionFactory.getTemporaryModel ()

Değerler

SYSTEM.DEFAULT.MODEL.QUEUE

Herhangi bir dizgi varsayılan değer olabilir.

TEMPQPREFIX

Bir IBM MQ dinamik kuyruğunun adını oluşturmak için kullanılan önek.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory

JMS yönetim aracı uzun adı: TEMPQPREFIX

JMS yönetim aracı kısa adı: TQP

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setTempQPrefix ()
- MQConnectionFactory.getTempQPrefix ()

Değerler

" " (boş dizgi)

Kullanılan önek, diğer tüm platformlarda z/OS üzerinde CSQ . * ve AMQ . * ' dir. Bunlar varsayılan değerlerdir.

Kuyruk öneki

Kuyruk öneki, IBM MQ nesne tanımlayıcısında (yapı MQOD) *DynamicQName* alanının içeriğini oluşturma kurallarına uyan herhangi bir dizedir, ancak son boş olmayan karakter bir yıldız işareti olmalıdır.

TEMPTOPICPREFIX

Geçici konular oluştururken JMS , " TEMP /*TEMPTOPICPREFIX/unique_id* " biçiminde bir konu dizgisi oluşturur ya da bu özellik varsayılan değerle bırakılırsa, yalnızca " TEMP /*unique_id* " olur. Boş olmayan bir TEMPTOPICPREFIX değerinin belirtilmesi, bu bağlantı altında yaratılan geçici konulara aboneler için yönetilen kuyruklar yaratmak üzere belirli model kuyruklarının tanımlanmasına olanak sağlar.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: TEMPTOPICPREFIX

JMS yönetim aracı kısa adı: TTP

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setTempTopicPrefix()
- MQConnectionFactory.getTempTopicPrefix()

Değerler

Bir IBM MQ konu dizgisi için yalnızca geçerli karakterlerden oluşan boş olmayan dizgilerdir. Varsayılan değer " " (boş dizgi) ' dir.

Konu

JMS konu hedefinin adı olan bu değer, kuyruk yöneticisi tarafından bir yayının ya da aboneliğin konu dizgisi olarak kullanılır.

Uygulanabilir Nesnelere

Konu

JMS yönetim aracı uzun adı: TOPIC

JMS yönetim aracı kısa adı: TOP

Değerler

Herhangi bir dizgi

Geçerli bir IBM MQ konu dizgisi oluşturan bir dize. IBM MQ ' i WebSphere Application Server ile ileti alışverişi sağlayıcısı olarak kullanırken, WebSphere Application Server içinde konunun yönetim amaçlarıyla tanıdığı adla eşleşen bir değer belirtin.

İlgili başvurular

[Konu dizgileri](#)

Aktarım

Bir kuyruk yöneticisine ya da aracıya yönelik bağlantının türü.

Uygulanabilir Nesnelere

ConnectionFactory, QueueConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: TRANSPORT

JMS yönetim aracı kısa adı: TRAN

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setTransportType ()
- MQConnectionFactory.getTransportType ()

Değerler

Bağ Tanımla

Bağ tanımlama kipinde bir kuyruk yöneticisine bağlantı için. Bu, yönetim araçları için varsayılan değerdir.

Programlar için WMQConstants.WMQ_CM_BINDINGSkomutunu kullanın.

CLIENT

İstemci kipinde bir kuyruk yöneticisine bağlantı için.

Programlar için WMQConstants.WMQ_CM_CLIENTkomutunu kullanın.

Doğrudan

HTTP tüneli kullanmayan bir aracıya gerçek zamanlı bağlantı için.

Programlar için WMQConstants.WMQ_CM_DIRECT_TCPIPkomutunu kullanın.

YÖNTEM

HTTP tüneli kullanan bir aracıya gerçek zamanlı bağlantı için. Yalnızca HTTP 1.0 desteklenir.

Programlar için WMQConstants.WMQ_CM_DIRECT_HTTPkomutunu kullanın.

İlgili kavramlar

[“IBM MQ classes for JMS nesnelere özellikleri arasındaki bağımlılıklar” sayfa 1867](#)

Bazı özelliklerin geçerliliği, diğer özelliklerin belirli değerlerine bağlıdır.

WILDCARDFORMAT

Bu özellik, genel arama karakteri sözdiziminin hangi sürümünün kullanılacağını belirler.

Uygulanabilir Nesnelere

ConnectionFactory, TopicConnectionFactory, XAConnectionFactory, XATopicConnectionFactory

JMS yönetim aracı uzun adı: WILDCARDFORMAT

JMS yönetim aracı kısa adı: WCFMT

Programlı erişim

Ayarlayıcılar/alıcılar

- MQConnectionFactory.setWildcardFormat()
- MQConnectionFactory.getWildcardFormat()

Değerler

YALNIZCA KONU

Aracı sürüm 2 'de kullanıldığı şekilde, yalnızca konu düzeyi genel arama karakterlerini tanır. Bu, yönetim araçları için varsayılan değerdir.

Programlar için WMQConstants.WMQ_WILDCARD_TOPIC_ONLYkomutunu kullanın.

YALNIZCA

Aracı sürüm 1 'de kullanıldığı şekilde, yalnızca genel arama karakterlerini tanır.

Programlar için WMQConstants.WMQ_WILDCARD_CHAR_ONLYkomutunu kullanın.

ENCODING özelliği

ENCODING özelliği, on iki olası birleşimde üç alt özellikten oluşur.

ENCODING özelliğinin alabileceği geçerli değerler üç alt özellikten oluşturulur:

tamsayı kodlaması

Normal ya da ters çevrilmiş

ondalık kodlama

Normal ya da ters çevrilmiş

kayan nokta kodlaması

IEEE normal, IEEE ters çevrilmiş ya da z/OS

ENCODING özelliği, aşağıdaki sözdizimiyle üç karakterli bir dizgi olarak ifade edilir:

```
{N|R}{N|R}{N|R|3}
```

Bu dizgide:

- N olağan durumu gösterir
- R , tersine çevrildiğini gösterir
- 3 , z/OS ögesini belirtir.
- İlk karakter *tamsayı kodlamasını* temsil eder.
- İkinci karakter *ondalık kodlamayı* temsil eder.
- Üçüncü karakter *kayan nokta kodlamasını* temsil eder

Bu, ENCODING özelliği için on iki olası değer kümesi sağlar.

Java platformu için uygun kodlama değerlerini ayarlayan NATIVEDizgisi olan ek bir değer vardır.

Aşağıdaki örneklerde ENCODINGiçin geçerli birleşimler gösterilmektedir:

```
ENCODING (NNR)  
ENCODING (NATIVE)  
ENCODING (RR3)
```

JMS nesnelere TLS özellikleri

SSLCIPHERSUITE özelliğini kullanarak TLS (Transport Layer Security; İletim Katmanı Güvenliği) şifrelemesini etkinleştirin. Daha sonra diğer birkaç özelliği kullanarak TLS şifrelemesinin özelliklerini değiştirebilirsiniz.

TRANSPORT (CLIENT) belirtilirse, SSLCIPHERSUITE özelliğini kullanarak TLS şifrelenmiş iletişimini etkinleştirebilirsiniz. Bu özelliği JSSE sağlayıcınız tarafından sağlanan geçerli bir CipherSuite değerine ayarlayın; CHANNEL özelliği tarafından belirtilen SVRCONN kanalında adı belirtilen CipherSpec ile eşleşmelidir.

Ancak, CipherSpecs (SVRCONN kanalında belirtildiği şekilde) ve CipherSuites (ConnectionFactory nesnelere belirtildiği gibi) aynı TLS şifreleme algoritmalarını temsil etmek için farklı adlandırma

şemaları kullanır. SSLCIPHERSUITE özelliğinde tanınan bir CipherSpec adı belirtilirse, JMSAdmin bir uyarı verir ve CipherSpec ögesini eşdeğer CipherSuite ile eşler. IBM MQ ve JMSAdmin tarafından tanınan CipherSpecs listesi için bkz. [IBM MQ classes for JMS içinde TLS CipherSpecs ve CipherSuites](#).

IBM Java JSSE FIPS sağlayıcısı (IBMJSSEFIPS) tarafından desteklenen bir CipherSuite kullanmak için bağlantıya gereksinim duyarsanız, bağlantı üreticisinin SSLFIPSREQUIRED özelliğini YES olarak ayarlayın. Bu özelliğin varsayılan değeri NO değeridir; bu, bir bağlantının desteklenen herhangi bir CipherSuite' i kullanabileceği anlamına gelir. SSLCIPHERSUITE ayarlanmazsa özellik yoksayılr.

SSLPEERNAME, kanal tanımlamalarında ayarlanabilen SSLPEER değiştirgesinin biçimiyle eşleşir. Virgül ya da noktalı virgülle ayrılmış öznitelik ad-değer çiftlerinin bir listesidir. Örneğin:

```
SSLPEERNAME(CN=QMGR.*, OU=IBM, OU=WEBSPPHERE)
```

Ad ve değer kümesi, *ayırt edici adı* oluşturur. Ayırt edici adlar ve bunların IBM MQ ile kullanımı hakkında daha fazla bilgi için bkz. [IBM MQ güvenliğinin sağlanması](#).

Verili örnek, bağlanma sırasında sunucu tarafından sunulan tanıtıcı sertifikayı denetler. Bağlantının başarılı olabilmesi için, sertifikanın QMGR. ile başlayan bir Ortak Adı olmalıdır, ve en az iki Kuruluş Birimi adı olmalıdır; bunlardan ilki IBM ve ikincisi WEBSPPHERE olmalıdır. Denetleme büyük ve küçük harfe duyarlı değildir.

SSLPEERNAME ayarlanmazsa, böyle bir denetim gerçekleştirilmez. SSLCIPHERSUITE ayarlanmazsa, SSLPEERNAME yoksayılr.

SSLCRL özelliği sıfır ya da daha fazla CRL (Sertifika İptal Listesi) sunucusunu belirtir. Bu özelliğin kullanılması için Java 2 v1.4 adresinde bir JVM olması gerekir. Bu, formun boşlukla ayrılmış bir giriş listesidir:

```
ldap:// hostname:[ port ]
```

isteğe bağlı olarak tek/ile takip edilir. *port* atlanırsa, varsayılan LDAP kapısı olan 389 kabul edilir. Bağlantı sırasında, sunucu tarafından sunulan TLS sertifikası, belirtilen CRL sunucularına karşı denetlenir. CRL güvenliği hakkında daha fazla bilgi için bkz. [IBM MQ güvenliğinin sağlanması](#).

SSLCRL ayarlanmazsa, böyle bir denetim gerçekleştirilmez. SSLCIPHERSUITE ayarlanmazsa, SSLCRL yoksayılr.

SSLRESETCOUNT özelliği, şifreleme için kullanılan gizli anahtar yeniden anlaşılmadan önce bir bağlantı tarafından gönderilen ve alınan toplam bayt sayısını gösterir. Gönderilen bayt sayısı, şifrelemeden önceki sayıdır ve alınan bayt sayısı, şifre çözmeden sonraki sayıdır. Bayt sayısı, IBM MQ classes for JMS tarafından gönderilen ve alınan denetim bilgilerini de içerir.

Örneğin, 4 MB veri aktıktan sonra yeniden görüşülen bir gizli anahtarla TLS etkin bir MQI kanalı üzerinden bağlantı oluşturmak için kullanılabilir bir ConnectionFactory nesnesi yapılandırmak için JMSAdmin 'e şu komutu verin:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

Varsayılan değer olan SSLRESETCOUNT değeri sıfırsa, gizli anahtar hiçbir zaman yeniden anlaşılmaz. SSLCIPHERSUITE ayarlanmazsa, SSLRESETCOUNT özelliği yoksayılr.

IBM MQ Message Service Client (XMS) for .NET başvuru

Bu başvuru kısmı, IBM MQ Message Service Client (XMS) for .NET (XMS .NET) sınıf arabirimleri ve XMStarafından tanımlanan nesne özellikleri hakkında bilgi sağlar.

.NET arabirimler

Bu bölümde, .NET sınıf arabirimleri ve bunların özellikleri ve yöntemleri açıklanmaktadır.

Aşağıdaki tablo, IBM .XMS ad alanında tanımlanan arabirimleri özetler.

<i>Çizelge 871. .NET sınıf arabirimlerinin özeti</i>	
Arabirim	Açıklama
“IBytesMessage” sayfa 1922	Bayt iletisi, gövdesi bir bayt akışını oluşturan bir iletidir.
“İBağlantı” sayfa 1931	Bağlantı nesnesi, uygulamanın bir ileti sistemi sunucusuna etkin bağlantısını gösterir.
“IConnectionFactory” sayfa 1934	Bir uygulama bağlantı yaratmak için bağlantı üreticisi kullanır.
“IConnectionMetaVerileri” sayfa 1935	Bir ConnectionMetaVeri nesnesi, bir bağlantı hakkında bilgi sağlar.
“IDestination (Hedef)” sayfa 1936	Hedef, bir uygulamanın iletileri gönderdiği ya da bir uygulamanın iletileri aldığı ya da her ikisini birden aldığı bir kaynaktır.
“ExceptionHandler” sayfa 1937	Bir uygulama, bir bağlantıyla ilgili bir sorunla ilgili zamanuyumsuz olarak bildirim almak için kural dışı durum dinleyicisini kullanır.
“IllegalStateKural Dışı Durumu” sayfa 1937	XMS , bir uygulama bir yöntemi yanlış ya da uygun olmayan bir zamanda ya da XMS istenen işlem için uygun durumda değilse bu kural dışı durumu verir.
“InitialContext” sayfa 1938	Bir uygulama, yönetilen nesnelere havuzundan alınan nesne tanımlamalarından nesne yaratmak için InitialContext nesnesini kullanır.
“InvalidClientIDException” sayfa 1940	XMS , bir uygulama bir bağlantı için istemci tanıtıcısı ayarlamaya çalışıyorsa, ancak istemci tanıtıcısı geçerli değilse ya da kullanımda ise bu kural dışı durumu verir.
“InvalidDestinationKural Dışı Durumu” sayfa 1940	XMS , bir uygulama geçerli olmayan bir hedef belirtiyorsa bu kural dışı durumu verir.
“InvalidSelectorKural Dışı Durumu” sayfa 1940	XMS , bir uygulama sözdizimi geçerli olmayan bir ileti seçici ifadesi sağlarsa bu kural dışı durumu verir.
“IMapMessage” sayfa 1941	Harita iletisi, gövdesi, her değer ilişkili bir veri tipine sahip olduğu bir ad-değer çiftleri kümesinden oluşan bir iletidir.
“IMessage” sayfa 1949	İleti nesnesi, bir uygulamanın gönderdiği ya da aldığı bir iletiyi gösterir. IMessage, IMapMessage gibi ileti sınıflarına ilişkin bir üst sınıftır.
“IMessageConsumer” sayfa 1955	Bir uygulama, hedefe gönderilen iletileri almak için bir ileti tüketicisi kullanır.
“MessageEOFException” sayfa 1957	XMS , XMS bir uygulama bayt iletisinin gövdesini okurken bir bayt ileti akımının sonuyla karşılaşarsa bu kural dışı durumu verir.
“MessageFormatKural Dışı Durumu” sayfa 1958	XMS , XMS geçerli olmayan bir biçime sahip bir iletiyle karşılaşarsa bu kural dışı durumu verir.

Çizelge 871. .NET sınıf arabirimlerinin özeti (devamı var)

Arabirim	Açıklama
“IMessageListener (temsilci)” sayfa 1958	Bir uygulama, iletileri zamanuyumsuz olarak almak için bir ileti dinleyicisini kullanır.
“MessageNotReadableException” sayfa 1958	XMS , bir uygulama yalnızca yazma işlemi olan bir iletinin gövdesini okuma girişiminde bulunursa bu kural dışı durumu verir.
“MessageNotWritableException” sayfa 1959	XMS , bir uygulama salt okunur bir iletinin gövdesine yazma girişiminde bulunursa bu kural dışı durumu verir.
“IMessageProducer” sayfa 1959	Bir uygulama, bir hedefe ileti göndermek için ileti üreticiyi kullanır.
“IObjectMessage” sayfa 1964	Nesne iletisi, gövdesi diziselleştirilmiş Java ya da .NET nesnesi olan bir iletidir.
“IPropertyContext” sayfa 1965	IPropertyContext , özellikleri alan ve ayarlanan yöntemleri içeren soyut bir üst sınıftır. Bu yöntemler diğer sınıflar tarafından edinilir.
“IQueueBrowser” sayfa 1974	Bir uygulama, iletileri kaldırmadan kuyruktaki iletilere göz atmak için bir kuyruk tarayıcısını kullanır.
“İstekte Bulunan” sayfa 1975	Bir uygulama, istek iletisi göndermek için bir istekte bulunan kullanır ve daha sonra yanıtı bekler ve alır.
“ResourceAllocationKural Dışı Durumu” sayfa 1977	XMS , XMS bir yöntemin gerektirdiği kaynakları ayıramazsa bu kural dışı durumu yayınlar.
“SecurityException” sayfa 1977	XMS , bir uygulamanın kimliğini doğrulamak için sağlanan kullanıcı kimliği ve parola reddedilirse bu kural dışı durumu verir. XMS ayrıca, bir yetki denetimi başarısız olursa ve bir yöntemin tamamlanmasını engellerse de bu kural dışı durumu verir.
“ISession (Oturum)” sayfa 1977	Oturum, ileti göndermek ve almak için tek iş parçacıklı bir bağlamdır.
“IStreamMessage” sayfa 1987	Akış iletisi, gövdesi her değerin ilişkili bir veri tipine sahip olduğu bir değer akımından oluşan bir iletidir.
“ITextMessage” sayfa 1996	Metin iletisi, gövdesi bir dizgiden oluşan bir iletidir.
“TransactionInProgressException” sayfa 1997	XMS , bir uygulama devam eden bir hareket nedeniyle geçerli olmayan bir işlem isterse bu kural dışı durumu verir.
“TransactionRolledBackException” sayfa 1997	XMS , bir uygulama yürürlükteki hareketi kesinleştirmek için Session.commit() komutunu çağırırsa, ancak daha sonra hareket geriye işlenirse bu kural dışı durumu verir.
XMSC	.NET için, XMS özellik adları ve değerleri bu sınıfta genel sabitler olarak tanımlanır. Daha fazla ayrıntı için bkz. “XMS nesnelerinin özellikleri” sayfa 2000.

Çizelge 871. .NET sınıf arabirimlerinin özeti (devamı var)

Arabirim	Açıklama
"XMSEException" sayfa 1997	XMS , .NET yöntemine yapılan bir çağrıyı işlerken bir hata saptarsa, XMS bir kural dışı durum yayınlar. Kural dışı durum, hatayla ilgili bilgileri içeren bir nesnedir. Farklı XMS kural dışı durumu tipleri var ve bir XMSEException nesnesi yalnızca bir kural dışı durum tipi. Ancak XMSEException sınıfı, diğer XMS kural dışı durum sınıflarının bir üst sınıfıdır. XMS , diğer kural dışı durum tiplerinin uygun olmadığı durumlarda bir XMSEException nesnesi verir.
"XMSFactoryFactory" sayfa 1998	Bir uygulama yönetilen nesnelere kullanmıyorsa, bağlantı üreticileri, kuyruklar ve konular yaratmak için bu sınıfı kullanın.

Her yöntemin tanımı, yöntemeye yönelik bir çağrı işlenirken bir hata saptarsa XMS ' in döndürebileceği kural dışı durum kodlarını listeler. Her kural dışı durum kodu, karşılık gelen bir kural dışı durum içeren adlandırılmış değişmeziyle gösterilir.

IBytesMessage

Bayt iletisi, gövdesi bir bayt akışını oluşturan bir iletidir.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IBytesMessage
```

.NET Özellikler

BodyLength -Gövde Uzunluğunu Al

Arabirim:

```
Int64 BodyLength
{
    get;
}
```

İletinin gövdesi salt okunur olduğunda ileti gövdesinin uzunluğunu bayt cinsinden alın.

Döndürülen değer, iletiyi okumak için imlecin bulunduğu yerden bağımsız olarak tüm gövdenin uzunluğudur.

Özel durumlar:

- XMSEException
- MessageNotReadableException

yöntemler

ReadBoolean -Okuma Boole Deęeri

Arabirim:

```
Boolean ReadBoolean();
```

Bayt ileti akışından bir Boole deęeri okuyun.

Parametreler:

Yok

Dönüşler:

Okunan Boole deęeri.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadSignedByte-Okuma Baytı

Arabirim:

```
Int16 ReadSignedByte();
```

Bayt ileti akışındaki sonraki baytı, işaretli 8 bitlik bir tamsayı olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan bayt.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadBytes -Okuma Baytları

Arabirim:

```
Int32 ReadBytes(Byte[] array);  
Int32 ReadBytes(Byte[] array, Int32 length);
```

İmlecin yürürlükteki konumundan başlayarak bayt ileti akımından bir bayt dizisi okuyun.

Parametreler:

dizi (çıkış)

Okunan bayt dizisini içerecek arabellek. Çağrıdan önce akımdan okunacak bayt sayısı arabellek uzunluğundan büyük ya da arabellek uzunluğuna eşitse, arabellek doldurulur. Ters durumda, arabellek geri kalan tüm baytlarla kısmen doldurulur.

Girişte boş deęerli gösterge belirtirseniz, yöntem byte 'ları okumadan atlar. Çağrıdan önce akımdan okunacak bayt sayısı arabellek uzunluğundan büyük ya da arabellek uzunluğuna eşitse, atlanan bayt sayısı arabelleğin uzunluğuna eşittir. Ters durumda, geri kalan tüm baytlar atlanır. İmleç, bayt ileti akımında okunacak bir sonraki konumda kalır.

uzunluk (giriş)

Arabelleğin bayt cinsinden uzunluğu

Dönüşler:

Arabelleğe okunan bayt sayısı. Arabellek kısmen doldurulursa, değer arabelleğin uzunluğundan azdır ve okunacak başka bayt kalmadığını gösterir. Çağrıdan önce akıştan okunacak bayt kalmazsa, değer XMSC_END_OF_STREAM olur.

Girişte boş değerli gösterge belirtirseniz, yöntem değer döndürmez.

Özel durumlar:

- XMSEException
- MessageNotReadableException

ReadChar -Okuma karakteri

Arabirim:

```
Char ReadChar();
```

Bayt ileti akımından sonraki 2 baytı karakter olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan karakter.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadDouble -Okuma Çift Duyarlıklı Kayan Noktalı Sayı

Arabirim:

```
Double ReadDouble();
```

Bayt ileti akışından sonraki 8 baytı çift duyarlıklı kayan noktalı sayı olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan çift duyarlıklı kayan noktalı sayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadFloat -Kayan Nokta Numarasını Oku

Arabirim:

```
Single ReadFloat();
```

Bayt ileti akışından sonraki 4 baytı kayan noktalı sayı olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan kayar noktalı sayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadInt -Okuma Tamsayı

Arabirim:

```
Int32 ReadInt();
```

Bayt ileti akımından sonraki 4 baytı 32 bitlik bir tamsayı olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan tamsayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadLong -Okuma Uzun Tamsayı

Arabirim:

```
Int64 ReadLong();
```

Bayt ileti akımından sonraki 8 baytı 64 bitlik imzalı bir tamsayı olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan uzun tamsayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadShort -Read Short Integer (Okuma Kısa Tamsayı)

Arabirim:

```
Int16 ReadShort();
```

Bayt ileti akışından sonraki 2 baytı 16 bitlik bir tamsayı olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan kısa tamsayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadByte -İmzalanmamış Baytı Oku

Arabirim:

```
Byte ReadByte();
```

Bayt ileti akımından sonraki baytı, imzalanmamış 8 bitlik bir tamsayı olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan bayt.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadUnsignedShort-Read Unsigned Short Integer (İmzalanmamış Kısa Tamsayı)

Arabirim:

```
Int32 ReadUnsignedShort();
```

Bayt ileti akımından sonraki 2 baytı, işaretli 16 bitlik bir tamsayı olarak okuyun.

Parametreler:

Yok

Dönüşler:

Okunan işaretli kısa tamsayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadUTF -UTF Dizgisini Oku

Arabirim:

```
String ReadUTF();
```

Bayt ileti akışından UTF-8 kodlu bir dizgi okuyun.

Not: ReadUTF() komutunu çağırılmadan önce, arabelleğin imlecinin byte ileti akımının başlangıcını gösterdiğini doğrulayın.

Parametreler:

Yok

Dönüşler:

Okunan dizgiyi saran bir String nesnesi.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

*Sıfırla-Sıfırla***Arabirim:**

```
void Reset();
```

İletinin gövdesini salt okunur kipe geçirin ve imleci bayt ileti akışının başına yeniden konumlandırın.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotReadableException

*WriteBoolean -Yazma Boole Değeri***Arabirim:**

```
void WriteBoolean(Boolean value);
```

Bayt ileti akımına bir Boole değeri yazın.

Parametreler:**değer (giriş)**

Yazılacak Boole değeri.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

*WriteByte -Yazma Baytı***Arabirim:**

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

Bayt ileti akımına bir bayt yazın.

Parametreler:**değer (giriş)**

Yazılacak bayt.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteBytes -Yazma Baytları

Arabirim:

```
void WriteBytes(Byte[] value);
```

Bayt ileti akımına bir bayt dizisi yazın.

Parametreler:

değer (giriş)

Yazılacak bayt dizisi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteBytes -Kısmi Bayt Dizisini Yaz

Arabirim:

```
void WriteBytes(Byte[] value, int offset, int length);
```

Belirtilen uzunlukta tanımlanan byte ileti akımına kısmi bir bayt dizisi yazın.

Parametreler:

değer (giriş)

Yazılacak bayt dizisi.

görelî konum (giriş)

Yazılacak bayt dizisinin başlangıç noktası.

uzunluk (giriş)

Yazılacak bayt sayısı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteChar -Yazma karakteri

Arabirim:

```
void WriteChar(Char value);
```

Bayt ileti akımına önce 2 bayt, yüksek sıralı bayt olarak bir karakter yazın.

Parametreler:

değer (giriş)

Yazılacak karakter.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteDouble -Write Double Precision Floating Point Number (Çift Duyarlıklı Kayan Noktalı Sayı Yaz)

Arabirim:

```
void WriteDouble(Double value);
```

Çift duyarlıklı kayan noktalı bir sayıyı uzun bir tamsayıya dönüştürün ve uzun tamsayıyı bayt ileti akımına 8 bayt, yüksek sıralı bayt olarak yazın.

Parametreler:**değer (giriş)**

Yazılacak çift duyarlıklı kayan noktalı sayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteFloat -Yazma Kayan Noktası Numarası

Arabirim:

```
void WriteFloat(Single value);
```

Kayan noktalı bir sayıyı tamsayıya dönüştürün ve tamsayıyı bayt ileti akımına 4 bayt, önce yüksek sıralı bayt olarak yazın.

Parametreler:**değer (giriş)**

Yazılacak kayan noktalı sayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteInt -Yazma Tamsayı

Arabirim:

```
void WriteInt(Int32 value);
```

Bayt ileti akımına 4 bayt, önce yüksek sıralı bayt olarak bir tamsayı yazın.

Parametreler:**değer (giriş)**

Yazılacak tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteLong -Yazma Uzun Tamsayı

Arabirim:

```
void WriteLong(Int64 value);
```

Bayt ileti akımına önce 8 bayt, yüksek sıralı bayt olarak uzun bir tamsayı yazın.

Parametreler:**değer (giriş)**

Yazılacak uzun tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteObject -Yazma Nesnesi

Arabirim:

```
void WriteObject(Object value);
```

Belirtilen nesneyi bayt ileti akımına yazın.

Parametreler:**değer (giriş)**

Yazılacak nesne; ilkel bir tipe başvuru olmalıdır.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteShort -Write Short Integer (Yazma Kısa Tamsayı)

Arabirim:

```
void WriteShort(Int16 value);
```

Bayt ileti akımına 2 bayt, önce yüksek sıralı bayt olarak kısa bir tamsayı yazın.

Parametreler:**değer (giriş)**

Yazılacak kısa tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageNotWritableException

WriteUTF -Yazma UTF Dizisi

Arabirim:

```
void WriteUTF(String value);
```

Bayt ileti akımına UTF-8kodlu bir dizgi yazın.

Parametreler:

değer (giriş)

Yazılacak dizgiyi saran bir String nesnesi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageNotWritableException

Devralınan özellikler ve yöntemler

Aşağıdaki özellikler IMessage arabiriminden edinilir:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSReteslim, JMSReplyTo, JMSTimestamp, JMSType, Özellikler

Aşağıdaki yöntemler IMessage arabiriminden edinilir:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden edinilir:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

İBağlantı

Bağlantı nesnesi, uygulamanın bir ileti sistemi sunucusuna etkin bağlantısını gösterir.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnection
```

Bir Connection nesnesinin XMS tanımlı özelliklerinin bir listesi için bkz. "Bağlantı Özellikleri" sayfa 2001.

.NET Özellikler

ClientID -İstemci Tanıtıcısını Al ve Ayarla

Arabirim:

```
String ClientID
{
```

```
    get;  
    set;  
}
```

Bağlantı için istemci tanıtıcısını alın ve ayarlayın.

İstemci tanıtıcısı, ConnectionFactory içinde yönetici tarafından önceden yapılandırılabilir ya da ClientIDayrılarak atanabilir.

Bir istemci tanıtıcısı yalnızca yayınlama/abone olma etki alanındaki sürekli abonelikleri desteklemek için kullanılır ve noktadan noktaya etki alanında yoksayılr.

Bir uygulama bağlantı için bir istemci tanıtıcısı ayarlarsa, uygulama bağlantıyı yarattıktan hemen sonra ve bağlantı üzerinde başka bir işlem gerçekleştirmeden önce bunu yapmalıdır. Uygulama bu noktadan sonra bir istemci tanıtıcısı ayarlamaya çalışırsa, çağrı IllegalStatekural dışı durumu yayınlar.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli deęil.

Özel durumlar:

- XMSException
- IllegalStateKural Dışı Durumu
- InvalidClientIDException

ExceptionListener -Özel Durum Dinleyicisini Al ve Ayarla

Arabirim:

```
ExceptionListener ExceptionListener  
{  
    get;  
    set;  
}
```

Bağlantıyla kayıtlı kural dışı durum dinleyicisini alın ve bağlantıya bir kural dışı durum dinleyicisi kaydedin.

Bağlantıyla kayıtlı bir kural dışı durum dinleyicisi yoksa, yöntem boş deęer döndürür. Bir kural dışı durum dinleyicisi bağlantıyla zaten kayıtlıysa, kural dışı durum dinleyicisi yerine boş deęer belirterek kaydı iptal edebilirsiniz.

Kural dışı durum dinleyicilerini kullanma hakkında daha fazla bilgi için bkz. [.NETiçinde ileti ve kural dışı durum dinleyicilerini kullanma](#).

Özel durumlar:

- XMSException

Meta Veri-Meta Verileri Al

Arabirim:

```
IConnectionMetaData MetaData  
{  
    get;  
}
```

Bağlantı için meta verileri alın.

Özel durumlar:

- XMSException

yöntemler

Bağlantıyı Kapat

Arabirim:

```
void Close();
```

Bağlantıyı kapatın.

Bir uygulama önceden kapatılmış bir bağlantıyı kapatmaya çalışırsa, çağrı yoksayılr.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

CreateSession - Oturum Yarat

Arabirim:

```
ISession CreateSession(Boolean transacted,  
AcknowledgeMode acknowledgeMode);
```

Bir oturum yaratın.

Parametreler:

işlemden geçildi (giriş)

True değeri, oturumun hareketinin yapıldığı anlamına gelir. False değeri, oturumun işlemden geçmediği anlamına gelir.

Bir aracıya gerçek zamanlı bağlantı için değer False olmalıdır.

acknowledgeMode (giriş)

Bir uygulama tarafından alınan iletilerin nasıl alındığını gösterir. Değer, AcknowledgeMode numaralandırıcısından aşağıdakilerden biri olmalıdır:

```
AcknowledgeMode.AutoAcknowledge  
AcknowledgeMode.ClientAcknowledge  
AcknowledgeMode.DupsOkAcknowledge
```

Bir aracıya gerçek zamanlı bağlantı için değer AcknowledgeMode.AutoAcknowledge ya da AcknowledgeMode.DupsOkAcknowledge olmalıdır

Oturum işlemden geçtiyse bu parametre yoksayılr. Alındı bildirim i kipleri hakkında daha fazla bilgi için bkz. [İleti alındı bildirim i](#).

Dönüşler:

Oturum nesnesi

Özel durumlar:

- XMSEException

Bağlantıyı Başlat

Arabirim:

```
void Start();
```

Bağlantı için gelen iletilerin teslimini başlatın ya da yeniden başlatın. Bağlantı önceden başlatılmışsa çağrı yoksayılr.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

Durdur-Bağlantıyı Durdur

Arabirim:

```
void Stop();
```

Bağlantı için gelen iletilerin teslimini durdurun. Bağlantı önceden durdurulduysa çağrı yoksayıdır.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden edinilir:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IConnectionFactory

Bir uygulama bağlantı yaratmak için bağlantı üreticisi kullanır.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnectionFactory
```

Bir ConnectionFactory nesnesinin XMS tanımlı özelliklerinin bir listesi için bkz. [“ConnectionFactory Özellikleri”](#) sayfa 2001.

yöntemler

CreateConnection -Bağlantı Üreticisi Yarat (varsayılan kullanıcı kimliğini kullanarak)

Arabirim:

```
IConnection CreateConnection();
```

Varsayılan özelliklerle bir bağlantı üreticisi oluşturun.

IBM MQ ' e bağlanıyorsanız ve XMSC_USERID ayarlanmamışsa, kuyruk yöneticisi varsayılan olarak oturum açmış kullanıcının userID değerini kullanır. Tek tek kullanıcılar için daha fazla bağlantı düzeyinde kimlik doğrulaması gerekiyorsa, IBM MQ içinde yapılandırılan bir istemci kimlik doğrulama çıkışı yazabilirsiniz.

Parametreler:

Yok

Özel durumlar:

- XMSException

CreateConnection -Bağlantı Yarat (belirtilen bir kullanıcı kimliğini kullanarak)

Arabirim:

```
IConnection CreateConnection(String userId, String password);
```

Belirtilen bir kullanıcı kimliğini kullanarak bağlantı yaratın.

IBM MQ ' e bağlanıyorsanız ve XMSC_USERID ayarlanmamışsa, kuyruk yöneticisi varsayılan olarak oturum açmış kullanıcının userID değerini kullanır. Tek tek kullanıcılar için daha fazla bağlantı düzeyinde kimlik doğrulaması gerekiyorsa, IBM MQ içinde yapılandırılan bir istemci kimlik doğrulama çıkışı yazabilirsiniz.

Bağlantı durduruldu kipinde yaratılır. Uygulama **Connection.start()** ' i arayana kadar ileti teslim edilmez.

Parametreler:

userID (giriş)

Uygulamanın kimliğini doğrulamak için kullanılacak kullanıcı kimliğini içeren bir String nesnesi. Boş değer sağlarsanız, kimlik doğrulaması olmadan bağlantı yaratma girişiminde bulunursunuz.

parola (giriş)

Uygulamanın kimliğini doğrulamak için kullanılacak parolayı içeren bir String nesnesi. Boş değer sağlarsanız, kimlik doğrulaması olmadan bağlantı yaratma girişiminde bulunursunuz.

Dönüşler:

Connection nesnesi.

Özel durumlar:

- XMSException
- XMS_X_SECURITY_EXCEPTION

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler IPropertyContext arabiriminden edinilir:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IConnectionMetaVerileri

Bir ConnectionMetaVeri nesnesi, bir bağlantı hakkında bilgi sağlar.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnectionMetaData
```

ConnectionMetaVeri nesnesinin XMS tanımlı özelliklerinin bir listesi için bkz. "ConnectionMetaVerilerinin Özellikleri" sayfa 2005.

.NET Özellikler

JMSXPropertyNames -JMS Tanımlı İleti Özelliklerini Al

Arabirim:

```
System.Collections.IEnumerator JMSXPropertyNames
{
    get;
}
```

Bağlantı tarafından desteklenen JMS tanımlı ileti özelliklerinin adlarına ilişkin bir numaralandırma döndürür.

JMS tanımlı ileti özellikleri, bir aracıya gerçek zamanlı bağlantı tarafından desteklenmez.

Özel durumlar:

- XMSException

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden edinilir:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IDestination (Hedef)

Hedef, bir uygulamanın iletileri gönderdiği ya da bir uygulamanın iletileri aldığı ya da her ikisini birden aldığı bir kaynaktır.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IDestination
```

Bir Hedef nesnenin XMS tanımlı özelliklerinin bir listesi için bkz. "[Hedef Özellikleri](#)" sayfa 2006.

.NET Özellikler

Ad-Hedef Adını Al

Arabirim:

```
String Name
{
    get;
}
```

Hedefin adını alın. Ad, bir kuyruğun adını ya da bir konunun adını içeren bir dizedir.

Özel durumlar:

- XMSException

TypeId -Hedef Tipi Al

Arabirim:

```
DestinationType TypeId
{
```



```
    get;  
}
```

Hedefin tipini alın. Hedefin tipi aşağıdaki değerlerden biridir:

DestinationType.Queue
DestinationType.Topic

Özel durumlar:

- XMSException

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden edinilir:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

ExceptionHandler

Bir uygulama, bir bağlantıyla ilgili bir sorunla ilgili zamanuyumsuz olarak bildirim almak için kural dışı durum dinleyicisini kullanır.

Kalıtım sıradüzeni:

Yok

Bir uygulama bir bağlantıyı yalnızca iletileri zamanuyumsuz olarak kullanmak için kullanıyorsa ve başka bir amaçla kullanmıyorsa, uygulamanın bağlantıyla ilgili bir sorunu öğrenmesinin tek yolu bir kural dışı durum dinleyicisi kullanmaktır. Diğer durumlarda, kural dışı durum dinleyicisi, bir bağlantıyla ilgili bir sorunu hemen öğrenmenin bir yolunu, XMS' e yapılan bir sonraki zamanuyumlu çağrıyı beklemekten daha iyi sağlayabilir.

Ata

ExceptionHandler -Kural Dışı Durum Dinleyicisi

Arabirim:

```
public delegate void ExceptionListener(Exception ex)
```

Bir bağlantıyla ilgili bir sorunu uygulamaya bildirin.

Bu temsilciyi gerçekleştiren yöntemler bağlantıyla kaydedilebilir.

Kural dışı durum dinleyicilerini kullanma hakkında daha fazla bilgi için bkz. [.NETiçinde ileti ve kural dışı durum dinleyicilerini kullanma](#).

Parametreler:

kural dışı durum (giriş)

XMStarafından oluşturulan bir kural dışı duruma ilişkin gösterge.

Dönüşler:

Geçersiz

IllegalStateKural Dışı Durumu

XMS , bir uygulama bir yöntemi yanlış ya da uygun olmayan bir zamanda ya da XMS istenen işlem için uygun durumda değilse bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSException
|
+----IBM.XMS.Exception
|
+----IBM.XMS.IllegalStateException
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [XMSException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

InitialContext

Bir uygulama, yönetilen nesnelere havuzundan alınan nesne tanımlamalarından nesne yaratmak için InitialContext nesnesini kullanır.

Kalıtım sıradüzeni:

Yok

.NET Özellikler

Ortam-Ortamı edinin

Arabirim:

```
Hashtable Environment
{
    get;
}
```

Ortamı edinin.

Özel durumlar:

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

Oluşturucular

InitialContext -Başlangıç Bağlamı Yarat

Arabirim:

```
InitialContext(Hashtable env);
```

Bir InitialContext nesnesi yaratın.

Parametreler:

Denetlenen nesnelere havuzuyla bağlantı kurmak için gereken bilgiler, Hashtable ortamındaki oluşturucuya sağlanır.

Özel durumlar:

- XMSException

yöntemler

AddToOrtamı-Ortama Yeni Bir Özellik Ekleyin

Arabirim:

```
Object AddToEnvironment(String propName, Object propVal);
```

Ortama yeni bir özellik ekleyin.

Parametreler:

propName (giriş)

Eklenecek özelliğin adını içeren bir String nesnesi.

propVal (giriş)

Eklenecek özelliğin değeri.

Dönüşler:

Özelliğin eski değeri.

Özel durumlar:

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

Bu bağlamı kapat

Arabirim:

```
void Close()
```

Bu bağlamı kapatın.

Parametreler:

Yok

Dönüşler:

Yok

Özel durumlar:

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

Arama-Nesneyi Başlangıç Bağlamında Ara

Arabirim:

```
Object Lookup(String name);
```

Denetlenen nesnelere havuzundan alınan bir nesne tanımlamasından nesne yaratın.

Parametreler:

ad (giriş)

Alınacak denetlenen nesnenin adını içeren bir String nesnesi. Ad basit bir ad ya da karmaşık bir ad olabilir. Daha fazla ayrıntı için bkz. [Denetlenen nesnelere alınması](#).

Dönüşler:

Alınmakta olan nesnenin tipine bağlı olarak bir IConnectionFactory ya da IDestination. İşlev dizine erişebiliyor, ancak gerekli nesneyi bulamıyorsa boş değer döndürülür.

Özel durumlar:

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

RemoveFromEnvironment-Ortamdan bir özelliği kaldırma

Arabirim:

```
Object RemoveFromEnvironment(String propName);
```

Ortamdan bir özelliği kaldırın.

Parametreler:

propName (giriş)

Kaldırılacak özelliğin adını saran bir String nesnesi.

Dönüşler:

Kaldırılan nesne.

Özel durumlar:

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

InvalidClientIDException

XMS , bir uygulama bir bağlantı için istemci tanıtıcısı ayarlamaya çalışıyorsa, ancak istemci tanıtıcısı geçerli değilse ya da kullanımda ise bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.InvalidClientIDException
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

InvalidDestinationKural Dışı Durumu

XMS , bir uygulama geçerli olmayan bir hedef belirtiyorsa bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.InvalidDestinationException
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

InvalidSelectorKural Dışı Durumu

XMS , bir uygulama sözdizimi geçerli olmayan bir ileti seçici ifadesi sağlarsa bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
```

```
|  
+----IBM.XMS.InvalidSelectorException
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler `XMSEException` arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

IMapMessage

Harita iletisi, gövdesi, her değer ilişkili bir veri tipine sahip olduğu bir ad-değer çiftleri kümesinden oluşan bir iletidir.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext  
|  
+----IBM.XMS.IMessage  
|  
+----IBM.XMS.IMapMessage
```

Bir uygulama ad-değer çiftinin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Bu örtük dönüştürme biçimiyle ilgili daha fazla bilgi için [XMS iletisinin gövdesi](#) içindeki eşlem iletileriyle ilgili bilgilere bakın.

.NET Özellikler

MapNames -Eşlem Adlarını Al

Arabirim:

```
System.Collections.IEnumerator MapNames  
{  
    get;  
}
```

Eşlem iletisinin gövdesindeki adlara ilişkin bir sıralı değer listesi alın.

Özel durumlar:

- `XMSEException`

yöntemler

GetBoolean -Boole Değerini Al

Arabirim:

```
Boolean GetBoolean(String name);
```

Eşlem iletisinin gövdesinden adla tanımlanan Boole değerini alın.

Parametreler:

ad (giriş)

Boole değerini tanıtan adı içeren bir String nesnesi.

Dönüşler:

Eşlem iletisinin gövdesinden alınan Boole değeri.

Özel durumlar:

- `XMSEException`

GetByte -Byte Al

Arabirim:

```
Byte    GetByte(String name);  
Int16   GetSignedByte(String name);
```

Eşlem iletisinin gövdesinden adla tanımlanan baytı alın.

Parametreler:

ad (giriş)

Baytı tanıtan adı saran bir String nesnesi.

Dönüşler:

Eşlem iletisinin gövdesinden alınan bayt. Bayt üzerinde veri dönüştürme işlemi gerçekleştirilmez.

Özel durumlar:

- XMSEException

GetBytes -Bayt Al

Arabirim:

```
Byte[]  GetBytes(String name);
```

Eşlem iletisinin gövdesinden adla tanıtılan bayt dizisini alın.

Parametreler:

ad (giriş)

Bayt dizisini tanıtan adı saran bir String nesnesi.

Dönüşler:

Dizideki bayt sayısı.

Özel durumlar:

- XMSEException

GetChar -Karakter Al

Arabirim:

```
Char    GetChar(String name);
```

Eşlem iletisinin gövdesinden adla tanıtılan karakteri alın.

Parametreler:

ad (giriş)

Karakter tanıtan adı içeren bir String nesnesi.

Dönüşler:

Eşlem iletisinin gövdesinden alınan karakter.

Özel durumlar:

- XMSEException

GetDouble -Çift Duyarlıklı Kayan Noktalı Sayı Al

Arabirim:

```
Double  GetDouble(String name);
```

Eşlem iletisinin gövdesinden, adla tanıtılan çift duyarlıklı kayan noktalı sayı alın.

Parametreler:**ad (giriş)**

Çift duyarlıklı kayan noktalı sayıyı tanıtan adı içeren bir String nesnesi.

Dönüşler:

Harita iletisinin gövdesinden alınan çift duyarlıklı kayan noktalı sayı.

Özel durumlar:

- XMSEException

GetFloat -Kayan Nokta Numarasını Al

Arabirim:

```
Single GetFloat(String name);
```

Eşlem iletisinin gövdesinden adla tanımlanan kayan nokta numarasını alın.

Parametreler:**ad (giriş)**

Kayan noktalı sayıyı tanıtan adı saran bir String nesnesi.

Dönüşler:

Harita iletisinin gövdesinden alınan kayan nokta numarası.

Özel durumlar:

- XMSEException

GetInt -Tamsayı Al

Arabirim:

```
Int32 GetInt(String name);
```

Eşlem iletisinin gövdesinden adla tanımlanan tamsayıyı alın.

Parametreler:**ad (giriş)**

Tamsayıyı tanıtan adı saran bir String nesnesi.

Dönüşler:

Eşlem iletisinin gövdesinden alınan tamsayı.

Özel durumlar:

- XMSEException

GetLong -Get Long Integer (Uzun Tamsayı Al)

Arabirim:

```
Int64 GetLong(String name);
```

Eşlem iletisinin gövdesinden adla tanımlanan uzun tamsayıyı alın.

Parametreler:**ad (giriş)**

Uzun tamsayıyı tanıtan adı saran bir String nesnesi.

Dönüşler:

Eşlem iletisinin gövdesinden alınan uzun tamsayı.

Özel durumlar:

- XMSEException

GetObject -Nesneyi Al

Arabirim:

```
Object GetObject(String name);
```

Eşlem iletisinin gövdesinden bir ad-değer çiftinin değerine ilişkin bir başvuru alın. Ad-değer çifti adla tanıtılır.

Parametreler:**ad (giriş)**

Ad-değer çiftinin adını içeren bir String nesnesi.

Dönüşler:

Aşağıdaki nesne tiplerinden biri olan değer:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

Özel durumlar:

- XMSEException

GetShort -Kısa Tamsayı Al

Arabirim:

```
Int16 GetShort(String name);
```

Eşlem iletisinin gövdesinden adla tanıtılan kısa tamsayıyı alın.

Parametreler:**ad (giriş)**

Kısa tamsayıyı tanıtan adı saran bir String nesnesi.

Dönüşler:

Eşlem iletisinin gövdesinden alınan kısa tamsayı.

Özel durumlar:

- XMSEException

GetString -Dizgiyi Al

Arabirim:

```
String GetString(String name);
```

Eşlem iletisinin gövdesinden adla tanıtılan dizgiyi alın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesinde dizgiyi tanıtan adı içeren bir String nesnesi.

Dönüşler:

Eşlem iletisinin gövdesinden alınan dizgiyi saran bir String nesnesi. Veri dönüştürme gerekiyorsa, bu değer, dönüştürmeden sonraki dizedir.

Özel durumlar:

- XMSEException

ItemExists -Denetim Adı-Değer çifti Var

Arabirim:

```
Boolean ItemExists(String name);
```

Eşlem iletisinin gövdesinin belirtilen ada sahip bir ad-değer çifti içerip içermediğini denetleyin.

Parametreler:**ad (giriş)**

Ad-değer çiftinin adını içeren bir String nesnesi.

Dönüşler:

- True, eşlem iletisinin gövdesi belirtilen ada sahip bir ad-değer çifti içeriyorsa.
- False, eşlem iletisinin gövdesi belirtilen ada sahip bir ad-değer çifti içermiyorsa.

Özel durumlar:

- XMSEException

SetBoolean -Boole Değerini Ayarla

Arabirim:

```
void SetBoolean(String name, Boolean value);
```

Eşlem iletisinin gövdesinde bir Boole değeri ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesindeki Boole değerini tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak Boole değeri.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetByte -Bayt Ayarla

Arabirim:

```
void SetByte(String name, Byte value);  
void SetSignedByte(String name, Int16 value);
```

Eşlem iletisinin gövdesinde bir bayt ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesindeki baytı tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak bayt.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetBytes -Baytları Ayarla

Arabirim:

```
void SetBytes(String name, Byte[] value);
```

Eşlem iletisinin gövdesinde bir bayt dizisi ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesindeki bayt dizisini tanıtmak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak bayt dizisi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetChar -Karakter Ayarla

Arabirim:

```
void SetChar(String name, Char value);
```

Eşlem iletisinin gövdesinde 2 baytlık bir karakter belirleyin.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesindeki karakteri tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak karakter.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetDouble -Çift Duyarlıklı Kayan Noktalı Sayı Ayarla

Arabirim:

```
void SetDouble(String name, Double value);
```

Eşlem iletisinin gövdesinde çift duyarlıklı bir kayan noktalı sayı ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesinde çift duyarlıklı kayan noktalı sayıyı tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak çift duyarlıklı kayan noktalı sayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetFloat -Kayan Nokta Numarasını Ayarla

Arabirim:

```
void SetFloat(String name, Single value);
```

Eşlem iletisinin gövdesinde bir kayan nokta numarası ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesinde kayan nokta numarasını tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak kayan noktalı sayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetInt -Tamsayı Ayarla

Arabirim:

```
void SetInt(String name, Int32 value);
```

Eşlem iletisinin gövdesinde bir tamsayı ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesindeki tamsayıyı tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetLong -Uzun Tamsayı Ayarla

Arabirim:

```
void SetLong(String name, Int64 value);
```

Eşlem iletisinin gövdesinde uzun bir tamsayı ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesindeki uzun tamsayıyı tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak uzun tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetObject -Nesne Ayarla

Arabirim:

```
void SetObject(String name, Object value);
```

Eşlem iletisinin gövdesinde XMS ilkel tipi olması gereken bir değer ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesindeki değeri tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak değeri içeren bir bayt dizisi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetShort -Kısa Tamsayı Ayarla

Arabirim:

```
void SetShort(String name, Int16 value);
```

Eşlem iletisinin gövdesinde kısa bir tamsayı ayarlayın.

Parametreler:**ad (giriş)**

Eşlem iletisinin gövdesindeki kısa tamsayıyı tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak kısa tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

SetString -Dizgiyi Ayarla

Arabirim:

```
void SetString(String name, String value);
```

Eşlem iletisinin gövdesinde bir dizgi ayarlayın.

Parametreler:

ad (giriş)

Eşlem iletişinin gövdesindeki dizgiyi tanımlamak için adı saran bir String nesnesi.

değer (giriş)

Ayarlanacak dizgiyi saran bir String nesnesi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

Devralınan özellikler ve yöntemler

Aşağıdaki özellikler IMessage arabiriminden edinilir:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSReteslim, JMSReplyTo, JMSTimestamp, JMSType, Özellikler

Aşağıdaki yöntemler IMessage arabiriminden edinilir:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden edinilir:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IMessage

İleti nesnesi, bir uygulamanın gönderdiği ya da aldığı bir iletiyi gösterir. IMessage, IMapMessage gibi ileti sınıflarına ilişkin bir üst sınıftır.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
```

Bir İleti nesnesindeki JMS ileti üstbilgisi alanlarının bir listesi için bkz. XMS iletişinin üstbilgi alanları. Bir ileti nesnesinin JMS tanımlı özelliklerinin bir listesi için iletinin JMS tanımlı özellikleri başlıklı konuya bakın. Bir İleti nesnesinin IBM tanımlı özelliklerinin bir listesi için bkz. IBM tanımlı ileti özellikleri. İleti nesnesine ilişkin JMS_IBM_MQMD* özelliklerinin bir listesi için bkz. "JMS_IBM_MQMD* özellikleri" sayfa 2009

İletiler çöp toplayıcı tarafından silinir. Bir ileti silindiğinde, bu, kullandığı kaynakları serbest bırakır.

.NET Özellikler

GetJMSCorrelationID-Get and Set JMSCorrelationID

Arabirim:

```
String JMSCorrelationID
{
    get;
    set;
}
```

İletinin ilinti tanıtıcısını String nesnesi olarak alın ve ayarlayın.

Özel durumlar:

- XMSEException

JMSDeliveryMode -Al ve Ayarla JMSDeliveryMode

Arabirim:

```
DeliveryMode JMSDeliveryMode
{
    get;
    set;
}
```

İletinin teslim kipini alın ve ayarlayın.

İletinin teslim kipi aşağıdaki değerlerden biridir:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

Gönderilmemiş yeni yaratılan bir ileti için, teslim kipi `DeliveryMode.NonPersistent` olan bir aracıya gerçek zamanlı bağlantı dışında, teslim kipi `DeliveryMode.Kalıcı` olur. Alınan bir ileti için, yöntem, ileti gönderildiğinde `IMessageProducer.send ()` çağrısı tarafından ayarlanan teslim kipini döndürür; ancak, alan uygulama `JMSDeliveryMode` ayarını yaparak teslim kipini değiştirmedeği sürece.

Özel durumlar:

- XMSEException

JMSDestination-JMSDestination değerini al ve ayarla

Arabirim:

```
IDestination JMSDestination
{
    get;
    set;
}
```

İletinin hedefini alın ve ayarlayın.

Hedef, ileti gönderildiğinde `IMessageProducer.send ()` çağrısı tarafından ayarlanır. `JMSDestination` değeri yoksayıldı. Ancak, alınan bir iletinin hedefini değiştirmek için `JMSDestination` kullanabilirsiniz.

Gönderilmeyen yeni yaratılan bir ileti için, gönderen uygulama `JMSDestination` değerini ayarlayarak bir hedef belirlemezse, yöntem boş değerli bir Hedef nesnesi döndürür. Alınan bir ileti için, yöntem, ileti gönderildiğinde `IMessageProducer.send ()` çağrısı tarafından ayarlanan hedefe ilişkin bir Hedef nesnesi döndürür; ancak, alan uygulama `JMSDestination` değerini ayarlayarak hedefi değiştirmez.

Özel durumlar:

- XMSEException

JMSExpiration-JMSExpiration 'ı alın ve ayarlayın

Arabirim:

```
Int64 JMSExpiration
{
    get;
    set;
}
```

İletinin süre bitim zamanını alın ve ayarlayın.

Süre bitim zamanı, ileti gönderildiğinde `IMessageProducer.send ()` çağrısı tarafından ayarlanır. Değeri, gönderen uygulamanın belirttiği şekilde, iletinin gönderildiği zamana yaşam süresi eklenerek hesaplanır. Süre bitim saati, 1 Ocak 1970 'te GMT 00:00:00 'dan bu yana milisaniye olarak ifade edilir.

Gönderilmeyen yeni yaratılan bir ileti için, gönderen uygulama `JMSExpiration` değerini ayarlayarak farklı bir süre bitimi ayarlamadıkça süre bitimi 0 olur. Alınan bir ileti için, yöntem, ileti gönderildiğinde `IMessageProducer.send ()` çağrısı tarafından ayarlanan süre bitim zamanını döndürür; ancak, alıcı uygulama `JMSExpiration` ayarlanarak süre bitim zamanını değiştirmez.

Yaşam süresi 0 ise, `IMessageProducer.send ()` çağrısı, iletinin süresinin dolmadığını belirtmek için süre bitimini 0 olarak ayarlar.

XMS süresi dolan iletileri atar ve uygulamalara teslim etmez.

Özel durumlar:

- `XMSException`

JMSMessageID - JMSMessageID değerini alın ve ayarlayın

Arabirim:

```
String JMSMessageID
{
    get;
    set;
}
```

İletinin ileti tanıtıcısını, ileti tanıtıcısını saran bir dizgi nesnesi olarak alın ve ayarlayın.

İleti gönderildiğinde, ileti tanıtıcısı `IMessageProducer.send ()` çağrısı tarafından ayarlanır. Alınan bir ileti için, yöntem, alan uygulama `JMSMessageID` ayarlanarak ileti tanıtıcısını değiştirmedikçe, ileti gönderildiğinde `IMessageProducer.send ()` çağrısı tarafından ayarlanan ileti tanıtıcısını döndürür.

İletinin ileti tanıtıcısı yoksa, yöntem boş değer döndürür.

Özel durumlar:

- `XMSException`

JMSPriority-JMSPriority değerini alın ve ayarlayın

Arabirim:

```
Int32 JMSPriority
{
    get;
    set;
}
```

İletinin önceliğini alın ve ayarlayın.

Öncelik, ileti gönderildiğinde `IMessageProducer.send ()` çağrısı tarafından ayarlanır. Değer, en düşük öncelik olan 0, en yüksek öncelik olan 9 aralığındaki bir tamsayıdır.

Gönderilmeyen yeni oluşturulan bir ileti için, gönderen uygulama `JMSPriority` ayarını kullanarak farklı bir öncelik belirlemezse, öncelik 4 olur. Alınan bir ileti için, alan uygulama `JMSPriority` ayarını kullanarak önceliği değiştirmezse, ileti gönderildiğinde `IMessageProducer.send ()` çağrısı tarafından ayarlanan önceliği döndürür.

Özel durumlar:

- `XMSException`

Arabirim:

```
Boolean JMSRedelivered
{
    get;
    set;
}
```

İletinin yeniden teslim edilip edilmediğini ve iletinin yeniden teslim edilip edilmediğini belirtin. Gösterge, ileti alındığında IMessageConsumer.receive () çağrısı tarafından ayarlanır.

Bu özellik aşağıdaki değerleri içerir:

- True, ileti yeniden teslim ediliyorsa.
- False, ileti yeniden teslim edilmezse.

Bir aracıya gerçek zamanlı bağlantı için değer her zaman False olur.

İleti gönderilmeden önce JMSRedelivery tarafından ayarlanan bir yeniden teslim göstergesi, ileti gönderildiğinde IMessageProducer.send () çağrısı tarafından yoksayılır ve yoksayılır ve ileti alındığında IMessageConsumer.receive () çağrısıyla değiştirilir. Ancak, alınan bir iletinin gösterimini değiştirmek için JMSReteslim edildi özelliğini kullanabilirsiniz.

Özel durumlar:

- XMSEException

JMSReplyTo - JMSReplyTo Alma ve Ayarlama

Arabirim:

```
IDestination JMSReplyTo
{
    get;
    set;
}
```

İletiyeye yanıt gönderileceği hedefi alın ve ayarlayın.

Bu özelliğin değeri, iletiye yanıt gönderileceği hedefe ilişkin bir Hedef nesnesidir. Boş değerli bir Hedef nesnesi, yanıt beklenmediği anlamına gelir.

Özel durumlar:

- XMSEException

JMSTimestamp-Get ve Set JMSTimestamp

Arabirim:

```
Int64 JMSTimestamp
{
    get;
    set;
}
```

İletinin gönderildiği saati alın ve ayarlayın.

Zaman damgası, ileti gönderildiğinde IMessageProducer.send () çağrısı tarafından ayarlanır ve 1 Ocak 1970 tarihinde 00:00:00 GMT saatinden itibaren milisaniye olarak ifade edilir.

Gönderilmeyen yeni yaratılan bir ileti için, gönderen uygulama JMSTimestamp ayarını kullanarak farklı bir zaman damgası ayarlamadıkça, zaman damgası 0 olur. Alınan bir ileti için, yöntem, alan uygulama JMSTimestamp ayarını kullanarak zaman damgasını değiştirmedikçe, ileti gönderildiğinde IMessageProducer.send () çağrısı tarafından ayarlanan zaman damgasını döndürür.

Özel durumlar:

- XMSEException

Notlar:

1. Zaman damgası tanımlanmamışsa, yöntem 0 değerini döndürür, ancak kural dışı durum oluşmaz.

JMSType-Get ve Set JMSType

Arabirim:

```
String JMSType
{
    get;
    set;
}
```

İletinin tipini alın ve ayarlayın.

JMSType değeri, iletinin tipini saran bir dizedir. Veri dönüştürme gerekiyorsa, bu değer, dönüştürmeden sonraki tiptir.

Özel durumlar:

- XMSEException

PropertyNames -Özellikleri Al

Arabirim:

```
System.Collections.IEnumerator PropertyNames
{
    get;
}
```

İletinin ad özelliklerine ilişkin bir sıralı değer listesi alın.

Özel durumlar:

- XMSEException

yöntemler

Onayla-Bildir

Arabirim:

```
void Acknowledge();
```

Bu iletiyi ve oturum tarafından daha önce alınmamış tüm iletileri onaylayın.

Oturumun alındı bildirimini kipi AcknowledgeMode.ClientAcknowledgeise, bir uygulama bu yöntemi çağırabilir. Oturumda başka bir alındı bildirimini kipi varsa ya da işlem yapıldıysa, yöntemeye yönelik çağrılar yoksayılr.

Alınan, ancak alınmayan iletiler yeniden teslim edilebilir.

İletileri onaylama hakkında daha fazla bilgi için bkz. [../develop/xms_cmesack.dita#xms_cmesack](#).

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- IllegalStateKural Dışı Durumu

ClearBody -Gövdeyi Temizle

Arabirim:

```
void ClearBody();
```

İletinin gövdesini temizleyin. Üstbilgi alanları ve ileti özellikleri temizlenmez.

Bir uygulama bir ileti gövdesini temizlerse, gövde, yeni yaratılan bir iletide boş bir gövde ile aynı durumda kalır. Yeni yaratılan bir iletideki boş bir gövdenin durumu, ileti gövdesinin tipine bağlıdır. Daha fazla bilgi için bkz. [XMS iletisinin gövdesi](#).

Bir uygulama, gövde ne durumda olursa olsun, bir ileti gövdesini istediği zaman temizleyebilir. Bir ileti gövdesi salt okunur ise, bir uygulamanın gövdeye yazabilmesinin tek yolu, uygulamanın önce gövdeyi temizlemesi içindir.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

ClearProperties -Özellikleri Temizle

Arabirim:

```
void ClearProperties();
```

İletinin özelliklerini temizleyin. Üstbilgi alanları ve ileti gövdesi temizlenmez.

Bir uygulama bir iletinin özelliklerini temizlerse, özellikler okunabilir ve yazılabilir olur.

Bir uygulama, özelliklerin hangi durumda olduğu fark etmez, bir iletinin özelliklerini istediği zaman temizleyebilir. Bir iletinin özellikleri salt okunur ise, özelliklerin yazılabilir hale gelmesinin tek yolu, uygulamanın önce özellikleri temizlemesi içindir.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

PropertyExists -Özellik Var Olup Olmadığını Denetleyin

Arabirim:

```
Boolean PropertyExists(String propertyName);
```

İletinin belirtilen ada sahip bir özelliği olup olmadığını denetleyin.

Parametreler:

propertyName (giriş)

Özelliğın adını saran bir String nesnesi.

Dönüşler:

- True, iletinin belirtilen ada sahip bir özelliğı varsa.
- False, iletinin belirtilen ada sahip bir özelliğı yoksa.

Özel durumlar:

- XMSException

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden edinilir:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IMessageConsumer

Bir uygulama, hedefe gönderilen iletileri almak için bir ileti tüketicisi kullanır.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessageConsumer
```

Bir MessageConsumer nesnesinin XMS tanımlı özelliklerinin bir listesi için bkz. "[MessageConsumer Özellikleri](#)" sayfa 2012.

.NET Özellikler

MessageListener -İleti Dinleyicisini Al ve Ayarla

Arabirim:

```
MessageListener MessageListener
{
    get;
    set;
}
```

İleti tüketicisine kayıtlı ileti dinleyiciyi alın ve ileti tüketicisine bir ileti dinleyici kaydedin.

İleti tüketicisine kayıtlı bir ileti dinleyici yoksa, MessageListener boş değerlidir. İleti dinleyici zaten ileti tüketicisine kayıtlıysa, bunun yerine boş değer belirterek kaydı iptal edebilirsiniz.

İleti dinleyicilerinin kullanılmasıyla ilgili ek bilgi için [.NET ' te ileti ve kural dışı durum dinleyicilerinin kullanılması](#) başlıklı konuya bakın.

Özel durumlar:

- XMSException

MessageSelector -İleti Seçiciyi Al

Arabirim:

```
String MessageSelector
{
    get;
}
```

İleti tüketicisine ilişkin ileti seçiciyi alın. Dönüş değeri, ileti seçici ifadesini saran bir String nesnesidir. Veri dönüştürme gerekiyorsa, bu değer, dönüştürmeden sonraki ileti seçici ifadesidir. İleti tüketicisinin ileti seçicisi yoksa, MessageSelector değeri boş değerli bir String nesnesidir.

Özel durumlar:

- XMSEException

yöntemler

İleti Tüketiciyi Kapat

Arabirim:

```
void Close();
```

İleti tüketicisini kapatın.

Bir uygulama, zaten kapalı olan bir ileti tüketiciyi kapatmaya çalışırsa, çağrı yoksayılr.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

Al-Al

Arabirim:

```
IMessage Receive();
```

İleti tüketicisine ilişkin sonraki iletiyi alın. Çağrı, bir ileti için süresiz olarak ya da ileti tüketicisi kapatılıncaya kadar bekler.

Parametreler:

Yok

Dönüşler:

İleti nesnesine ilişkin bir işaretçi. Çağrı bir ileti beklerken ileti tüketicisi kapanırsa, yöntem boş değerli bir ileti nesnesine bir gösterge döndürür.

Özel durumlar:

- XMSEException

Al-Al (bekleme aralığıyla)

Arabirim:

```
IMessage Receive(Int64 delay);
```

İleti tüketicisine ilişkin sonraki iletiyi alın. Çağrı, bir ileti için yalnızca belirtilen bir dönemi ya da ileti tüketicisi kapatılıncaya kadar bekler.

Parametreler:

gecikme (giriş)

Aramanın ileti bekleyeceği milisaniye cinsinden süre. 0 bekleme aralığı belirtirseniz, çağrı ileti için süresiz olarak bekler.

Dönüşler:

İleti nesnesine ilişkin bir işaretçi. Bekleme aralığı sırasında hiçbir ileti gelmezse ya da çağrı bir ileti beklerken ileti tüketicisi kapanırsa, yöntem boş değerli bir ileti nesnesine bir gösterge döndürür, ancak kural dışı durum göndermez.

Özel durumlar:

- XMSEException

ReceiveNoBekleme-Bekleme ile Alma

Arabirim:

```
IMessage ReceiveNoWait();
```

Hemen varsa, ileti tüketicisine ilişkin sonraki iletiyi alın.

Parametreler:

Yok

Dönüşler:

Bir İleti nesnesine ilişkin gösterge. Hemen kullanılabilir bir ileti yoksa, yöntem boş değerli bir ileti nesnesine bir gösterge döndürür.

Özel durumlar:

- XMSEException

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler IPropertyContext arabiriminden edinilir:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

MessageEOFException

XMS , XMS bir uygulama bayt iletisinin gövdesini okurken bir bayt ileti akımının sonuyla karşılaşır bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageEOFException
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler XMSEException arabiriminden edinilir:

GetErrorCode, GetLinkedException

MessageFormatKural Dışı Durumu

XMS , XMS geçerli olmayan bir biçime sahip bir iletiyle karşılaşırsa bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSEnception
|
+----IBM.XMS.XMSEnception
|
+----IBM.XMS.MessageFormatEnception
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [XMSEnception](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedEnception](#)

IMessageListener (temsilci)

Bir uygulama, iletileri zamanuyumsuz olarak almak için bir ileti dinleyicisini kullanır.

Kalıtım sıradüzeni:

Yok

Ata

MessageListener -İleti Dinleyici

Arabirim:

```
public delegate void MessageListener(IMessage msg);
```

Bir iletiyi ileti tüketicisine zamanuyumsuz olarak teslim edin.

Bu temsilciyi gerçekleştiren yöntemler bağlantıyla kaydedilebilir.

İleti dinleyicilerini kullanma hakkında daha fazla bilgi için bkz. [.NET içinde ileti ve kural dışı durum dinleyicilerini kullanma.](#)

Parametreler:

mesg (giriş)
İleti nesnesi.

Dönüşler:

Geçersiz

MessageNotReadableEnception

XMS , bir uygulama yalnızca yazma işlemi olan bir iletinin gövdesini okuma girişiminde bulunursa bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSEnception
|
+----IBM.XMS.XMSEnception
|
+----IBM.XMS.MessageNotReadableEnception
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [XMSEnception](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

MessageNotWritableException

XMS , bir uygulama salt okunur bir iletinin gövdesine yazma girişiminde bulunursa bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSException
|
+----IBM.XMS.XMSException
|
+----IBM.XMS.MessageNotWritableException
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [XMSException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

IMessageProducer

Bir uygulama, bir hedefe ileti göndermek için ileti üreticiyi kullanır.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessageProducer
```

Bir [MessageProducer](#) nesnesinin XMS tanımlı özelliklerinin bir listesi için bkz. "[MessageProducer Özellikleri](#)" sayfa 2013.

.NET Özellikler

DeliveryMode -Varsayılan Teslim Kipini Al ve Ayarla

Arabirim:

```
DeliveryMode DeliveryMode
{
    get;
    set;
}
```

İleti üreticisi tarafından gönderilen iletiler için varsayılan teslim kipini alın ve ayarlayın.

Varsayılan teslim kipi aşağıdaki değerlerden birine sahiptir:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

Bir aracıya gerçek zamanlı bağlantı için değer `DeliveryMode.NonPersistent` olmalıdır.

Varsayılan değer, varsayılan değeri `DeliveryMode.NonPersistent` olan bir aracıya gerçek zamanlı bağlantı dışında `DeliveryMode.Persistent` değeridir.

Özel durumlar:

- `XMSException`

Hedef-Hedef Al

Arabirim:

```
IDestination Destination
{
  get;
}
```

İleti üreticisinin hedefini alın.

Parametreler:

Yok

Dönüşler:

Hedef nesne. İleti üreticisinin hedefi yoksa, yöntem boş değerli bir Hedef nesnesi döndürür.

Özel durumlar:

- XMSException

DisableMsgTanıtıcısı-İleti Tanıtıcısı İşaretini Al ve Ayarla

Arabirim:

```
Boolean DisableMessageID
{
  get;
  set;
}
```

Bir alıcı uygulamanın ileti üreticisi tarafından gönderilen iletilere ileti tanıtıcılarının eklenmesini gerektirip gerektirmediğini ve alıcı uygulamanın ileti üreticisinin gönderdiği iletilere ileti tanıtıcılarının eklenmesini gerektirip gerektirmediğini gösteren bir gösterge alın.

Bir kuyruk yöneticisine yönelik bağlantıda ya da bir aracıya gerçek zamanlı bağlantıda bu işaret yoksayıdır. Bir hizmet bütünleştirme veriyolu bağlantısında işaret yerine getirilmiştir.

DisabledMsgtanıtıcısı aşağıdaki değerleri içerir:

- True, alan bir uygulama ileti tanıtıcılarının ileti üreticisi tarafından gönderilen iletilere eklenmesini gerektirmiyorsa.
- False, alan bir uygulama ileti tanıtıcılarının ileti üreticisi tarafından gönderilen iletilere eklenmesini gerektiriyorsa.

Özel durumlar:

- XMSException

DisableMsgTS-Zaman Damgasını Al ve Ayarla İşareti

Arabirim:

```
Boolean DisableMessageTimestamp
{
  get;
  set;
}
```

Bir alıcı uygulamanın, ileti üreticisi tarafından gönderilen iletilere zaman damgası eklenmesini gerektirip gerektirmediğini ve alıcı uygulamanın, ileti üreticisi tarafından gönderilen iletilere zaman damgası eklenmesini gerektirip gerektirmediğini belirtin.

Bir aracıya gerçek zamanlı bağlantıda bu işaret yoksayıdır. Bir kuyruk yöneticisine yönelik bağlantıda ya da bir hizmet tümleştirme veriyolu bağlantısında, bu işaret kullanılır.

DisableMsgTS aşağıdaki değerlere sahiptir:

- True, alan bir uygulama, ileti üreticisi tarafından gönderilen iletilere zaman damgalarının eklenmesini gerektirmiyorsa.
- False, alan bir uygulama, ileti üreticisi tarafından gönderilen iletilere zaman damgalarının eklenmesini gerektiriyorsa.

Dönüşler:

Özel durumlar:

- XMSEException

Öncelik-Varsayılan Önceliği Al ve Ayarla

Arabirim:

```
Int32 Priority
{
    get;
    set;
}
```

İleti üreticisi tarafından gönderilen iletiler için varsayılan önceliği alın ve ayarlayın.

Varsayılan ileti önceliğinin değeri, en düşük öncelik olan 0 aralığındaki bir tamsayıdır ve en yüksek öncelik olan 9 değerine ayarlanır.

Bir aracıya gerçek zamanlı bağlantıda, iletinin önceliği yoksayılr.

Özel durumlar:

- XMSEException

TimeToCanlılık-Varsayılan canlılık süresini alın ve ayarlayın

Arabirim:

```
Int64 TimeToLive
{
    get;
    set;
}
```

Bir iletinin süresi dolmadan önce var olduğu varsayılan süreyi alın ve ayarlayın.

Süre, ileti üreticisinin iletiyi göndermesinden itibaren ölçülür ve milisaniye cinsinden varsayılan yaşam süresidir. 0 değeri, bir iletinin süresinin hiçbir zaman sona ermeyeceğini gösterir.

Bir aracıya gerçek zamanlı bağlantı için bu değer her zaman 0 'dır.

Özel durumlar:

- XMSEException

yöntemler

İleti Üreticisini Kapat

Arabirim:

```
void Close();
```

İleti üreticisini kapatın.

Bir uygulama zaten kapalı olan bir ileti üreticisi kapatmaya çalışırsa, çağrı yoksayılr.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

Gönder-Gönder

Arabirim:

```
void Send(IMessage msg) ;
```

İleti üreticisi yaratıldığında belirtilen hedefe bir ileti gönderin. İleti üreticisinin varsayılan teslim kipini, önceliğini ve yaşam süresini kullanarak iletiyi gönderin.

Parametreler:**msg (giriş)**

İleti nesnesi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageFormatKural Dışı Durumu
- InvalidDestinationKural Dışı Durumu

Gönder-Gönder (bir teslim kipi, öncelik ve yaşam süresi belirleme)

Arabirim:

```
void Send(IMessage msg,  
          DeliveryMode deliveryMode,  
          Int32 priority,  
          Int64 timeToLive);
```

İleti üreticisi yaratıldığında belirtilen hedefe bir ileti gönderin. Belirtilen teslim kipini, önceliğini ve yaşam süresini kullanarak iletiyi gönderin.

Parametreler:**msg (giriş)**

İleti nesnesi.

deliveryMode (giriş)

Aşağıdaki değerlerden biri olması gereken ileti için teslim kipi:

DeliveryMode.Persistent
DeliveryMode.NonPersistent

Bir aracıya gerçek zamanlı bağlantı için değer DeliveryMode.NonPersistent olmalıdır.

öncelik (giriş)

İletinin önceliği. Değer, en düşük öncelik için 0, en yüksek öncelik için 9 aralığındaki bir tamsayı olabilir. Bir aracıya gerçek zamanlı bağlantıda değer yoksa 0'dır.

timeToCanlı (giriş)

Milisaneye cinsinden ileti için yaşam süresi. 0 değeri, iletinin süresinin hiçbir zaman dolmadığını gösterir. Bir aracıya gerçek zamanlı bağlantı için değer 0 olmalıdır.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageFormatKural Dışı Durumu
- InvalidDestinationKural Dışı Durumu
- IllegalStateKural Dışı Durumu

Gönder-Gönder (belirli bir hedefe)

Arabirim:

```
void Send(IDestination dest, IMessage msg) ;
```

İleti üreticisi yaratıldığında hedefi belirtilmemiş bir ileti üreticisi kullanıyorsanız, belirtilen hedefe ileti gönderin. İleti üreticisinin varsayılan teslim kipini, önceliğini ve yaşam süresini kullanarak iletiyi gönderin.

Genellikle, bir ileti üreticisi yarattığınızda hedef belirtirsiniz, ancak oluşturmadıysanız, her ileti gönderdiğinizde hedef belirtmeniz gerekir.

Parametreler:

dest (giriş)

Hedef nesne.

msg (giriş)

İleti nesnesi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageFormatKural Dışı Durumu
- InvalidDestinationKural Dışı Durumu

Gönder-Gönder (bir teslim kipi, öncelik ve yaşam süresi belirterek, belirtilen bir hedefe gönder)

Arabirim:

```
void Send(IDestination dest,  
          IMessage msg,  
          DeliveryMode deliveryMode,  
          Int32 priority,  
          Int64 timeToLive) ;
```

İleti üreticisi yaratıldığında hedefi belirtilmemiş bir ileti üreticisi kullanıyorsanız, belirtilen hedefe ileti gönderin. Belirtilen teslim kipini, önceliğini ve yaşam süresini kullanarak iletiyi gönderin.

Genellikle, bir ileti üreticisi yarattığınızda hedef belirtirsiniz, ancak oluşturmadıysanız, her ileti gönderdiğinizde hedef belirtmeniz gerekir.

Parametreler:

dest (giriş)

Hedef nesne.

msg (giriş)

İleti nesnesi.

deliveryMode (giriş)

Aşağıdaki değerlerden biri olması gereken ileti için teslim kipi:

```
DeliveryMode.Persistent  
DeliveryMode.NonPersistent
```

Bir aracıya gerçek zamanlı bağlantı için değer `DeliveryMode.NonPersistent` olmalıdır.

öncelik (giriş)

İletinin önceliği. Değer, en düşük öncelik için 0, en yüksek öncelik için 9 aralığındaki bir tamsayı olabilir. Bir aracıya gerçek zamanlı bağlantıda değer yoksayıdır.

timeToCanlı (giriş)

Milisaniye cinsinden ileti için yaşam süresi. 0 değeri, iletinin süresinin hiçbir zaman dolmadığını gösterir. Bir aracıya gerçek zamanlı bağlantı için değer 0 olmalıdır.

Dönüşler:

Geçersiz

Özel durumlar:

- `XMSException`
- `MessageFormatKural Dışı Durumu`
- `InvalidDestinationKural Dışı Durumu`
- `IllegalStateKural Dışı Durumu`

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler `IPropertyContext` arabiriminden edinilir:

`GetBooleanProperty`, `GetByteProperty`, `GetBytesProperty`, `GetCharProperty`, `GetDoubleProperty`, `GetFloatProperty`, `GetIntProperty`, `GetLongProperty`, `GetObjectProperty`, `GetShortProperty`, `GetStringProperty`, `SetBooleanProperty`, `SetByteProperty`, `SetBytesProperty`, `SetCharProperty`, `SetDoubleProperty`, `SetFloatProperty`, `SetIntProperty`, `SetLongProperty`, `SetObjectProperty`, `SetShortProperty`, `SetStringProperty`

IOBJECTMESSAGE

Nesne iletisi, gövdesi diziselleştirilmiş Java ya da .NET nesnesi olan bir iletidir.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IObjectMessage
```

.NET Özellikler

Nesne-Nesneyi Byte Olarak Al ve Ayarla

Arabirim:

```
System.Object Object
{
    get;
    set;
}

Byte[] GetObject();
```

Nesne iletisinin gövdesini oluşturan nesneyi alın ve ayarlayın.

Özel durumlar:

- `XMSException`
- `MessageNotReadableException`
- `MessageEOFException`

- MessageNotWritableException

Devralınan özellikler ve yöntemler

Aşağıdaki özellikler IMessage arabiriminden edinilir:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSReteslim, JMSReplyTo, JMSTimestamp, JMSType, Özellikler

Aşağıdaki yöntemler IMessage arabiriminden edinilir:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden edinilir:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IPropertyContext

IPropertyContext , özellikleri alan ve ayarlanan yöntemleri içeren soyut bir üst sınıftır. Bu yöntemler diğer sınıflar tarafından edinilir.

Kalıtım sıradüzeni:

Yok

yöntemler

GetBooleanÖzelliği-Boole Özelliğini Al

Arabirim:

```
Boolean GetBooleanProperty(String property_name);
```

Belirtilen ada sahip Boole özelliğinin değerini alın.

Parametreler:

özellik_adi (giriş)

Özelliğin adını saran bir String nesnesi.

Dönüşler:

Özelliğin değeri.

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

GetByteÖzelliği-Byte Özelliğini Al

Arabirim:

```
Byte GetByteProperty(String property_name) ;  
Int16 GetSignedByteProperty(String property_name) ;
```

Ada göre tanımlanan bayt özelliğinin değerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Özelliğın deęeri.

İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

GetBytesÖzellięi-Bayt Dizisi Özellięini Al

Arabirim:

```
Byte[] GetBytesProperty(String property_name) ;
```

Adla tanıtılan bayt dizisi özelliğinin deęerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Dizideki bayt sayısı.

İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

GetCharÖzellięi-Karakter Özellięini Al

Arabirim:

```
Char GetCharProperty(String property_name) ;
```

Adla tanıtılan 2 baytlık karakter özelliğinin deęerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Özelliğın deęeri.

İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

GetDoubleÖzellięi-Çift Duyarlıklı Kayan Nokta Özellięini Al

Arabirim:

```
Double GetDoubleProperty(String property_name) ;
```

Adla tanıtılan çift duyarlıklı kayan nokta özelliğinin deęerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Özelliğın deęeri.

İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

GetFloatÖzellięi-Kayan Nokta Özellięini Al

Arabirim:

```
Single GetFloatProperty(String property_name) ;
```

Adla tanıtılan kayan nokta özelliğinin deęerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Özelliğın deęeri.

İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

GetIntÖzellięi- GetIntÖzellięi

Arabirim:

```
Int32 GetIntProperty(String property_name) ;
```

Adla tanıtılan tamsayı özelliğinin deęerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Özelliğın deęeri.

İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

GetLongÖzellięi-Uzun Tamsayı Özellięini Al

Arabirim:

```
Int64 GetLongProperty(String property_name) ;
```

Adla tanıtılan uzun tamsayı özelliğinin deęerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Özelliğın deęeri.

İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSException

GetObjectÖzellięi-Nesne Özellięini Al

Arabirim:

```
Object GetObjectProperty( String property_name) ;
```

Adla tanıtılan özelliğın deęerini ve veri tipini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Aşağıdaki nesne tiplerinden biri olan özelliğın deęeri:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSException

GetShortÖzellięi-Kısa Tamsayı Özellięini Al

Arabirim:

```
Int16 GetShortProperty(String property_name) ;
```

Adla tanıtılan kısa tamsayı özelliğının deęerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Özelliğın deęeri.

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

GetStringÖzelliği- GetStringÖzelliği

Arabirim:

```
String GetStringProperty(String property_name) ;
```

Ad ile tanıtilan dizgi özelliğinin değerini alın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

Dönüşler:

Özelliğın değeri olan dizgiyi saran bir String nesnesi. Veri dönüştürme gerekiyorsa, bu değeri, dönüştürmeden sonraki dizedir.

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException

SetBooleanÖzelliği-Boole Özelliğini Ayarla

Arabirim:

```
void SetBooleanProperty( String property_name, Boolean value) ;
```

Adla tanıtilan Boole özelliğinin değerini ayarlayın.

Parametreler:**özellik_adi (giriş)**

Özelliğın adını saran bir String nesnesi.

değer (giriş)

Özelliğın değeri.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

SetByteÖzelliği-Byte Özelliğini Ayarla

Arabirim:

```
void SetByteProperty( String property_name, Byte value) ;  
void SetSignedByteProperty( String property_name, Int16 value) ;
```

Adla tanıtilan byte özelliğinin değerini belirleyin.

Parametreler:**özellik_adi (giriş)**

Özelliğin adını saran bir String nesnesi.

değer (giriş)

Özelliğin değeri.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

SetBytesÖzelliği-Set Byte Array Özelliği

Arabirim:

```
void SetBytesProperty( String property_name, Byte[] value ) ;
```

Ada göre tanımlanan bayt dizisi özelliğinin değerini ayarlayın.

Parametreler:**özellik_adi (giriş)**

Özelliğin adını saran bir String nesnesi.

değer (giriş)

Bir bayt dizisi olan özelliğin değeri.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

SetCharÖzellik-Karakter Özelliğini Ayarla

Arabirim:

```
void SetCharProperty( String property_name, Char value) ;
```

Adla tanımlanan 2 baytlık karakter özelliğinin değerini belirleyin.

Parametreler:**özellik_adi (giriş)**

Özelliğin adını saran bir String nesnesi.

değer (giriş)

Özelliğin değeri.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

SetDoubleÖzelliđi-Çift Duyarlıklı Kayan Nokta Özelliđini Ayarla

Arabirim:

```
void SetDoubleProperty( String property_name, Double value) ;
```

Adla tanıtılan çift duyarlıklı kayan nokta özelliđinin deđerini ayarlayın.

Parametreler:

özelliđ_adi (giriş)

Özelliđin adını saran bir String nesnesi.

deđer (giriş)

Özelliđin deđeri.

Dönüşler:

Geçersiz

İş parçacığı bađlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

SetFloatÖzelliđi-Kayan Nokta özelliđini Ayarla

Arabirim:

```
void SetFloatProperty( String property_name, Single value) ;
```

Adla tanıtılan kayan nokta özelliđinin deđerini ayarlayın.

Parametreler:

özelliđ_adi (giriş)

Özelliđin adını saran bir String nesnesi.

deđer (giriş)

Özelliđin deđeri.

Dönüşler:

Geçersiz

İş parçacığı bađlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

SetIntÖzelliđi-Tamsayı Özelliđi Ayarla

Arabirim:

```
void SetIntProperty( String property_name, Int32 value) ;
```

Adla tanıtılan tamsayı özelliđinin deđerini ayarlayın.

Parametreler:**özellik_adi (giriş)**

Özelliğin adını saran bir String nesnesi.

değer (giriş)

Özelliğin değeri.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

SetLongÖzelliği-Uzun Tamsayı Özelliği Ayarla

Arabirim:

```
void SetLongProperty( String property_name, Int64 value) ;
```

Adla tanıtilan uzun tamsayı özelliğinin değerini ayarlayın.

Parametreler:**özellik_adi (giriş)**

Özelliğin adını saran bir String nesnesi.

değer (giriş)

Özelliğin değeri.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

SetObjectÖzelliği-Nesne Özelliğini Ayarla

Arabirim:

```
void SetObjectProperty( String property_name, Object value) ;
```

Adla tanıtilan bir özelliğin değerini ve veri tipini ayarlayın.

Parametreler:**özellik_adi (giriş)**

Özelliğin adını saran bir String nesnesi.

objectType (giriş)

Aşağıdaki nesne tiplerinden biri olması gereken özelliğin değeri:

Boolean
Byte
Byte []
Char
Double

Single
Int32
Int64
Int16
String

değer (giriş)

Özelliğın bayt dizisi olarak değeri.

uzunluk (giriş)

Dizideki bayt sayısı.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSException
- MessageNotWritableException

SetShortÖzelliği-Kısa Tamsayı Özelliği Ayarla

Arabirim:

```
void SetShortProperty( String property_name, Int16 value) ;
```

Adla tanıtilan kısa tamsayı özelliğinin değeri ayarlayın.

Parametreler:

özellik_adi (giriş)

Özelliğın adını saran bir String nesnesi.

değer (giriş)

Özelliğın değeri.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSException
- MessageNotWritableException

SetStringÖzellik-Dizgi Özelliğini Ayarla

Arabirim:

```
void SetStringProperty( String property_name, String value);
```

Adla tanıtilan dizgi özelliğinin değeri ayarlayın.

Parametreler:

özellik_adi (giriş)

Özelliğın adını saran bir String nesnesi.

değer (giriş)

Özelliğın değeri olan dizgiyi saran bir String nesnesi.

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

Özel durumlar:

- XMSEException
- MessageNotWritableException

IQueueBrowser

Bir uygulama, iletileri kaldırmadan kuyruktaki iletilere göz atmak için bir kuyruk tarayıcısını kullanır.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
System.Collections.IEnumerable
|
+---- IBM.XMS.IQueueBrowser
```

.NET Özellikler

MessageSelector -İleti Seçiciyi Al

Arabirim:

```
String MessageSelector
{
    get;
}
```

Kuyruk tarayıcısı için ileti seçiciyi alın.

İleti seçici, ileti seçici ifadesini saran bir Dizgi nesnesidir. Veri dönüştürme gerekiyorsa, bu değer, dönüştürmeden sonraki ileti seçici ifadesidir. Kuyruk tarayıcısının ileti seçicisi yoksa, yöntem boş bir String nesnesi döndürür.

Özel durumlar:

- XMSEException

Kuyruk-Kuyruğu Al

Arabirim:

```
IDestination Queue
{
    get;
}
```

Kuyruk tarayıcısıyla ilişkili kuyruğu, kuyruğu gösteren bir hedef nesne olarak alın.

Özel durumlar:

- XMSEException

yöntemler

Kuyruk Tarayıcısını Kapat

Arabirim:

```
void Close();
```

Kuyruk tarayıcısını kapatın.

Bir uygulama, önceden kapatılmış bir kuyruk tarayıcısını kapatmayı denerse, çağrı yoksayıdır.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException

GetEnumerator -İletileri Al

Arabirim:

```
IEnumerator GetEnumerator();
```

Kuyruktaki iletilerin bir listesini alın.

Bu yöntem, bir İleti nesnelere listesini çevreleyen bir sıralı değer listeleyici döndürür. İleti nesnelere sırası, iletilerin kuyruktan alınacağı sırayla aynıdır. Uygulama daha sonra her iletiye sırayla göz atmak için numaralandırıcıyı kullanabilir.

Kuyruğa ileti konup kuyruktan kaldırıldığında sıralı değer listeleyici dinamik olarak güncellenir. Uygulamanın kuyruktaki sonraki iletiye göz atmak için `IEnumerator.MoveNext()` komutunu her çağırıldığında, ileti kuyruğun yürürlükteki içeriğini yansıtır.

Bir uygulama kuyruk tarayıcısı için bu yöntemi bir kereden fazla çağırırsa, her çağrı yeni bir numaralandırıcı döndürür. Bu nedenle uygulama, kuyruktaki iletilere göz atmak ve kuyruk içinde birden çok konumu korumak için birden çok numaralandırıcıyı kullanabilir.

Parametreler:

Yok

Dönüşler:

Yineleyici nesnesi.

Özel durumlar:

- XMSEException

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler `IPropertyContext` arabiriminden edinilir:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

İstekte Bulunan

Bir uygulama, istek ileti göndermek için bir istekte bulunan kullanır ve daha sonra yanıtı bekler ve alır.

Kalıtım sıradüzeni:

Yok

Oluşturucular

İstekte Bulunan-İstek Sahibi Oluştur

Arabirim:

```
Requestor(ISession sess, IDestination dest);
```

Bir istekte bulunan oluşturun.

Parametreler:

sys (giriş)

Oturum nesnesi. Oturum işleminden geçmemeli ve aşağıdaki alındı bildirim kiplerinden birine sahip olmalıdır:

```
AcknowledgeMode.AutoAcknowledge  
AcknowledgeMode.DupsOkAcknowledge
```

dest (giriş)

Uygulamanın istek iletileri gönderebildiği hedefi gösteren bir Hedef nesne.

İş parçacığı bağlamı:

İstekçiyle ilişkili oturum

Özel durumlar:

- XMSException

yöntemler

İstekluyu Kapat

Arabirim:

```
void Close();
```

İstekliyi kapatın.

Bir uygulama zaten kapalı olan bir istekçiyi kapatmaya çalışırsa, çağrı yoksayılr.

Not: Bir uygulama istekte bulunan bir oturumu kapattığında, ilişkili oturum da kapanmaz. Bu bakımdan XMS , JMS ile karşılaştırıldığında farklı davranır.

Parametreler:

Yok

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Herhangi Biri

Özel durumlar:

- XMSException

İstek-İstek Yanıtı

Arabirim:

```
IMessage Request(IMessage requestMessage);
```

Bir istek iletisi gönderin ve istek iletisini alan uygulamadan bir yanıt bekleyin ve alın.

Bu yonteme yapilan bir cagrı, bir yanıt alınincaya ya da oturum sona erinceye kadar (hangisi daha erken olursa) bu yontemi engeller.

Parametreler:

requestMessage (giriş)

İstek iletisini saran İleti nesnesi.

Dönüşler:

Yanıt iletisini saran İleti nesnesine ilişkin gösterge.

İş parçacığı bağlamı:

İstekçiyle ilişkili oturum

Özel durumlar:

- XMSEException

ResourceAllocationKural Dışı Durumu

XMS , XMS bir yontemin gerektirdiği kaynakları ayıramazsa bu kural dışı durumu yayınlar.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.ResourceAllocationException
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

SecurityException

XMS , bir uygulamanın kimliğini doğrulamak için sağlanan kullanıcı kimliği ve parola reddedilirse bu kural dışı durumu verir. XMS ayrıca, bir yetki denetimi başarısız olursa ve bir yontemin tamamlanmasını engellerse de bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.SecurityException
```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

ISession (Oturum)

Oturum, ileti göndermek ve almak için tek iş parçacıklı bir bağlamdır.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.ISession
```

Oturum nesnesinin XMS tanımlı özelliklerinin bir listesi için bkz. [“Oturum Özellikleri” sayfa 2013.](#)

.NET Özellikler

AcknowledgeMode -Alındı Bildirimi Kipini Al

Arabirim:

```
AcknowledgeMode AcknowledgeMode
{
    get;
}
```

Oturuma ilişkin alındı bildirimi kipini alın.

Alındı bildirimi kipi, oturum yaratıldığında belirtilir.

Oturumun işlemden geçirilmemesi durumunda, alındı bildirimi kipi aşağıdaki değerlerden biridir:

```
AcknowledgeMode.AutoAcknowledge
AcknowledgeMode.ClientAcknowledge
AcknowledgeMode.DupsOkAcknowledge
```

Alındı bildirimi kipleri hakkında daha fazla bilgi için bkz. [İleti alındı bildirimi.](#)

İşlem uygulanan bir oturumun alındı bildirimi kipi yok. Oturum işlemden geçtiyse, yöntem `AcknowledgeMode` ni döndürür. `SessionTransacted`.

Özel durumlar:

- `XMSEException`

İşlemden Geçildi-İşlemden Geçilip Geçilmeyeceğini Belirle

Arabirim:

```
Boolean Transacted
{
    get;
}
```

Oturumun hareket edip etmediğini belirleyin.

Belirtilen işlem:

- Oturum işlemden geçtiyse doğru (True) değerini alır.
- Oturum işlem yapılmazsa false (yanlış) değerini alır.

Bir aracıya gerçek zamanlı bağlantı için yöntem her zaman False değerini döndürür.

Özel durumlar:

- `XMSEException`

yöntemler

Oturumu Kapat

Arabirim:

```
void Close();
```

Oturumu kapatın. Oturum işlemden geçtiyse, devam eden tüm hareketler geriye işlenir.

Bir uygulama, önceden kapatılmış bir oturumu kapatmayı denerse, çağrı yoksayıdır.

Parametreler:

Yok

Dönüşler:

Geçersiz

İş parçacığı bağlamı:

Herhangi Biri

Özel durumlar:

- XMSException

Kesinleştir-Kesinleştir

Arabirim:

```
void Commit();
```

Yürürlükteki harekette işlenen tüm iletileri kesinleştir.

Oturum, işlem uygulanan bir oturum olmalıdır.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- IllegalStateKural Dışı Durumu
- TransactionRolledBackException

CreateBrowser -Kuyruk Tarayıcısı Yarat

Arabirim:

```
IQueueBrowser CreateBrowser(IDestination queue) ;
```

Belirtilen kuyruk için bir kuyruk tarayıcısı yaratın.

Parametreler:**kuyruk (giriş)**

Kuyruğu gösteren bir Hedef nesne.

Dönüşler:

QueueBrowser nesnesi.

Özel durumlar:

- XMSException
- InvalidDestinationKural Dışı Durumu

CreateBrowser -Kuyruk Tarayıcısı Yarat (ileti seçici ile)

Arabirim:

```
IQueueBrowser CreateBrowser(IDestination queue, String selector) ;
```

İleti seçici kullanarak belirtilen kuyruk için bir kuyruk tarayıcısı yaratın.

Parametreler:**kuyruk (giriş)**

Kuyruğu gösteren bir Hedef nesne.

seçici (giriş)

Bir ileti seçici ifadesini saran String nesnesi. Kuyruk tarayıcısına yalnızca, ileti seçici ifadesiyle eşleşen özelliklere sahip iletiler teslim edilir.

Boş değerli bir String nesnesi, kuyruk tarayıcısı için ileti seçici olmadığı anlamına gelir.

Dönüşler:

QueueBrowser nesnesi.

Özel durumlar:

- XMSException
- InvalidDestinationKural Dışı Durumu
- InvalidSelectorKural Dışı Durumu

CreateBytesİletisi-Bayt İletisi Yarat

Arabirim:

```
IBytesMessage CreateBytesMessage();
```

Bir bayt iletisi oluşturun.

Parametreler:

Yok

Dönüşler:

BytesMessage nesnesi.

Özel durumlar:

- XMSException
- IllegalStateKural Dışı Durumu (Oturum kapatıldı)

CreateConsumer -Tüketici Yarat

Arabirim:

```
IMessageConsumer CreateConsumer(IDestination dest) ;
```

Belirtilen hedef için bir ileti tüketicisi yaratın.

Parametreler:**dest (giriş)**

Hedef nesne.

Dönüşler:

MessageConsumer nesnesi.

Özel durumlar:

- XMSException
- InvalidDestinationKural Dışı Durumu

CreateConsumer -Tüketici Yarat (iletici seçici ile)

Arabirim:

```
IMessageConsumer CreateConsumer(IDestination dest,  
String selector) ;
```

İleti seçici kullanarak, belirtilen hedef için bir ileti tüketicisi yaratın.

Parametreler:

dest (giriş)

Hedef nesne.

seçici (giriş)

Bir ileti seçici ifadesini saran String nesnesi. Yalnızca ileti seçici ifadesiyle eşleşen özelliklere sahip iletiler ileti tüketicisine teslim edilir.

Boş değerli bir Dizgi nesnesi, ileti tüketicisi için ileti seçici olmadığı anlamına gelir.

Dönüşler:

MessageConsumer nesnesi.

Özel durumlar:

- XMSException
- InvalidDestinationKural Dışı Durumu
- InvalidSelectorKural Dışı Durumu

CreateConsumer -Tüketici Yarat (iletici seçici ve yerel ileti işaretiyle)

Arabirim:

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector,  
                                Boolean noLocal) ;
```

Bir ileti seçici kullanarak belirtilen hedef için bir ileti tüketicisi yaratın ve hedef bir konuysa, ileti tüketicisinin kendi bağlantısıyla yayınlanan iletileri alıp almayacağını belirtin.

Parametreler:

dest (giriş)

Hedef nesne.

seçici (giriş)

Bir ileti seçici ifadesini saran String nesnesi. Yalnızca ileti seçici ifadesiyle eşleşen özelliklere sahip iletiler ileti tüketicisine teslim edilir.

Boş değerli bir Dizgi nesnesi, ileti tüketicisi için ileti seçici olmadığı anlamına gelir.

noLocal (giriş)

True değeri, ileti tüketicisinin kendi bağlantısıyla yayınlanan iletileri almadığı anlamına gelir.

False değeri, ileti tüketicisinin kendi bağlantısı tarafından yayınlanan iletileri aldığı anlamına gelir.

Varsayılan değer False değeridir.

Dönüşler:

MessageConsumer nesnesi.

Özel durumlar:

- XMSException
- InvalidDestinationKural Dışı Durumu
- InvalidSelectorKural Dışı Durumu

CreateDurableAbone-Dayanıklı Abone Yarat

Arabirim:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                         String subscription) ;
```

Belirtilen konu için sürekli bir abone yaratın.

Bu yöntem, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

Sürekli aboneler hakkında daha fazla bilgi için bkz. [Durable subscribers](#).

Parametreler:

dest (giriş)

Konuyu gösteren bir Hedef nesnesi. Konu geçici bir konu olmamalıdır.

abonelik (giriş)

Sürekli aboneliği tanıtan bir adı saran bir String nesnesi. Ad, bağlantıya ilişkin istemci tanıtıcısı içinde benzersiz olmalıdır.

Dönüşler:

Sürekli aboneyi temsil eden MessageConsumer nesnesi.

Özel durumlar:

- XMSException
- InvalidDestinationKural Dışı Durumu

CreateDurableAbone-Dayanımlı Abone Yarat (ileti seçici ve yerel ileti işaretiyle)

Arabirim:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                         String subscription,  
                                         String selector,  
                                         Boolean noLocal) ;
```

Bir ileti seçici kullanarak ve sürekli abonenin kendi bağlantısıyla yayınlanan iletileri alıp almayacağını belirterek, belirtilen konu için sürekli bir abone yaratın.

Bu yöntem, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

Sürekli aboneler hakkında daha fazla bilgi için bkz. [Durable subscribers](#).

Parametreler:

dest (giriş)

Konuyu gösteren bir Hedef nesnesi. Konu geçici bir konu olmamalıdır.

abonelik (giriş)

Sürekli aboneliği tanıtan bir adı saran bir String nesnesi. Ad, bağlantıya ilişkin istemci tanıtıcısı içinde benzersiz olmalıdır.

seçici (giriş)

Bir ileti seçici ifadesini saran String nesnesi. Kalıcı aboneye yalnızca, ileti seçici ifadesiyle eşleşen özelliklere sahip iletiler teslim edilir.

Boş değerli bir Dizgi nesnesi, sürekli abone için ileti seçici olmadığı anlamına gelir.

noLocal (giriş)

True değeri, sürekli abonenin kendi bağlantısı tarafından yayınlanan iletileri almadığı anlamına gelir. False değeri, sürekli abonenin kendi bağlantısı tarafından yayınlanan iletileri aldığı anlamına gelir. Varsayılan değer False değeridir.

Dönüşler:

Sürekli aboneyi temsil eden MessageConsumer nesnesi.

Özel durumlar:

- XMSException
- InvalidDestinationKural Dışı Durumu
- InvalidSelectorKural Dışı Durumu

CreateMapİletisi-Eşlem İletisi Yarat

Arabirim:

```
IMapMessage CreateMapMessage();
```

Bir harita iletisi oluşturun.

Parametreler:

Yok

Dönüşler:

MapMessage nesnesi.

Özel durumlar:

- XMSEException
- IllegalStateKural Dışı Durumu (Oturum kapatıldı)

CreateMessage -İleti Yarat

Arabirim:

```
IMessage CreateMessage();
```

Gövdesi olmayan bir ileti oluşturun.

Parametreler:

Yok

Dönüşler:

İleti nesnesi.

Özel durumlar:

- XMSEException
- IllegalStateKural Dışı Durumu (Oturum kapatıldı)

CreateObjectİletisi-Nesne İletisi Yarat

Arabirim:

```
IObjectMessage CreateObjectMessage();
```

Bir nesne iletisi oluşturun.

Parametreler:

Yok

Dönüşler:

ObjectMessage nesnesi.

Özel durumlar:

- XMSEException
- IllegalStateKural Dışı Durumu (Oturum kapatıldı)

CreateProducer -Üretici Yarat

Arabirim:

```
IMessageProducer CreateProducer(IDestination dest) ;
```

Belirtilen hedefe ileti göndermek için bir ileti üreticisi yaratın.

Parametreler:**dest (giriş)**

Hedef nesne.

Boş değerli bir Hedef nesne belirtirseniz, ileti üreticisi hedef olmadan yaratılır. Bu durumda, uygulamanın ileti göndermek için ileti üreticisi her kullanımında bir hedef belirtmesi gerekir.

Dönüşler:

MessageProducer nesnesi.

Özel durumlar:

- XMSEException
- InvalidDestinationKural Dışı Durumu

CreateQueue -Kuyruk Yarat

Arabirim:

```
IDestination CreateQueue(String queue) ;
```

İleti alışverişi sunucusundaki bir kuyruğu gösterecek bir Hedef nesnesi yaratın.

Bu yöntem, ileti alışverişi sunucusunda kuyruğu yaratmaz. Bir uygulamanın bu yöntemi çağırabilmesi için önce kuyruğu yaratmanız gerekir.

Parametreler:**kuyruk (giriş)**

Kuyruğun adını kapsülleyen ya da kuyruğu tanıtan bir URI ' yi (birörnek kaynak tanıtıcısı) saran bir String nesnesi.

Dönüşler:

Kuyruğu gösteren Hedef nesne.

Özel durumlar:

- XMSEException

CreateStreamİletisi-Akış İletisi Yarat

Arabirim:

```
IStreamMessage CreateStreamMessage();
```

Akış iletisi yaratın.

Parametreler:

Yok

Dönüşler:

StreamMessage nesnesi.

Özel durumlar:

- XMSEException
- XMS_ILLEGAL_STATE_EXCEPTION

CreateTemporaryKuyruk-Geçici Kuyruk Yarat

Arabirim:

```
IDestination CreateTemporaryQueue() ;
```

Geçici bir kuyruk oluşturun.

Geçici kuyruğun kapsamı bağlantıdır. Geçici kuyruğu yalnızca, bağlantı tarafından yaratılan oturumlar kullanabilir.

Geçici kuyruk, belirtik olarak silininceye ya da bağlantı sona erinceye kadar (hangisi daha erken ise) kalır.

Geçici kuyruklar hakkında daha fazla bilgi için bkz. [Geçici hedefler](#).

Parametreler:

Yok

Dönüşler:

Geçici kuyruğu gösteren Hedef nesne.

Özel durumlar:

- XMSEException

CreateTemporaryKonu-Geçici Konu Yarat

Arabirim:

```
IDestination CreateTemporaryTopic() ;
```

Geçici bir konu oluşturun.

Geçici konunun kapsamı bağlantıdır. Geçici konuyu yalnızca bağlantı tarafından yaratılan oturumlar kullanabilir.

Geçici konu belirtik olarak silininceye ya da bağlantı sona erinceye kadar (hangisi daha önceyse) kalır.

Geçici konularla ilgili daha fazla bilgi için [Geçici hedefler](#) başlıklı konuya bakın.

Parametreler:

Yok

Dönüşler:

Geçici konuyu gösteren Hedef nesnesi.

Özel durumlar:

- XMSEException

CreateTextİletisi-Metin İletisi Yarat

Arabirim:

```
ITextMessage CreateTextMessage();
```

Boş gövdeli bir metin iletisi oluşturun.

Parametreler:

Yok

Dönüşler:

TextMessage nesnesi.

Özel durumlar:

- XMSEException

CreateTextİletisi-Metin İletisi Yarat (ilk kullanıma hazırlandı)

Arabirim:

```
ITextMessage CreateTextMessage(String initialValue);
```

Gövdesi belirtilen metinle kullanıma hazırlanan bir metin iletisi oluşturun.

Parametreler:**initialValue (giriş)**

Metin iletisinin gövdesini kullanıma hazırlamak için metni içeren bir String nesnesi.

Yok

Dönüşler:

TextMessage nesnesi.

Özel durumlar:

- XMSEException

CreateTopic -Konu Yarat

Arabirim:

```
IDestination CreateTopic(String topic) ;
```

Bir konuyu temsil edecek bir Hedef nesne oluşturun.

Parametreler:**konu (giriş)**

Konunun adını içeren ya da konuyu tanıtan bir URI ' yi (tekdüzen kaynak tanıtıcısı) kapsülleyen bir String nesnesi.

Dönüşler:

Konuyu gösteren Hedef nesnesi.

Özel durumlar:

- XMSEException

Kurtar-Kurtar

Arabirim:

```
void Recover();
```

Oturumu kurtarın. İleti teslimi durdurulur ve daha sonra, en eski alınmamış iletiyle yeniden başlatılır.

Oturum, işlem uygulanan bir oturum olmamalıdır.

Bir oturumu kurtarma hakkında daha fazla bilgi için bkz. [İleti alındı bildirimi](#).

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- IllegalStateKural Dışı Durumu

Geri Alma-Geri Alma

Arabirim:

```
void Rollback();
```

Yürürlükteki harekette işlenen tüm iletileri geri alır.

Oturum, işlem uygulanan bir oturum olmalıdır.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- IllegalStateKural Dışı Durumu

Aboneliği Kaldır-Abonelikten çık

Arabirim:

```
void Unsubscribe(String subscription);
```

Sürekli aboneliğin silinmesi İleti alışverişi sunucusu, bakımını yaptığı sürekli aboneliğin kaydını siler ve kalıcı aboneye başka ileti göndermez.

Bir uygulama, aşağıdaki koşullardan herhangi birinde sürekli bir aboneliği silemez:

- Sürekli abonelik için etkin bir ileti tüketicisi varken
- Tüketilen bir ileti, bekleyen bir hareketin parçasıyken
- Tüketilen bir ileti onaylanmamışken

Bu yöntem, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

Parametreler:**abonelik (giriş)**

Sürekli aboneliği tanıtan adı saran bir String nesnesi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- InvalidDestinationKural Dışı Durumu
- IllegalStateKural Dışı Durumu

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden edinilir:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IStreamMessage

Akış iletisi, gövdesi her değer ilişkili bir veri tipine sahip olduğu bir değer akımından oluşan bir iletidir. Gövdenin içeriği sıralı olarak yazılır ve okunur.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IStreamMessage
```

Bir uygulama ileti akışından bir değer okuduğunda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Bu örtük dönüştürme biçimiyle ilgili daha fazla bilgi için bkz. [XMS iletisinin gövdesi](#).

yöntemler

ReadBoolean -Okuma Boole Değeri

Arabirim:

```
Boolean ReadBoolean();
```

İleti akışından bir Boole değeri okuyun.

Parametreler:

Yok

Dönüşler:

Okunan Boole değeri.

Özel durumlar:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadByte -Okuma Baytı

Arabirim:

```
Int16 ReadSignedByte();  
Byte ReadByte();
```

İleti akışından 8 bitlik imzalı bir tamsayı okuyun.

Parametreler:

Yok

Dönüşler:

Okunan bayt.

Özel durumlar:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadBytes -Okuma Baytları

Arabirim:

```
Int32 ReadBytes(Byte[] array);
```

İleti akımından bir bayt dizisini okuyun.

Parametreler:

dizi (giriş)

Okunan bayt dizisini ve arabelleğin bayt cinsinden uzunluğunu içeren arabellek.

Dizideki bayt sayısı arabelleğin uzunluğundan az ya da ona eşitse, tüm dizi arabelleğe okunur. Dizideki bayt sayısı arabellek uzunluğundan fazlaysa, arabellek dizinin bir parçasıyla doldurulur ve bir iç imleç okunacak sonraki baytın konumunu işaretler. Sonraki readBytes() çağırısı, imlecin yürürlükteki konumundan başlayarak diziden bayt okur.

Girişte boş değerli bir gösterge belirtirseniz, çağrı bayt dizisini okumadan atlar.

Dönüşler:

Arabelleğe okunan bayt sayısı. Arabellek kısmen doldurulmuşsa, değer arabelleğin uzunluğundan azdır ve okunacak dizide başka bayt kalmadığını gösterir. Aramadan önce diziden okunacak bayt kalmazsa, değer `XMSC_END_OF_BYTEARRAY` olur.

Girişte boş değerli gösterge belirtirseniz, yöntem değer döndürmez.

Özel durumlar:

- `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadChar -Okuma karakteri

Arabirim:

```
Char ReadChar();
```

İleti akımından 2 baytlık bir karakter okuyun.

Parametreler:

Yok

Dönüşler:

Okunan karakter.

Özel durumlar:

- `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadDouble -Okuma Çift Duyarlıklı Kayan Noktalı Sayı

Arabirim:

```
Double ReadDouble();
```

İleti akımından 8 baytlık çift duyarlıklı kayan noktalı bir sayı okuyun.

Parametreler:

Yok

Dönüşler:

Okunan çift duyarlıklı kayan noktalı sayı.

Özel durumlar:

- `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadFloat -Kayan Nokta Numarasını Oku

Arabirim:

```
Single ReadFloat();
```

İleti akışından 4 baytlık bir kayan noktalı sayı okuyun.

Parametreler:

Yok

Dönüşler:

Okunan kayar noktalı sayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadInt -Okuma Tamsayı

Arabirim:

```
Int32 ReadInt();
```

İleti akımından 32 bitlik imzalı bir tamsayı okuyun.

Parametreler:

Yok

Dönüşler:

Okunan tamsayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadLong -Okuma Uzun Tamsayı

Arabirim:

```
Int64 ReadLong();
```

İleti akımından imzalı 64 bitlik bir tamsayı okuyun.

Parametreler:

Yok

Dönüşler:

Okunan uzun tamsayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadObject -Nesneyi Oku

Arabirim:

```
Object ReadObject();
```

İleti akımından bir değeri okuyun ve veri tipini döndürün.

Parametreler:

Yok

Dönüşler:

Aşağıdaki nesne tiplerinden biri olan değer:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

Özel durumlar:

XMSEException

ReadShort -Read Short Integer (Okuma Kısa Tamsayı)

Arabirim:

```
Int16 ReadShort();
```

İleti akışından 16 bitlik imzalı bir tamsayı okuyun.

Parametreler:

Yok

Dönüşler:

Okunan kısa tamsayı.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadString -Okuma Dizisi

Arabirim:

```
String ReadString();
```

İleti akışından bir dizgi okuyun. Gerekirse, XMS dizedeki karakterleri yerel kod sayfasına dönüştürür.

Parametreler:

Yok

Dönüşler:

Okunan dizgiyi saran bir String nesnesi. Veri dönüştürme gerekliyse, bu, dönüştürmeden sonraki dizedir.

Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageEOFException

Sıfırla-Sıfırla

Arabirim:

```
void Reset();
```

İletin gövdesini salt okunur kipine alın ve imleci ileti akışının başına konumlandırın.

Parametreler:

Yok

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageNotReadableException
- MessageEOFException

WriteBoolean -Yazma Boole Değeri

Arabirim:

```
void WriteBoolean(Boolean value);
```

İleti akımına bir Boole değeri yazın.

Parametreler:

değer (giriş)

Yazılacak Boole değeri.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageNotWritableException

WriteByte -Yazma Baytı

Arabirim:

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

İleti akımına bir bayt yazın.

Parametreler:

değer (giriş)

Yazılacak bayt.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageNotWritableException

WriteBytes -Yazma Baytları

Arabirim:

```
void WriteBytes(Byte[] value);
```

İleti akımına bir bayt dizisi yazın.

Parametreler:

değer (giriş)

Yazılacak bayt dizisi.

uzunluk (giriş)

Dizideki bayt sayısı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteChar -Yazma karakteri

Arabirim:

```
void WriteChar(Char value);
```

İleti akımına 2 bayt, önce yüksek sıralı bayt olarak bir karakter yazın.

Parametreler:

değer (giriş)

Yazılacak karakter.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteDouble -Write Double Precision Floating Point Number (Çift Duyarlıklı Kayan Noktalı Sayı Yaz)

Arabirim:

```
void WriteDouble(Double value);
```

Çift duyarlıklı bir kayan noktalı sayıyı uzun bir tamsayıya dönüştürün ve uzun tamsayıyı önce 8 byte, yüksek sıralı byte olarak ileti akımına yazın.

Parametreler:

değer (giriş)

Yazılacak çift duyarlıklı kayan noktalı sayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteFloat -Yazma Kayan Noktası Numarası

Arabirim:

```
void WriteFloat(Single value);
```

Kayan noktalı bir sayıyı tamsayıya dönüştürün ve tamsayıyı ileti akımına 4 bayt, yüksek sıralı bayt olarak yazın.

Parametreler:

değer (giriş)

Yazılacak kayar noktalı sayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteInt -Yazma Tamsayı

Arabirim:

```
void WriteInt(Int32 value);
```

İleti akımına önce 4 bayt, yüksek sıralı bayt olarak bir tamsayı yazın.

Parametreler:

değer (giriş)

Yazılacak tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteLong -Yazma Uzun Tamsayı

Arabirim:

```
void WriteLong(Int64 value);
```

İleti akımına önce 8 bayt, yüksek sıralı bayt olarak uzun bir tamsayı yazın.

Parametreler:

değer (giriş)

Yazılacak uzun tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSEException
- MessageNotWritableException

WriteObject -Yazma Nesnesi

Arabirim:

```
void WriteObject(Object value);
```

Belirtilen veri tipine sahip bir değeri ileti akımına yazın.

Parametreler:

objectType (giriş)

Aşağıdaki nesne tiplerinden biri olması gereken değer:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

değer (giriş)

Yazılacak değeri içeren bir bayt dizisi.

uzunluk (giriş)

Dizideki bayt sayısı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException

WriteShort -Write Short Integer (Yazma Kısa Tamsayı)

Arabirim:

```
void WriteShort(Int16 value);
```

İleti akımına 2 bayt, önce yüksek sıralı bayt olarak kısa bir tamsayı yazın.

Parametreler:

değer (giriş)

Yazılacak kısa tamsayı.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageNotWritableException

WriteString -Dizgiyi Yaz

Arabirim:

```
void WriteString(String value);
```

İleti akımına bir dizgi yazın.

Parametreler:

değer (giriş)

Yazılacak dizgiyi saran bir String nesnesi.

Dönüşler:

Geçersiz

Özel durumlar:

- XMSException
- MessageNotWritableException

Devralınan özellikler ve yöntemler

Aşağıdaki özellikler IMessage arabiriminden edinilir:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSReteslim, JMSReplyTo, JMSTimestamp, JMSType, Özellikler

Aşağıdaki yöntemler IMessage arabiriminden edinilir:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden edinilir:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

ITextMessage

Metin iletisi, gövdesi bir dizgiden oluşan bir iletidir.

Kalıtım sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
|
+---- IBM.XMS.ITextMessage
```

.NET Özellikler

Metin-Metni Al ve Ayarla

Arabirim:

```
String Text
{
    get;
    set;
}
```

Metin iletisinin gövdesini oluşturan dizgiyi alın ve ayarlayın.

Gerekirse, XMS dizedeki karakterleri yerel kod sayfasına dönüştürür.

Özel durumlar:

- XMSException
- MessageNotReadableException

- MessageNotWritableException
- MessageEOFException

Devralınan özellikler ve yöntemler

Aşağıdaki özellikler IMessage arabiriminden edinilir:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSReteslim, JMSReplyTo, JMSTimestamp, JMSType, Özellikler

Aşağıdaki yöntemler IMessage arabiriminden edinilir:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden edinilir:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

TransactionInProgressException

XMS , bir uygulama devam eden bir hareket nedeniyle geçerli olmayan bir işlem isterse bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```

IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.TransactionInProgressException

```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler XMSEException arabiriminden edinilir:

GetErrorCode, GetLinkedException

TransactionRolledBackException

XMS , bir uygulama yürürlükteki hareketi kesinleştirmek için Session.commit() komutunu çağırırsa, ancak daha sonra hareket geriye işlenirse bu kural dışı durumu verir.

Kalıtım sıradüzeni:

```

IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.TransactionRolledBackException

```

Devralınan özellikler ve yöntemler

Aşağıdaki yöntemler XMSEException arabiriminden edinilir:

GetErrorCode, GetLinkedException

XMSEException

XMS , .NET yöntemine yapılan bir çağrıyı işlerken bir hata saptarsa, XMS bir kural dışı durum yayınlar. Kural dışı durum, hatayla ilgili bilgileri içeren bir nesnedir.

Kalıtım sıradüzeni:

```
System.Exception
|
+----IBM.XMS.XMSEException
```

Farklı XMS kural dışı durumu tipleri var ve bir XMSEException nesnesi yalnızca bir kural dışı durum tipi. Ancak XMSEException sınıfı, diğer XMS kural dışı durum sınıflarının bir üst sınıfıdır. XMS , diğer kural dışı durum tiplerinin uygun olmadığı durumlarda bir XMSEException nesnesi verir.

.NET Özellikler

ErrorCode -Hata Kodunu Al

Arabirim:

```
public String ErrorCode
{
    get {return errorCode_;}
}
```

Hata kodunu alın.

Özel durumlar:

- XMSEException

LinkedException -Bağlantılı Kural Dışı Durumu Al

Arabirim:

```
public Exception LinkedException
{
    get { return linkedException_;}
    set { linkedException_ = value;}
}
```

Kural dışı durumlar zincirinde sonraki kural dışı durumu alın.

Zincirde başka kural dışı durum yoksa yöntem boş değer döndürür.

Özel durumlar:

- XMSEException

XMSFactoryFactory

Bir uygulama yönetilen nesnelere kullanmıyorsa, bağlantı üreticileri, kuyuklar ve konular yaratmak için bu sınıfı kullanın.

Kalıtım sıradüzeni:

Yok

.NET Özellikler

Meta Veri-Meta verileri al

Arabirim:

```
IConnectionMetaData MetaData
```

XMSFactoryFactory nesnesinin bağlantı tipine uygun meta verileri alın.

Özel durumlar:

Yok

yöntemler

CreateConnectionÜreticisi-Bağlantı Üreticisi Yarat

Arabirim:

```
ConnectionFactory CreateConnectionFactory();
```

Bildirilen tipte bir ConnectionFactory nesnesi yaratın.

Parametreler:

Yok

Dönüşler:

ConnectionFactory nesnesi.

Özel durumlar:

- XMSEException

CreateQueue -Kuyruk Yarat

Arabirim:

```
IDestination CreateQueue(String name);
```

İleti alışverişi sunucusundaki bir kuyruğu gösterecek bir Hedef nesnesi yaratın.

Bu yöntem, ileti alışverişi sunucusunda kuyruğu yaratmaz. Bir uygulamanın bu yöntemi çağırabilmesi için önce kuyruğu yaratmanız gerekir.

Parametreler:**ad (giriş)**

Kuyruğun adını kapsülleyen ya da kuyruğu tanıtan bir URI ' yi (birörnek kaynak tanıtıcısı) saran bir String nesnesi.

Dönüşler:

Kuyruğu gösteren Hedef nesne.

Özel durumlar:

- XMSEException

CreateTopic -Konu Yarat

Arabirim:

```
IDestination CreateTopic(String name);
```

Bir konuyu temsil edecek bir Hedef nesne oluşturun.

Parametreler:**ad (giriş)**

Konunun adını içeren ya da konuyu tanıtan bir URI ' yi (tekdüzen kaynak tanıtıcısı) kapsülleyen bir String nesnesi.

Dönüşler:

Konuyu gösteren Hedef nesnesi.

Özel durumlar:

- XMSEException

GetInstance - *XMSFactoryFactory* eşgörünümünü alın

Arabirim:

```
static XMSFactoryFactory GetInstance(int connectionType);
```

Bir *XMSFactoryFactory* eşgörünümü yaratın. XMS uygulaması, gerekli protokol tipine uygun bir *ConnectionFactory* nesnesine başvuru almak için bir *XMSFactoryFactory* nesnesi kullanır. Bu *ConnectionFactory* nesnesi daha sonra yalnızca o protokol tipi için bağlantı üretebilir.

Parametreler:

connectionType (giriş)

ConnectionFactory nesnesinin bağlantı ürettiği bağlantı tipi:

- XMSC.CT_WPM
- XMSC.CT_RTT
- XMSC.CT_WMQ

Dönüşler:

Bildirilen bağlantı tipine adanmış *XMSFactoryFactory* nesnesi.

Özel durumlar:

- *NotSupportedKural Dışı Durumu*

XMS nesnelерinin özellikleri

Bu kısım, *XMStarafından* tanımlanan nesne özelliklerini belgeler.

Bu kısım, aşağıdaki nesne tipleriyle ilgili bilgi içerir:

- [“Bağlantı Özellikleri” sayfa 2001](#)
- [“ConnectionFactory Özellikleri” sayfa 2001](#)
- [“ConnectionMetaVerilerinin Özellikleri” sayfa 2005](#)
- [“Hedef Özellikleri” sayfa 2006](#)
- [“InitialContext özellikleri” sayfa 2007](#)
- [“İleti Özellikleri” sayfa 2008](#)
- [“MessageConsumer Özellikleri” sayfa 2012](#)
- [“MessageProducer Özellikleri” sayfa 2013](#)
- [“Oturum Özellikleri” sayfa 2013](#)

Her nesne tipinin tanımı, belirtilen tipte bir nesnenin özelliklerini listeler ve her özelliğin kısa bir açıklamasını sağlar.

Bu kısım, her özelliğin tanımlamasını da sağlar (bkz. [“Özellik Tanımları” sayfa 2013](#)).

Bir uygulama bu kısımda açıklanan nesnelere ilişkin kendi özelliklerini tanımlıyorsa, bu bir hataya neden olmaz, ancak beklenmedik sonuçlara neden olabilir.

Not: Bu bölümdeki özellik adları ve değerleri, C ve C++ için kullanılan form olan *XMSC.NAME* biçiminde gösterilir. Ancak, .NET’inde, özellik adının biçimi *XMSC.NAME* ya da *XMSC_NAME* olabilir; bu biçimi nasıl kullandığınıza bağlı olarak:

- Bir özellik belirtiyorsanız, özellik adı aşağıdaki örnekte gösterildiği gibi *XMSC.NAME* biçiminde olmalıdır:

```
cf.SetStringProperty(XMSC.WMQ_CHANNEL, "DOTNET.SVRCONN");
```

- Bir dizgi belirtiyorsanız, özellik adı aşağıdaki örnekte gösterildiği gibi *XMSC_NAME* biçiminde olmalıdır:

```
cf.SetStringProperty("XMSC_WMQ_CHANNEL", "DOTNET.SVRCONN");
```


.NETiçinde özellik adları ve değerleri, XMSC sınıfında sabitler olarak sağlanır. Bu sabitler dizgileri tanımlar ve herhangi bir XMS .NET uygulaması tarafından kullanılır. Bu önceden tanımlanmış değişmezleri kullanıyorsanız, özellik adları ve değerleri XMSC.NAMEbiçimindedir; bu nedenle, örneğin, XMSC.USERIDkullanıcı kimliği.

Veri tipleri C/C + + için kullanılan formda da bulunur. .NET ile ilgili değerleri .NET için veri tipleri içinde bulabilirsiniz.

Bağlantı Özellikleri

Daha ayrıntılı başvuru bilgileri bağlantılarıyla birlikte Connection nesnesinin özelliklerine genel bakış.

<i>Çizelge 872. Bağlantı Özellikleri</i>	
Özelliğin adı	Açıklama
“XMSC_WMQ_RESOLVED_QUEUE_MANAGER” sayfa 2046	Bu özellik, bağlı olduğu kuyruk yöneticisinin adını almak için kullanılır.
“XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID” sayfa 2046	Bu özelliğe, bağlantıdan sonra kuyruk yöneticisinin tanıtıcısı yerleştirilir.
XMSC_WPM_CONNECTION_PROTOCOL	İleti alışverişi altyapısına bağlantı için kullanılan iletişim protokolü. Bu özellik salt okunur.
XMSC_WPM_ANASISTEM_ADI	Uygulamanın bağlı olduğu ileti alışverişi altyapısını içeren sistemin anasistem adı ya da IP adresi. Bu özellik salt okunur.
XMSC_WPM_ME_NAME	Uygulamanın bağlı olduğu ileti alışverişi altyapısının adı. Bu özellik salt okunur.
XMSC_WPM_PORT	Uygulamanın bağlı olduğu ileti alışverişi altyapısı tarafından dinlenen kapının sayısı. Bu özellik salt okunur.

Bir Connection nesnesi, bağlantıyı oluşturmak için kullanılan bağlantı üreticisinin özelliklerinden türetilen salt okunur özelliklere de sahiptir. Bu özellikler, yalnızca bağlantının yaratıldığı sırada ayarlanan bağlantı üreticisi özelliklerinden değil, ayarlanmayan özelliklerin varsayılan değerlerinden de türetilir. Özellikler yalnızca, uygulamanın bağlı olduğu ileti sistemi sunucusu tipiyle ilgili olanları içerir. Özelliklerin adları, bağlantı üreticisi özelliklerinin adlarıyla aynıdır.

ConnectionFactory Özellikleri

Daha ayrıntılı başvuru bilgileri bağlantılarıyla birlikte ConnectionFactory nesnesinin özelliklerine genel bakış.

<i>Çizelge 873. ConnectionFactory Özellikleri</i>	
Özelliğin adı	Açıklama
“XMSC_ASYNC_ISTISNALAR” sayfa 2022	Bu özellik, XMS 'nin bir ExceptionListener ' i yalnızca bir bağlantı kesildiğinde mi, yoksa bir XMS API çağrısına zamanuyumsuz olarak bir kural dışı durum oluştuğunda mı bilgilendirdiğini belirler. Bu özellik, ExceptionListener kaydı olan bu ConnectionFactory ' den oluşturulan tüm Connections için geçerlidir.
“XMSC_WMQ_BALANCING_APPLICATION_TYPE” sayfa 2031	Dengeleme seçeneği tipi

Çizelge 873. ConnectionFactory Özellikleri (devamı var)	
Özelliğın adı	Açıklama
V 9.3.0 "XMSC_WMQ_BALANCING_OPTIONS" sayfa 2031	Düzenleyen uygulama tarafından ayarlanan dengeleme seçenekleri
V 9.3.0 "XMSC_WMQ_BALANCING_TIMEOUT" sayfa 2031	Yeniden dengeleme, uygulama etkinliğini kesintiye uğratabilir.
XMSC_CLIENT_ID	Bir bağlantıya ilişkin istemci tanıtıcısı.
XMSC_CONNECTION_TYPE	Bir uygulamanın bağlandığı ileti sistemi sunucusunun tipi.
XMSC_PASSWORD	Bir ileti sistemi sunucusuna bağlanmaya çalışıldığında uygulamanın kimliğini doğrulamak için kullanılabilir parola.
"XMSC_RTT_BROKER_PING_INTERVAL" sayfa 2028	XMS.NET ' in milisaniye cinsinden zaman aralığı, herhangi bir etkinliği saptamak için Gerçek Zamanlı ileti sistemi sunucusuna olan bağlantıyı denetler.
XMSC_RTT_CONNECTION_PROTOCOL	Bir aracıya gerçek zamanlı bağlantı için kullanılan iletişim protokolü.
XMSC_RTT_HOST_NAME	Aracının çalıştığı sistemin anasistem adı ya da IP adresi.
XMSC_RTT_LOCAL_ADDRESS	Bir aracıya gerçek zamanlı bağlantı için kullanılacak yerel ağ arabiriminin anasistem adı ya da IP adresi.
XMSC_RTT_MULTICAST	Bir bağlantı üreticisi ya da hedefi için çoklu yayın ayarı.
XMSC_RTT_PORT	Bir aracının gelen istekleri dinlediğı kapının numarası.
XMSC_USERID	Bir ileti sistemi sunucusuna bağlanmayı denediğinde uygulamanın kimliğini doğrulamak için kullanılabilir kullanıcı kimliği.
XMSC_WMQ_BROKER_CONTROLQ	Bir aracı tarafından kullanılan denetim kuyruğunun adı.
XMSC_WMQ_BROKER_PUBQ	Uygulamaların yayınladıkları iletileri gönderdikleri bir aracı tarafından izlenen kuyruğun adı.
XMSC_WMQ_BROKER_QMGR	Aracının bağlı olduğı kuyruk yöneticisinin adı.
XMSC_WMQ_BROKER_SUBQ	Sürekli olmayan bir ileti tüketicisine ilişkin abone kuyruğunun adı.
XMSC_WMQ_BROKER_VERSION	Uygulama tarafından bir bağlantı ya da hedef için kullanılan aracının tipi.
"XMSC_WMQ_CCDTURL" sayfa 2033	İstemci kanal tanımlama çizelgesini içeren dosyanın adını ve yerini tanıtan ve dosyaya nasıl erişilebileceğini belirten URL.
XMSC_WMQ_KANAL	Bağlantı için kullanılacak kanalın adı.
"XMSC_WMQ_CLIENT_RECONNECT_OPTIONS" sayfa 2034	Bu özellik, bu üretici tarafından yaratılan yeni bağlantılar için istemci yeniden bağlanma seçeneklerini belirtir.
"XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT" sayfa 2035	Bu özellik, bir istemci bağlantısının yeniden başlanmaya çalışılması için saniye cinsinden belirler.
XMSC_WMQ_CONNECTION_MODE	Bir uygulamanın kuyruk yöneticisine bağlanma kipi.

<i>Çizelge 873. ConnectionFactory Özellikleri (devamı var)</i>	
Özelliğın adı	Açıklama
"XMSC_WMQ_CONNECTION_NAME_LIST" sayfa 2035	Bu özellik, istemcinin bağlantısı kesildikten sonra yeniden bağlanmaya çalıştığı anasistemleri belirtir.
XMSC_WMQ_FAIL_IF_QUIESCE	Uygulamanın bağlı olduğu kuyruk yöneticisi susturma durumundaysa, belirli yöntemlere yapılan çağrıların başarısız olup olmadığını belirler.
XMSC_WMQ_HOST_NAME	Kuyruk yöneticisinin çalıştığı sistemin anasistem adı ya da IP adresi.
XMSC_WMQ_LOCAL_ADDRESS	Kuyruk yöneticisine yönelik bir bağlantı için bu özellik, kullanılacak yerel ağ arabirimini ya da kullanılacak yerel kapı ya da yerel kapı aralığını ya da her ikisini de belirtir.
XMSC_WMQ_MESSAGE_SELECTION	İleti seçiminin XMS istemcisi tarafından mı, yoksa aracı tarafından mı yapılacağını belirler.
XMSC_WMQ_MSG_BATCH_SIZE	Zamanuyumsuz ileti teslimi kullanılırken bir toplu işte kuyruktan alınacak ileti sayısı üst sınırı.
XMSC_WMQ_POLLING_INTERVAL	Bir oturum içindeki her ileti dinleyicinin kuyruğunda uygun bir ileti yoksa, bu değer, her ileti dinleyicisinin kuyruğundan ileti almaya yeniden çalışmadan önce geçmesi gereken milisaniye cinsinden aralık üst sınırıdır.
"XMSC_WMQ_PROVIDER_VERSION" sayfa 2043	Uygulamanın bağlanmayı planladığı kuyruk yöneticisinin sürümü, yayını, değişiklik düzeyi ve düzeltme paketi.
XMSC_WMQ_PORT	Kuyruk yöneticisinin gelen istekleri dinlediği kapının numarası.
XMSC_WMQ_PUB_ACK_INTERVAL	XMS istemcisi aracından bir alındı bildirimini istemeden önce bir yayınlıyıcı tarafından yayınlanan iletilerin sayısı.
"XMSC_WMQ_PUT_ASYNC_ALLOWED" sayfa 2039	Bu özellik, ileti üreticilerinin bu hedefe ileti göndermek için zamanuyumsuz girişler kullanmalarına izin verilip verilmeyeceğini belirler.
XMSC_WMQ_QMGR_CCSD	İleti Kuyruğu Arabirimi 'nde (MQI) tanımlanan karakter verileri alanlarının XMS istemcisi ile IBM MQ istemcisi arasında değiş tokuş edildiği, kodlanmış karakter takımının ya da kod sayfasının tanıtıcısı (CCSID).
XMSC_WMQ_QUEUE_MANAGER	Bağlanılacak kuyruk yöneticisinin adı.
XMSC_WMQ_RECEIVE_EXIT	çalıştırılacak kanal alma çıkışı tanımlar.
XMSC_WMQ_RECEIVE_EXIT_INIT	Bir kanala geçirilen kullanıcı verileri, çağrıldığında çıkış alır.
XMSC_WMQ_SECURITY_EXIT	Bir kanal güvenlik çıkışı tanımlar.
XMSC_WMQ_SECURITY_EXIT_INIT	Bir kanal güvenlik çıkışına çağrıldığında geçirilen kullanıcı verileri.
"XMSC_WMQ_SEND_CHECK_COUNT" sayfa 2048	Tek bir hareket etmeyen XMS oturumunda, zamanuyumsuz koyma hatalarının denetlenmesi arasında izin verilecek gönderme çağrılarının sayısı.
XMSC_WMQ_SEND_EXIT	Bir kanal gönderme çıkışı tanımlar.
XMSC_WMQ_SEND_EXIT_INIT	Çağrıldığında kanal gönderme çıkışlarına geçirilen kullanıcı verileri.

Çizelge 873. ConnectionFactory Özellikleri (devamı var)	
Özelliğin adı	Açıklama
“XMSC_WMQ_SHARE_CONV_ALLOWED” sayfa 2048	Bir istemci bağlantısının, kanal tanımlamaları eşleşiyorsa, aynı işlemde aynı kuyruk yöneticisine yapılan diğer üst düzey XMS bağlantılarla yuvasını paylaşım paylaşamayabileceğini belirler. Bu özellik, uygulama geliştirme, bakım ya da işletim nedenleriyle gerektiğinde Connections 'ın ayrı yuvalarda tam olarak yalıtılmasına olanak sağlamak için sağlanır.
XMSC_WMQ_SSL_CERT_STORES	Kuyruk yöneticisine SSL bağlantısında kullanılacak sertifika iptal listelerini (CRL) bulunduran sunucuların yerleri.
XMSC_WMQ_SSL_CIPHER_SPEC	Bir kuyruk yöneticisine güvenli bir bağlantıda kullanılacak CipherSpec ' in adı.
XMSC_WMQ_SSL_CIPHER_SUITE	Bir kuyruk yöneticisine TLS bağlantısında kullanılacak CipherSuite ' in adı. Güvenli bağlantının kararlaştırılmasında kullanılan protokol, belirtilen CipherSuite' e bağlıdır.
XMSC_WMQ_SSL_CRYPTO_HW	İstemci sistemine bağlı şifreleme donanımına ilişkin yapılandırma ayrıntıları.
XMSC_WMQ_SSL_FIPS_REQUIRED	Bu özelliğin değeri, bir uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanıp kullanamayacağını belirler. Bu özellik true olarak ayarlanırsa, istemci-sunucu bağlantısı için yalnızca FIPS algoritmaları kullanılır.
XMSC_WMQ_SSL_KEY_REPOSITORY	Anahtarların ve sertifikaların saklandığı anahtar veritabanı dosyasının konumu.
XMSC_WMQ_SSL_KEY_RESETCOUNT	KeyResetSayısı, gizli anahtar yeniden anlaşılmadan önce SSL etkileşimi içinde gönderilen ve alınan toplam şifrelenmemiş bayt sayısını temsil eder.
XMSC_WMQ_SSL_PEER_NAME	Kuyruk yöneticisine SSL bağlantısında kullanılacak eşdüzey ad.
XMSC_WMQ_SYNCPOINT_ALL_TR	Eşitleme noktası denetimi içindeki kuyruklardan tüm iletilerin alınıp alınmayacağı.
“XMSC_WMQ_TARGET_CLIENT” sayfa 2054	
XMSC_WMQ_TEMP_Q_PREFIX	Uygulama bir XMS geçici kuyruğu oluşturduğunda oluşturulan IBM MQ dinamik kuyruğunun adını oluşturmak için kullanılan önek.
XMSC_WMQ_TEMP_TOPIC_PREFIX	Geçici konular oluştururken XMS , "TEMP/TEMPTOPICPREFIX/unique_id" biçiminde bir konu dizgisi oluşturur ya da bu özellik varsayılan değeri içeriyorsa, bu dizgi ("TEMP/unique_id") oluşturulur. Boş olmayan bir değer belirtilmesi, abonelerin bu bağlantı altında yaratılan geçici konulara ilişkin yönetilen kuyruklarının yaratılması için belirli model kuyruklarının tanımlanmasına olanak sağlar.
XMSC_WMQ_TEMPORARY_MODEL	Uygulama bir XMS geçici kuyruğu yarattığında dinamik kuyruğun yaratıldığı IBM MQ model kuyruğunun adı.

<i>Çizelge 873. ConnectionFactory Özellikleri (devamı var)</i>	
Özelliğın adı	Açıklama
<u>XMSC_WPM_BUS_NAME</u>	Bir bağlantı üreticisi için, uygulamanın bağlandığı hizmet tümleştirme veriyolunun adı ya da hedef için, hedefin bulunduğı hizmet tümleştirme veriyolunun adı.
<u>XMSC_WPM_CONNECTION_CLOSE</u>	Bağlantı için bağlantı yakınlığı ayarı.
<u>XMSC_WPM_DUR_SUB_HOME</u>	Bir bağlantıya ya da hedefe ilişkin tüm sürekli aboneliklerin yönetildiğı ileti alışverişi altyapısının adı.
<u>XMSC_WPM_LOCAL_ADDRESS</u>	Bir hizmet bütünleştirme veriyolu bağlantısı için bu özellik, kullanılacak yerel ağ arabirimini ya da kullanılacak yerel kapı ya da yerel kapı aralığını ya da her ikisini de belirtir.
<u>XMSC_WPM_NON_PERSISTENT_MAP</u>	Bağlantı kullanılarak gönderilen kalıcı olmayan iletilerin güvenilirlik düzeyi.
<u>XMSC_WPM_PERSISTENT_MAP</u>	Bağlantı kullanılarak gönderilen kalıcı iletilerin güvenilirlik düzeyi.
<u>XMSC_WPM_PROVIDER_ENDPOINTS</u>	Önyükleme sunucularının bir ya da daha çok uç noktası adresi sırası.
<u>XMSC_WPM_TARGET_GROUP</u>	İleti alışverişi altyapılarının hedef grubunun adı.
<u>XMSC_WPM_TARGET_ANLAM</u>	İleti alışverişi altyapılarının hedef grubunun önemi.
<u>XMSC_WPM_TARGET_TRANSPORT_CHAIN</u>	Uygulamanın bir ileti alışverişi altyapısına bağlanmak için kullanması gereken gelen iletim zincirinin adı.
<u>XMSC_WPM_TARGET_TYPE</u>	İleti alışverişi altyapılarının hedef grubunun tipi.
<u>XMSC_WPM_TEMP_Q_PREFIX</u>	Uygulama bir XMS geçici kuyruğı yarattığında hizmet tümleştirme veriyolunda yaratılan geçici kuyruğun adını oluşturmak için kullanılan önek.
<u>XMSC_WPM_TEMP_TOPIC_PREFIX</u>	Uygulama tarafından oluşturulan geçici bir konunun adını oluşturmak için kullanılan önek.

ConnectionMetaVerilerinin Özellikleri

Daha ayrıntılı başvuru bilgileri bağlantılarıyla birlikte ConnectionMetaVeri nesnesinin özelliklerine genel bakış.

<i>Çizelge 874. ConnectionMetaVerilerinin Özellikleri</i>	
Özelliğın adı	Açıklama
<u>XMSC_JMS_ANKESÖR_SÜRÜMÜ</u>	XMS belirtiminin temel aldığı JMS belirtiminin ana sürüm numarası. Bu özellik salt okunur.
<u>XMSC_JMS_MINOR_VERSION</u>	XMS belirtiminin temel aldığı JMS belirtiminin ikincil sürüm numarası. Bu özellik salt okunur.
<u>XMSC_JMS_VERSION</u>	XMS belirtiminin temel aldığı JMS belirtiminin sürüm tanıtıcısı. Bu özellik salt okunur.
<u>XMSC_ANCE_SÜRÜMÜ</u>	XMS istemcisinin sürüm numarası. Bu özellik salt okunur.
<u>XMSC_MINOR_VERSION</u>	XMS istemcisinin yayın numarası. Bu özellik salt okunur.
<u>XMSC_PROVIDER_NAME</u>	XMS istemcisinin sağlayıcısı. Bu özellik salt okunur.

Çizelge 874. ConnectionMetaVerilerinin Özellikleri (devamı var)

Özelliğın adı	Açıklama
<u>XMSC_VERSION</u>	cliXMS' nin sürüm tanıtıcısı. Bu özellik salt okunur.

Hedef Özellikleri

Daha ayrıntılı başvuru bilgileri bağlantılarıyla birlikte, Hedef nesnenin özelliklerine ilişkin genel bakış.

Çizelge 875. Hedef Özellikleri

Özelliğın adı	Açıklama
<u>XMSC_DELIVERY_MODE</u>	Hedefe gönderilen iletilerin teslim kipi.
<u>XMSC_PRIORITY</u>	Hedefe gönderilen iletilerin önceliği.
<u>XMSC_RTT_MULTICAST</u>	Bir bağlantı üreticisi ya da hedefi için çoklu yayın ayarı.
<u>XMSC_TIME_TO_LIVE</u>	Hedefe gönderilen iletiler için yaşam süresi.
<u>XMSC_WMQ_BROKER_VERSION</u>	Uygulama tarafından bir bağlantı ya da hedef için kullanılan aracının tipi.
<u>XMSC_WMQ_CCSD</u>	XMS istemcisi iletiyi hedefe ilettiğinde, iletinin gövdesindeki karakter verileri dizgilerinin içinde olduğu kodlanmış karakter takımının ya da kod sayfasının tanıtıcısı (CCSID).
<u>XMSC_WMQ_DUR_SUBQ</u>	Hedefte ileti alan sürekli bir aboneye ilişkin abone kuyruğunun adı. Not: Bu özellik, IBM Message Service Client for .NET olanağının 2.0 sürümüyle kullanılabilir; ancak, bağlantı üreticisinin XMSC_WMQ_PROVIDER_VERSION özelliği 7 'den küçük bir sürüm numarasına ayarlanmadıkça, IBM WebSphere MQ 7.0 kuyruk yöneticisine bağlı bir uygulama için herhangi bir etkisi yoktur.
<u>XMSC_WMQ_ENCODING</u>	XMS istemcisi iletiyi hedefe ilettiğinde, iletinin gövdesindeki sayısal verilerin nasıl gösterileceği.
<u>XMSC_WMQ_FAIL_IF_QUIESCE</u>	Uygulamanın bağlı olduğu kuyruk yöneticisi susturma durumundaysa, belirli yöntemlere yapılan çağrıların başarısız olup olmadığını belirler.
<u>“XMSC_WMQ_MESSAGE_BODY” sayfa 2037</u>	Bu özellik, bir XMS uygulamasının bir IBM MQ iletinin MQRFH2 ' sini ileti bilgi yükünün (ileti gövdesinin bir parçası olarak) bir parçası olarak işleyip işlemediğini belirler.
<u>“XMSC_WMQ_MQMD_MESSAGE_CONTEXT” sayfa 2038</u>	XMS uygulaması tarafından ayarlanacak ileti bağlamı düzeyini belirler. Bu özelliğın yürürlüğe girmesi için uygulamanın uygun bağlam yetkisiyle çalışması gerekir.
<u>“XMSC_WMQ_MQMD_READ_ENABLED” sayfa 2039</u>	Bu özellik, bir XMS uygulamasının MQMD alanlarının değerlerini alıp almayacağını belirler.
<u>“XMSC_WMQ_MQMD_WRITE_ENABLED” sayfa 2039</u>	Bu özellik, bir XMS uygulamasının MQMD alanlarının değerlerini ayarlayıp ayarlayamayacağını belirler.

<i>Çizelge 875. Hedef Özellikleri (devamı var)</i>	
Özellğin adı	Açıklama
“XMSC_WMQ_READ_AHEAD_ALLOWED” sayfa 2040	Bu özellik, ileti tüketicilerinin ve kuyruk tarayıcılarının, bu hedeften kalıcı olmayan, işlemsel olmayan iletileri almadan önce iç arabelleğe almak için önden okuma özelliğini kullanıp kullanamayacağını belirler.
“XMSC_WMQ_READ_AHEAD_CLOSE_POLICY” sayfa 2040	Bu özellik, zamanuyumsuz bir ileti dinleyicisine teslim edilen iletiler için, ileti tüketicisi kapatıldığında iç önden okuma arabelleğindeki iletilere ne olacağını belirler.
“XMSC_WMQ_RECEIVE_CCSD” sayfa 2045	Kuyruk yöneticisi ileti dönüştürmesi için hedef CCSID 'yi belirleyen hedef özellik. XMSC_WMQ_RECEIVE_CONVERSION_QMGR değeri WMQ_RECEIVE_CONVERSION_QMGR olarak ayarlanmadıkça değer yoksayılr.
“XMSC_WMQ_RECEIVE_CONVERSION” sayfa 2045	Veri dönüştürme işleminin kuyruk yöneticisi tarafından gerçekleştirilip gerçekleştirilmeyeceğini belirleyen hedef özellik.
XMSC_WMQ_TARGET_CLIENT	Hedefe gönderilen iletilerin bir MQRFH2 üstbilgisi içerip içermediğini belirler.
XMSC_WMQ_TEMP_TOPIC_PREFIX	Geçici konular oluştururken XMS, "TEMP/TEMPTOPICPREFIX/unique_id" biçiminde bir konu dizgisi oluşturur ya da bu özellik varsayılan değeri içeriyorsa, bu dizgi ("TEMP/unique_id") oluşturulur. Boş olmayan bir değer belirtilmesi, abonelerin bu bağlantı altında yaratılan geçici konulara ilişkin yönetilen kuyruklarının yaratılması için belirli model kuyruklarının tanımlanmasına olanak sağlar.
XMSC_WPM_BUS_NAME	Bir bağlantı üreticisi için, uygulamanın bağlandığı hizmet tümleştirme veriyolunun adı ya da hedef için, hedefin bulunduğu hizmet tümleştirme veriyolunun adı.
XMSC_WPM_TOPIC_SPACE	Konuyu içeren konu alanının adı.

InitialContext özellikleri

Daha ayrıntılı başvuru bilgileri bağlantılarıyla birlikte InitialContext nesnesinin özelliklerine genel bakış.

<i>Çizelge 876. InitialContext özellikleri</i>	
Özellğin adı	Açıklama
XMSC_IC_PROVIDER_URL	COS adlandırma hizmetinin web hizmetiyle aynı sunucuda olması gerekmeyecek şekilde JNDI adlandırma dizinini bulmak için kullanılır.
XMSC_IC_SECURITY_AUTHENTICATION	Java Bağlam arabirimi SECURITY_AUTHENTICATION ögesine dayalı olarak. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.
XMSC_IC_SECURITY_CREDENTIALS	Java Bağlam arabirimi SECURITY_CREDENTIALS. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.
XMSC_IC_SECURITY_PRINCIPAL	Java Bağlam arabirimi SECURITY_PRINCIPAL değerine dayalı olarak. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

<i>Çizelge 876. InitialContext özellikleri (devamı var)</i>	
Özellğin adı	Açıklama
<u>XMSC_IC_SECURITY_PROTOCOL</u>	Java Bağlam arabirimi SECURITY_PROTOCOL değerine dayalı Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.
<u>XMSC_IC_URL</u>	LDAP ve FileSystem bağlamları için, yönetilen nesnelere içeren havuzun adresi. COS adlandırma bağlamları için, dizindeki nesnelere tarayan web hizmetinin adresi.

İleti Özellikleri

Daha ayrıntılı başvuru bilgileri bağlantılarıyla birlikte, İleti nesnesinin özelliklerine ilişkin genel bakış.

<i>Çizelge 877. İleti Özellikleri</i>	
Özellğin adı	Açıklama
<u>JMS_IBM_CHARACTER_SET</u>	XMS istemcisi iletiyi istenen hedefe ilettiğinde, iletinin gövdesindeki karakter verileri dizgilerinin içinde olduğu kodlanmış karakter takımının ya da kod sayfasının tanıtıcısı (CCSID). XMS içinde bu özellik sayısal bir değere sahiptir ve CCSID ile eşlenir. Ancak, bu özellik bir JMS özelliğine dayalıdır; dizgi tipi değeri vardır ve bu sayısal CCSID 'yi gösteren Java karakter takımıyla eşlenir.
<u>JMS_IBM_ENCODING</u>	XMS istemcisi iletiyi hedeflenen hedefine ilettiğinde, iletinin gövdesindeki sayısal verilerin nasıl temsil edileceği.
<u>JMS_IBM_EXCEPTIONMESSAGE</u>	İletin kural dışı durum hedefine neden gönderildiğini açıklayan metin. Bu özellik salt okunur.
<u>JMS_IBM_EXCEPTIONPROBLEMDESTINATION</u>	İletin kural dışı durum hedefine gönderilmeden önce olduğu hedefin adı.
<u>JMS_IBM_EXCEPTIONREASON</u>	İletin kural dışı durum hedefine gönderilme nedenini gösteren neden kodu.
<u>JMS_IBM_EXCEPTIONTIMESTAMP</u>	İletin kural dışı durum hedefine gönderildiği zaman.
<u>JMS_IBM_FEEDBACK</u>	Rapor iletisinin niteliğini gösteren kod.
<u>JMS_IBM_FORMAT</u>	İletideki uygulama verilerinin türü.
<u>JMS_IBM_LAST_MSG_IN_GROUP</u>	İletin bir ileti grubundaki son ileti olup olmadığını belirtin.
<u>JMS_IBM_MSGTYPE</u>	İletin tipi.
<u>JMS_IBM_PUTAPPLTYPE</u>	İletiyi gönderen uygulamanın tipi.
<u>JMS_IBM_PUTDATE</u>	İletin gönderildiği tarih.
<u>JMS_IBM_PUTTIME</u>	İletin gönderildiği saat.
<u>JMS_IBM_REPORT_COA</u>	Özgün iletideki uygulama verilerinin bir rapor iletisine ne kadar eklenmesi gerektiğini belirten 'varış sırasında onayla' rapor iletilerini isteyin.
<u>JMS_IBM_REPORT_COD</u>	Özgün iletideki uygulama verilerinin bir rapor iletisine ne kadar eklenmesi gerektiğini belirten 'teslimde onayla' rapor iletilerini isteyin.

Çizelge 877. İleti Özellikleri (devamı var)	
Özelliğın adı	Açıklama
<u>JMS_IBM_REPORT_DISCARD_MSG</u>	İletinin hedeflenen hedefine teslim edilememesi durumunda atılmasını isteyin.
<u>JMS_IBM_REPORT_EXCEPTION</u>	Özgün iletideki uygulama verilerinin bir rapor iletisine ne kadar eklenmesi gerektiğini belirten kural dışı durum raporu iletileri isteyin.
<u>JMS_IBM_REPORT_SÜRE sonu</u>	Özgün iletideki uygulama verilerinin bir rapor iletisine ne kadar dahil edilmesi gerektiğini belirten, süre bitim raporu iletileri isteyin.
<u>JMS_IBM_REPORT_NAN</u>	Olumsuz eylem bildirim raporu iletileri isteyin.
<u>JMS_IBM_REPORT_PAN</u>	Pozitif eylem bildirim raporu iletileri isteyin.
<u>JMS_IBM_REPORT_PASS_CORREL_ID</u>	Herhangi bir raporun ya da yanıt iletisinin ilinti tanıtıcısının, özgün iletinin ilinti tanıtıcısıyla aynı olması isteğı.
<u>JMS_IBM_REPORT_PASS_MSG_ID</u>	Herhangi bir raporun ya da yanıt iletisinin ileti tanıtıcısının, özgün iletinin ileti tanıtıcısıyla aynı olduğunu isteyin.
<u>JMS_IBM_RETAIN</u>	Bu özelliğın ayarlanması, kuyruk yöneticisine bir iletinin Ulaşılan Yayın olarak kabul edileceğini gösterir.
<u>JMS_IBM_SYSTEM_MESSAGEID</u>	İletiyi hizmet bütünleştirme veriyolu içinde benzersiz olarak tanımlayan bir tanıtıcı. Bu özellik salt okunur.
<u>JMSX_APPID</u>	İletiyi gönderen uygulamanın adı.
<u>JMSX_DELIVERY_COUNT</u>	İletiyi teslim etme girişimi sayısı.
<u>JMSX_GROUPID</u>	İletinin ait olduğu ileti grubunun tanıtıcısı.
<u>JMSX_GROUPSEQ</u>	İleti grubundaki iletinin sıra numarası.
<u>JMSX_USERID</u>	İletiyi gönderen uygulamayla ilişkili kullanıcı kimliğı.

JMS_IBM_MQMD* özellikleri

IBM Message Service Client for .NET , istemci uygulamalarının API ' leri kullanarak MQMD alanlarını okumasını/yazmasını sağlar. Ayrıca, MQ ileti verilerine erişim sağlar. Varsayılan olarak MQMD erişimi devre dışıdır ve Hedef özellikleri XMSC_WMQ_MQMD_WRITE_ENABLED ve XMSC_WMQ_MQMD_READ_ENABLED kullanılarak uygulama tarafından belirtik olarak etkinleştirilmelidir. Bu iki özellik birbirinden bağımsızdır.

StrucId ve Version dışındaki tüm MQMD alanları ek Message nesnesi özellikleri olarak gösterilir ve öneki JMS_IBM_MQMD ' dir.

JMS_IBM_MQMD* özellikleri, önceki çizelgede açıklanan JMS_IBM* gibi diğer özelliklerden daha önceliklidir.

İleti gönderilmesi

StrucId ve Version dışındaki tüm MQMD alanları gösterilir. Bu özellikler yalnızca MQMD alanlarını belirtir; bir özellik hem MQMD 'de hem de MQRFH2 üstbilgisinde gerçekleştiğinde, MQRFH2 'deki sürüm ayarlanmaz ya da alınmaz. JMS_IBM_MQMD_BackoutCount dışında, bu özelliklerden herhangi biri ayarlanabilir. JMS_IBM_MQMD_BackoutCount için ayarlanan herhangi bir değer yoksayılr.

Bir özelliğın uzunluk üst sınırı varsa ve çok uzun bir değer sağlarsanız, değer kesilir.

Belirli özellikler için, Hedef nesnede XMSC_WMQ_MQMD_MESSAGE_CONTEXT özelliğini de ayarlamanız gerekir. Bu özelliğin yürürlüğe girmesi için uygulamanın uygun bağlam yetkisiyle çalışması gerekir. XMSC_WMQMD_MESSAGE_CONTEXT değerini uygun bir değere ayarlamazsanız, özellik değeri yoksayılır. XMSC_WMQMD_MESSAGE_CONTEXT değerini uygun bir değere ayarlarsanız, ancak kuyruk yöneticisi için yeterli bağlam yetkiniz yoksa, bir kural dışı durum yayınlanır. XMSC_WMQMD_MESSAGE_CONTEXT için belirli değerler gerektiren özellikler şunlardır.

Aşağıdaki özellikler XMSC_WMQ_MQMD_MESSAGE_CONTEXT 'in XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT ya da XMSC_WMQ_MDCTX_SET_ALL_CONTEXT olarak ayarlanmasını gerektirir:

- JMS_IBM_MQMD_UserIdentifier
- JMS_IBM_MQMD_AccountingToken
- JMS_IBM_MQMD_ApplIdentityVerileri

Aşağıdaki özellikler, XMSC_WMQMD_MESSAGE_CONTEXT 'in XMSC_WMQ_MDCTX_SET_ALL_CONTEXT olarak ayarlanmasını gerektirir:

- JMS_IBM_MQMD_PutApplTipi
- JMS_IBM_MQMD_PutApplAdı
- JMS_IBM_MQMD_PutDate
- JMS_IBM_MQMD_PutTime
- JMS_IBM_MQMD_ApplOriginVerileri

İleti alınması

XMSC_WMQ_MQMD_READ_ENABLED özelliği, üreten uygulamanın ayarladığı gerçek özelliklerden bağımsız olarak true değerine ayarlanırsa, alınan bir iletide tüm bu özellikler kullanılabilir. Bir uygulama, JMS belirtimine göre önce tüm özellikler temizlenmedikçe, alınan bir iletinin özelliklerini değiştiremez. Alınan ileti, özellikler değiştirilmeden iletilebilir.

Not: Uygulamanız, XMSC_WMQ_MQMD_READ_ENABLED özelliği true olarak ayarlanmış bir hedeften bir ileti alırsa ve bunu, XMSC_WMQ_MQMD_WRITE_ENABLED true olarak ayarlanmış bir hedefe iletiyorsa, bu, iletilen iletiye kopyalanan iletinin tüm MQMD alan değerleriyle sonuçlanır. Özellikler tablosu

<i>Çizelge 878. MQMD alanlarını gösteren İleti nesnesinin özellikleri</i>		
Özellik	Açıklama	Tip
JMS_IBM_MQMD_REPORT	Rapor iletileri için seçenekler	System.Int32
JMS_IBM_MQMD_MSGTYPE	İleti tipi	System.Int32
JMS_IBM_MQMD_EXPIRY	ileti geçerlik süresi	System.Int32
JMS_IBM_MQMD_FEEDBACK	Geribildirim ya da neden kodu	System.Int32
JMS_IBM_MQMD_ENCODING	İleti verilerinin sayısal kodlaması	System.Int32
JMS_IBM_MQMD_CODEDCHARSETID	İleti verilerinin karakter kümesi tanıtıcısı	System.Int32
JMS_IBM_MQMD_FORMAT	İleti verilerinin biçim adı	System.String
JMS_IBM_MQMD_PRIORITY	İleti önceliği	System.Int32
	Not: JMS_IBM_MQMD_PRIORITY değerine 0-9 aralığında olmayan bir değer atarsanız, bu değer JMS belirtimini ihlal eder.	
JMS_IBM_MQMD_PERSISTENCE	İleti kalıcılığı	System.Int32

Çizelge 878. MQMD alanlarını gösteren İleti nesnesinin özellikleri (devamı var)

Özellik	Açıklama	Tip
JMS_IBM_MQMD_MSGID Not: JMS belirtimi, ileti tanıtıcısının JMS sağlayıcısı tarafından ayarlanması gerektiğini ve benzersiz ya da boş olması gerektiğini belirtiyor. JMS_IBM_MQMD_MSGID 'ye bir değer atarsanız, bu değer JMSMessageID' ye kopyalanır. Bu nedenle, JMS sağlayıcısı tarafından ayarlanmaz ve benzersiz olmayabilir: bu değer JMS belirtimini ihlal eder.	İleti Tanıtıcısı	Bayt Dizisi Not: Bir iletide byte dizisi özelliklerinin kullanılması JMS belirtimine aykırılık oluşturur.
JMS_IBM_MQMD_CORRELID Not: 'ID:' dizisiyle başlayan JMS_IBM_MQMD_CORRELID 'ye bir değer atarsanız, bu değer JMS belirtimini ihlal eder.	İlinti tanıtıcısı	Bayt Dizisi Not: Bir iletide byte dizisi özelliklerinin kullanılması JMS belirtimine aykırılık oluşturur.
JMS_IBM_MQMD_BACKOUTCOUNT	Geriletme sayacı	System.Int32
JMS_IBM_MQMD_REPLYTOQ	Yanıt kuyruğunun adı	System.String
JMS_IBM_MQMD_REPLYTOQMGR	Yanıt kuyruğu yöneticisinin adı	System.String
JMS_IBM_MQMD_USERIDENTIFIER	Kullanıcı kimliği	System.String
JMS_IBM_MQMD_ACCOUNTINGTOKEN	Muhasebe simgesi	Bayt Dizisi Not: Bir iletide byte dizisi özelliklerinin kullanılması JMS belirtimine aykırılık oluşturur.
JMS_IBM_MQMD_APPLIDENTITYDATA	Kimlikle ilgili uygulama verileri	System.String
JMS_IBM_MQMD_PUTAPPLTYPE	İletiyi koyan uygulamanın tipi	System.Int32
JMS_IBM_MQMD_PUTAPPLNAME	İletiyi koyan uygulamanın adı	System.String
JMS_IBM_MQMD_PUTDATE	İletinin konma tarihi	System.String
JMS_IBM_MQMD_PUTTIME	İletinin konma zamanı	System.String
JMS_IBM_MQMD_APPLORIGINDATA	Kaynak ile ilgili uygulama verileri	System.String
JMS_IBM_MQMD_GROUPID	Grup tanıtıcısı	Bayt Dizisi Not: Bir iletide byte dizisi özelliklerinin kullanılması JMS belirtimine aykırılık oluşturur.
JMS_IBM_MQMD_MSGSEQNUMBER	Grup içindeki yerel iletinin sıra numarası	System.Int32
JMS_IBM_MQMD_OFFSET	Mantıksal iletinin başlangıcından fiziksel iletideki verilerin göreceli konumu	System.Int32

Çizelge 878. MQMD alanlarını gösteren İleti nesnesinin özellikleri (devamı var)

Özellik	Açıklama	Tip
JMS_IBM_MQMD_MSGFLAGS	İleti İşaretleri	System.Int32
JMS_IBM_MQMD_ORIGINALLENBU	Özgün iletinin uzunluğu	System.Int32

Daha fazla ayrıntı için bkz. [MQMD](#) .

Örnekler

Bu örnek, bir iletinin MQMD.UserIdentifier , "JoeBloggs" olarak ayarlandı.

```
// 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
// ...
```

JMS_IBM_MQMD_USERIDENTIFIER ayarlanmadan önce XMSC_WMQMD_MESSAGE_CONTEXT ayarlanmalıdır. XMSC_WMQ_MQMD_MESSAGE_CONTEXT kullanımı hakkında daha fazla bilgi için İleti nesnesi özelliklerine bakın.

Benzer şekilde, bir ileti almadan önce XMSC_WMQ_MQMD_READ_ENABLED değerini true değerine ayarlayarak ve daha sonra, getStringözelligi gibi iletinin alma yöntemlerini kullanarak MQMD alanlarının içeriğini çıkarabilirsiniz. Alınan özellikler salt okunur özelliklerdir.

Bu örnek, MQMD.ApplIdentityData alanı.

```
// 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);
```

MessageConsumer Özellikleri

Daha ayrıntılı başvuru bilgileri bağlantılarıyla birlikte MessageConsumer nesnesinin özelliklerine genel bakış.

Çizelge 879. MessageConsumer Özellikleri	
Özelliğın adı	Açıklama
XMSC_IS_SUBSCRIPTION_MULTICAST	İletilerin WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim edilip edilmediğini gösterir. Bu özellik salt okunur.
XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST	İletilerin güvenilir bir hizmet kalitesiyle WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim edilip edilmediğini gösterir. Bu özellik salt okunur.

Bkz. [.Daha fazla ayrıntı için IMessageConsumer](#) ' in NET özellikleri.

MessageProducer Özellikleri

Daha ayrıntılı başvuru bilgileri bağlantılarıyla birlikte MessageProducer nesnesinin özelliklerine genel bakış.

Bkz. [.Daha fazla ayrıntı için IMessageProducer](#) ürününün NET özellikleri.

Oturum Özellikleri

Oturum nesnesinin özelliklerine ilişkin genel bakış ve daha ayrıntılı başvuru bilgilerine ilişkin bağlantılar.

Bkz. [.Daha fazla ayrıntı için ISession](#) ' un NET özellikleri.

Özellik Tanımları

Bu kısım, her nesne özelliğinin tanımlamasını sağlar.

Her özellik tanımlaması aşağıdaki bilgileri içerir:

- Özelliğın veri tipi
- Özelliğın olan nesne tipleri
- Bir Hedef özelliğın için, bir örnek kaynak tanıtıcısında (URI) kullanılabilir ad
- Özelliğın daha ayrıntılı bir açıklaması
- Özelliğın geçerli değerleri
- Özelliğın varsayılan değeri

Adları aşağıdaki örneklerden biriyle başlayan özellikler yalnızca belirtilen bağlantı tipiyle ilgilidir:

XMSC_RTT

Özellikler yalnızca gerçek zamanlı bir aracı bağlantısı için geçerlidir. Özelliklerin adları, `xmsc_rtt`. hüstbilgi dosyasında adlandırılmış sabitler olarak tanımlanır.

XMSC_WMQ

Özellikler yalnızca bir uygulama bir IBM MQ kuyruk yöneticisine bağlandığında geçerlidir. Özelliklerin adları, `xmsc_wmq`. hüstbilgi dosyasında adlandırılmış sabitler olarak tanımlanır.

XMSC_WPM

Özellikler yalnızca bir uygulama bir WebSphere hizmet bütünleştirme veriyoluna bağlandığında ilgilidir. Özelliklerin adları, `xmsc_wpm`. hüstbilgi dosyasında adlandırılmış sabitler olarak tanımlanır.

Tanımlarında aksi belirtilmedikçe, geri kalan özellikler her bağlantı tipi için geçerlidir. Özelliklerin adları, `xmsc`. hüstbilgi dosyasında adlandırılmış sabitler olarak tanımlanır. Adları JMSX önekiyle başlayan özellikler, bir iletinin JMS tanımlı özellikleri ve adları JMS-IBM önekiyle başlayan özellikler IBM iletinin tanımlı özellikleridir. İletilerin özellikleri hakkında daha fazla bilgi için bkz. [XMS iletisinin özellikleri](#).

Tanımlarında aksi belirtilmedikçe, her özellik hem noktadan noktaya hem de yayınlama abonelik etki alanlarıyla ilgilidir.

Bir uygulama, özellik salt okunur olarak atanmadıkça, herhangi bir özelliğin değerini alabilir ve ayarlayabilir.

JMS_IBM_CHARACTER_SET

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

XMS istemcisi iletiyi istenen hedefe ilettiğinde, iletinin gövdesindeki karakter verileri dizgilerinin içinde olduğu kodlanmış karakter takımının ya da kod sayfasının tanıtcısı (CCSID). XMS içinde bu özellik sayısal bir değere sahiptir ve CCSID ile eşlenir. Ancak, bu özellik bir JMS özelliğine dayalıdır; dizgi tipi değeri vardır ve bu sayısal CCSID ' yi gösteren Java karakter takımıyla eşlenir. Bu özellik, XMSC_WMQ_CCSID özelliği tarafından hedef için belirtilen CCSID ' yi geçersiz kılar.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

JMS_IBM_ENCODING

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

XMS istemcisi iletiyi hedeflenen hedefine ilettiğinde, iletinin gövdesindeki sayısal verilerin nasıl temsil edileceği. Bu özellik, XMSC_WMQ_ENCODING özelliği tarafından hedef için belirtilen herhangi bir kodlamayı geçersiz kılar. Özellik, ikili tamsayıların, paketlenmiş ondalık tamsayıların ve kayan noktalı sayıların gösterimini belirtir.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **Encoding** alanında belirtilebilir değerlerle aynıdır.

Bir uygulama, özelliği ayarlamak için aşağıdaki adlandırılmış değişmezleri kullanabilir:

Adlandırılmış değişmez

Anlamı

MQENC_INTEGER_NORMAL

Normal tamsayı kodlaması

MQENC_INTEGER_REVERSED

Tersine çevrilmiş tamsayı kodlaması

MQENC_DECIMAL_NORMAL

Normal paketlenmiş ondalık kodlama

MQENC_DECIMAL_REVERSED

Paketlenmiş onlu kodlama tersine çevrildi

MQENC_FLOAT_IEEE_NORMAL

Olağan IEEE kayan nokta kodlaması

MQENC_FLOAT_IEEE_REVERSED

Ters IEEE kayan nokta kodlaması

MQENC_FLOAT_S390

z/OS mimarisi kayan nokta kodlaması

MQENC_NATIVE

Yerli makine kodlaması

Özelliğe ilişkin bir değer oluşturmak için, uygulama bu sabitlerden üçünü aşağıdaki gibi ekleyebilir:

- İkili tamsayıların gösterimini belirtmek için adı MQENC_INTEGER ile başlayan bir değişmez
- Paketlenmiş ondalık tamsayıların gösterimini belirtmek için adı MQENC_DECIMAL ile başlayan bir değişmez
- Kayan noktalı sayıların gösterimini belirtmek için adı MQENC_FLOAT ile başlayan bir değişmez

Diğer bir seçenek olarak, uygulama özelliği değeri ortama bağımlı olan MQENC_NATIVE olarak ayarlayabilir.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

JMS_IBM_EXCEPTIONMESSAGE

Veri tipi:

Dizgi

Bu özelliğin özelliği:

İleti

İletinin kural dışı durum hedefine neden gönderildiğini açıklayan metin. Bu özellik salt okunur.

Bu özellik yalnızca bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ve bir kural dışı durum hedefinden bir ileti alındığında geçerlidir.

JMS_IBM_EXCEPTIONPROBLEMDESTINATION

Veri tipi:

Dizgi

Bu özelliğin özelliği:

İleti

İletinin kural dışı durum hedefine gönderilmeden önce olduğu hedefin adı.

Bu özellik yalnızca bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ve bir kural dışı durum hedefinden bir ileti alındığında geçerlidir.

JMS_IBM_EXCEPTIONREASON

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

İletinin kural dışı durum hedefine gönderilme nedenini gösteren neden kodu.

Bu özellik yalnızca bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ve bir kural dışı durum hedefinden bir ileti alındığında geçerlidir.

JMS_IBM_EXCEPTIONTIMESTAMP

Veri tipi:

System.Int64

Bu özelliğin özelliği:

İleti

İletinin kural dışı durum hedefine gönderildiği zaman.

Saat, 1 Ocak 1970 'te GMT 00:00:00 'dan bu yana milisaniye olarak ifade edilir.

Bu özellik yalnızca bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ve bir kural dışı durum hedefinden bir ileti alındığında geçerlidir.

JMS_IBM_FEEDBACK

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

Rapor iletisinin niteliğini gösteren kod.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **Feedback** alanında belirlenebilecek geribildirim kodları ve neden kodlarıdır.

Varsayılan olarak özellik ayarlanmaz.

JMS_IBM_FORMAT

Veri tipi:

Dizgi

Bu özelliğin özelliği:

İleti

İletideki uygulama verilerinin türü.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **Format** alanında belirtilebilir değerlerle aynıdır.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

JMS_IBM_LAST_MSG_IN_GROUP

Veri tipi:

System.Boolean

Bu özelliğin özelliği:

İleti

İletinin bir ileti grubundaki son ileti olup olmadığını belirtin.

İleti bir ileti grubundaki son iletiyse özelliği true olarak ayarlayın. Ters durumda, özelliği false olarak ayarlayın ya da özelliği ayarlamayın. Varsayılan olarak özellik ayarlanmaz.

True değeri, bir ileti açıklayıcısının **MsgFlags** alanında belirlenebilecek MQMF_LAST_MSG_IN_GROUP durum işaretine karşılık gelir.

Bu özellik yayınlama/abone olma etki alanında yoksayılr ve bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ilgili değildir.

JMS_IBM_MSGTYPE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

İletinin tipi.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer	Anlamı
MQMT_DATAGRAM	İleti, yanıt gerektirmeyen bir ileti.
MQMT_REQUEST	İleti, yanıt gerektiren bir iletidir.
MQMT_REPLY	İleti bir yanıt iletisidir.
MQMT_REPORT	İleti bir rapor iletisidir.

Bu değerler, bir ileti açıklayıcısının **MsgType** alanında belirtilebilecek ileti tiplerine karşılık gelir.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

JMS_IBM_PUTAPPLTYPE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

İletiyi gönderen uygulamanın tipi.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **PutApp1Type** alanında belirtilebilir uygulama tipleridir.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

JMS_IBM_PUTDATE**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

İleti

İletinin gönderildiği tarih.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **PutDate** alanında belirtilebilir değerlerle aynıdır.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

JMS_IBM_PUTTIME**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

İleti

İletinin gönderildiği saat.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **PutTime** alanında belirtilebilir değerlerle aynıdır.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

JMS_IBM_REPORT_COA**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

İleti

Özgün iletideki uygulama verilerinin bir rapor iletisine ne kadar eklenmesi gerektiğini belirten 'varış sırasında onayla' rapor iletilerini isteyin.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

MORO_COA

MORO_COA_WITH_DATA

MORO_COA_WITH_FULL_DATA

Anlamı

Bir rapor iletisine dahil edilen özgün iletiden herhangi bir uygulama verisi olmadan 'gelişte onayla' rapor iletilerini isteyin.

Bir rapor iletisinde yer alan özgün iletiden ilk 100 baytlık uygulama verileriyle birlikte 'gelişi onayla' rapor iletilerini isteyin.

Bir rapor iletisine dahil edilen özgün iletideki tüm uygulama verileriyle birlikte 'varışta onayla' rapor iletilerini isteyin.

Bu deęerler, bir ileti tanımlayıcısının **Report** alanında belirlenebilecek rapor seçeneklerine karşılık gelir. Bu seçeneklerle ilgili daha fazla bilgi için bkz. [Rapor \(MQLONG\)](#).

Varsayılan olarak özellik ayarlanmaz.

JMS_IBM_REPORT_COD

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

Özgün iletideki uygulama verilerinin bir rapor iletisine ne kadar eklenmesi gerektiğini belirten 'teslimde onayla' rapor iletilerini isteyin.

Özelliğin geçerli deęerleri şunlardır:

Geçerli deęer

Anlamı

MQRO_COD

Bir rapor iletisine dahil edilen özgün iletiden herhangi bir uygulama verisi olmadan 'teslimde onayla' rapor iletilerini isteyin.

MQRO_COD_WITH_DATA

Bir rapor iletisinde yer alan özgün iletiden ilk 100 baytlık uygulama verileriyle birlikte 'teslimde onayla' rapor iletilerini isteyin.

MQRO_COD_WITH_FULL_DATA

Bir rapor iletisine dahil edilen özgün iletideki tüm uygulama verileriyle birlikte 'teslimde onayla' rapor iletilerini isteyin.

Bu deęerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilen rapor seçeneklerine karşılık gelir.

Varsayılan olarak özellik ayarlanmaz.

JMS_IBM_REPORT_DISCARD_MSG

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

İletinin hedeflenen hedefine teslim edilememesi durumunda atılmasını isteyin.

İletinin hedeflenen hedefine teslim edilememesi durumunda atılması isteğinde bulunmak için özelliği MQRO_DISCARD_MSG olarak ayarlayın. İletinin gitmeyen ileteler kuyruğuna konması ya da bir kural dışı durum hedefine gönderilmesi gerekiyorsa, özelliği ayarlamayın. Varsayılan olarak özellik ayarlanmaz.

MQRO_DISCARD_MSG deęeri, bir ileti açıklayıcısının **Report** alanında belirtilebilir bir rapor seçeneğine karşılık gelir.

JMS_IBM_REPORT_EXCEPTION

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

Özgün iletideki uygulama verilerinin bir rapor iletisine ne kadar eklenmesi gerektiğini belirten kural dışı durum raporu iletileri isteyin.

Özelliğin geçerli deęerleri şunlardır:

Geçerli değer

MORO_EXCEPTION

MORO_EXCEPTION_WITH_DATA

MORO_EXCEPTION_WITH_FULL_DATA

Anlamı

Bir rapor iletisine dahil edilen özgün iletiden uygulama verisi olmadan kural dışı durum raporu iletileri isteyin.

Bir rapor iletisinde yer alan özgün iletiden alınan ilk 100 baytlık uygulama verileriyle birlikte kural dışı durum raporu iletileri isteyin.

Bir rapor iletisine dahil edilen özgün iletideki tüm uygulama verileriyle birlikte kural dışı durum raporu iletileri isteyin.

Bu değerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilen rapor seçeneklerine karşılık gelir.

Varsayılan olarak özellik ayarlanmaz.

JMS_IBM_REPORT_SÜRE SONU**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

İleti

Özgün iletideki uygulama verilerinin bir rapor iletisine ne kadar dahil edilmesi gerektiğini belirten, süre bitim raporu iletileri isteyin.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

MORO_SÜRE SONU

MORO_EXPIRATION_WITH_DATA

MORO_EXPIRATION_WITH_FULL_DATA

Anlamı

Bir rapor iletisine dahil edilen özgün iletiden uygulama verisi olmadan, son kullanma tarihi rapor iletilerini isteyin.

Bir rapor iletisine dahil edilen özgün iletiden ilk 100 baytlık uygulama verileriyle birlikte süre bitimi raporu iletileri isteyin.

Bir rapor iletisine dahil edilen özgün iletideki tüm uygulama verileriyle birlikte süre bitimi raporu iletilerini isteyin.

Bu değerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilen rapor seçeneklerine karşılık gelir.

Varsayılan olarak özellik ayarlanmaz.

JMS_IBM_REPORT_NAN**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

İleti

Olumsuz eylem bildirim raporu iletileri isteyin.

Negatif işlem bildirim raporu iletileri istemek için özelliği MORO_NAN olarak ayarlayın. Negatif eylem bildirim raporu iletilerine gerek duymuyorsanız, özelliği ayarlamayın. Varsayılan olarak özellik ayarlanmaz.

MORO_NAN değeri, bir ileti tanımlayıcısının **Report** alanında belirtilebilir bir rapor seçeneğine karşılık gelir.

JMS_IBM_REPORT_PAN

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

Pozitif eylem bildirim raporu iletileri isteyin.

Pozitif eylem bildirim raporu iletileri istemek için özelliği MQRO_PAN olarak ayarlayın. Pozitif eylem bildirim raporu iletilerine gerek duymuyorsanız, özelliği ayarlamayın. Varsayılan olarak özellik ayarlanmaz.

MQRO_PAN değeri, bir ileti tanımlayıcısının **Report** alanında belirtilebilir bir rapor seçeneğine karşılık gelir.

JMS_IBM_REPORT_PASS_CORREL_ID

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

Herhangi bir raporun ya da yanıt iletilerinin ilinti tanıtıcısının, özgün iletilerinin ilinti tanıtıcısıyla aynı olması isteği.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

MQRO_PASS_CORREL_ID

Anlamı

Herhangi bir rapor ya da yanıt iletilerinin ilinti tanıtıcısının, özgün iletilerinin ilinti tanıtıcısıyla aynı olması isteği.

MQRO_COPY_MSG_ID_TO_CORREL_ID

Herhangi bir rapor ya da yanıt iletilerinin ilinti tanıtıcısının, özgün iletilerinin ileti tanıtıcısıyla aynı olması isteği.

Bu değerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilen rapor seçeneklerine karşılık gelir.

Özelliğin varsayılan değeri MQRO_COPY_MSG_ID_TO_CORREL_ID 'dir.

JMS_IBM_REPORT_PASS_MSG_ID

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

Herhangi bir raporun ya da yanıt iletilerinin ileti tanıtıcısının, özgün iletilerinin ileti tanıtıcısıyla aynı olduğunu isteyin.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

MQRO_PASS_MSG_ID

Anlamı

Herhangi bir rapor ya da yanıt iletilerinin ileti tanıtıcısının, özgün iletilerinin ileti tanıtıcısıyla aynı olması isteği.

MQRO_NEW_MSG_ID

Her rapor ya da yanıt iletilerinin için yeni bir ileti tanıtıcısı oluşturulmasına ilişkin istek.

Bu değerler, bir ileti tanımlayıcısının **Rapor** alanında belirlenebilecek rapor seçeneklerine karşılık gelir.

Özelliğin varsayılan değeri MQRO_NEW_MSG_ID 'dir.

JMS_IBM_RETAIN

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

Bu özelliğin ayarlanması, kuyruk yöneticisine bir iletinin Ulaşılan Yayın olarak kabul edileceğini gösterir. Bir abone konulardan ileti aldığı anda, önceki yayınlarda alınan iletilerin ötesinde, abone olduktan hemen sonra ek iletiler alabilir. Bu iletiler, abone olunan konular için isteğe bağlı olarak tutulan yayınlardır. Aboneliğe eşleşen her konu için, alıkonan bir yayın varsa, yayın abone olan ileti tüketicisine teslim edilmek üzere kullanıma sunulur.

RETAIN_YAYIN, bu özellik için geçerli olan tek değerdir. Varsayılan olarak bu özellik ayarlanmaz.

Not: Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir

JMS_IBM_SYSTEM_MESSAGEID

Veri tipi:

Dizgi

Bu özelliğin özelliği:

İleti

İletiyi hizmet bütünleştirme veriyolu içinde benzersiz olarak tanımlayan bir tanıtıcı. Bu özellik salt okunur.

Bu özellik yalnızca uygulama bir hizmet bütünleştirme veriyoluna bağlandığında geçerlidir.

JMSX_APPID

Veri tipi:

Dizgi

Bu özelliğin özelliği:

İleti

İletiyi gönderen uygulamanın adı.

Bu özellik, JMS adı JMSXAppIDolan JMS tanımlı özelliktir. Özellik hakkında daha fazla bilgi için bkz. *Java Message Service Specification, Version 1.1*.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değil.

JMSX_DELIVERY_COUNT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

İleti

İletiyi teslim etme girişimi sayısı.

Bu özellik, JMS adı JMSXDeliveryCountolan JMS tanımlı özelliktir. Özellik hakkında daha fazla bilgi için bkz. *Java Message Service Specification, Version 1.1*.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değil.

JMSX_GRPID

Veri tipi:

Dizgi

Bu özelliğin özelliği:

İleti

İletinin ait olduğu ileti grubunun tanıtıcısı.

Bu özellik, JMS adı JMSXGroupIDolan JMS tanımlı özelliktir. Özellik hakkında daha fazla bilgi için bkz. *Java Message Service Specification, Version 1.1*.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değil.

JMSX_GRPSEQ**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

İleti

İleti grubundaki iletinin sıra numarası.

Bu özellik, JMS adı JMSXGroupSeqolan JMS tanımlı özelliktir. Özellik hakkında daha fazla bilgi için bkz. *Java Message Service Specification, Version 1.1*.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değil.

JMSX_USERID**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

İleti

İletiyi gönderen uygulamayla ilişkili kullanıcı kimliği.

Bu özellik, JMS adı JMSXUserIDolan JMS tanımlı özelliktir. Özellik hakkında daha fazla bilgi için bkz. *Java Message Service Specification, Version 1.1*.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değil.

XMSC_ASYNC_ISTISNALAR**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: ASYNCEXCEPTION

JMS yönetim aracı kısa adı: AEX

Bu özellik, XMS 'nin bir ExceptionListener ' i yalnızca bir bağlantı kesildiğinde mi, yoksa bir XMS API çağırısına zamanuyumsuz olarak bir kural dışı durum oluştuğunda mı bildiğini belirler. Bu özellik, ExceptionListener kaydı olan bu ConnectionFactory ' den oluşturulan tüm Connections için geçerlidir.

Bu özellik için geçerli değerler şunlardır:

XMSC_ASYNC_EXCEPTIONS_ALL

Zamanuyumlu bir API çağırısının kapsamı dışında zamanuyumsuz olarak saptanan herhangi bir kural dışı durum ve tüm bağlantı bozuk kural dışı durumlar ExceptionListener ' a gönderilir.

XMSC_ASYNC_EXCEPTIONS_CONNECTIONBROKEN

ExceptionHandler'ına yalnızca bozuk bir bağlantıyı gösteren kural dışı durumlar gönderilir. Zamanuyumsuz işleme sırasında oluşan diğer kural dışı durumlar ExceptionListener' a raporlanmaz ve bu nedenle uygulamaya bu kural dışı durumlar bildirilmez.

Varsayılan olarak bu özellik XMSC_ASYNC_EXCEPTIONS_ALL olarak ayarlanır.

XMSC_CLIENT_ID

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: CLIENTID

JMS yönetim aracı kısa adı: CID

Bir bağlantıya ilişkin istemci tanıtıcısı.

Bir istemci tanıtıcısı yalnızca yayınlama/abone olma etki alanındaki sürekli abonelikleri desteklemek için kullanılır ve noktadan noktaya etki alanında yoksayılar. İstemci tanıtıcılarını ayarlama hakkında daha fazla bilgi için bkz. [ConnectionFactories](#) ve [Connection](#) nesnelere.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için uygun değildir.

XMSC_CONNECTION_TYPE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Bir uygulamanın bağlandığı ileti sistemi sunucusunun tipi.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer	Anlamı
XMSC_CT_RTT	Bir aracıya gerçek zamanlı bağlantı.
XMSC_CT_WMQ	Bir IBM MQ kuyruk yöneticisine yönelik bağlantı.
XMSC_CT_WPM	Bir WebSphere Application Server service integration busbağlantısı.

Varsayılan olarak özellik ayarlanmaz.

XMSC_DELIVERY_MODE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Hedef

URI ' de kullanılan ad:

kalıcılık (IBM MQ hedefi için)

deliveryMode (WebSphere varsayılan ileti alışverişi sağlayıcısı hedefi için)

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: PERSISTENCE

JMS yönetim aracı kısa adı: PER

Hedefe gönderilen iletilerin teslim kipi.

Özelliğın geçerli değeri şunlardır:

Geçerli değeri

XMSC_DELIVERY_NOT_KALICI

Anlamı

Hedefe gönderilen bir ileti kalıcı değil. İleti üreticisinin varsayılan teslim kipi ya da Gönderme çağrısında belirtilen herhangi bir teslim kipi yoksayılr. Hedef bir IBM MQ kuyruğuyrsa, *DefPersistence* kuyruk özniteliğinin değeri de yoksayılr.

XMSC_DELIVERY_KALICI

Hedefe gönderilen bir ileti kalıcı. İleti üreticisinin varsayılan teslim kipi ya da Gönderme çağrısında belirtilen herhangi bir teslim kipi yoksayılr. Hedef bir IBM MQ kuyruğuyrsa, *DefPersistence* kuyruk özniteliğinin değeri de yoksayılr.

XMSC_DELIVERY_AS_APP

Hedefe gönderilen bir ileti, Gönderme çağrısında belirtilen teslim kipine sahip. Gönderme çağrısında teslim kipi belirtilmezse, bunun yerine ileti üreticisinin varsayılan teslim kipi kullanılır. Hedef bir IBM MQ kuyruğuyrsa, *DefPersistence* kuyruk özniteliğinin değeri yoksayılr.

XMSC_DELIVERY_AS_DEST

Hedef bir IBM MQ kuyruğuyrsa, kuyruğa konan bir ileti, *DefPersistence* kuyruk özniteliğinin değeriyle belirtilen teslim kipine sahiptir. İleti üreticisinin varsayılan teslim kipi ya da Gönderme çağrısında belirtilen herhangi bir teslim kipi yoksayılr.

Hedef bir IBM MQ kuyruğu değilse, anlamı XMSC_DELIVERY_AS_APP ile aynıdır.

Varsayılan değeri: XMSC_DELIVERY_AS_APP.

XMSC_IC_PROVIDER_URL

Veri tipi:

Dizgi

Bu özelliğın özelliğı:

InitialContext

COS adlandırma hizmetinin web hizmetiyle aynı sunucuda olması gerekmeyecek şekilde JNDI adlandırma dizinini bulmak için kullanılır.

XMSC_IC_SECURITY_AUTHENTICATION

Veri tipi:

Dizgi

Bu özelliğın özelliğı:

InitialContext

Java Bağlam arabirimi SECURITY_AUTHENTICATION ögesine dayalı olarak. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

XMSC_IC_SECURITY_CREDENTIALS

Veri tipi:

Dizgi

Bu özelliğin özelliği:

InitialContext

Java Bağlam arabirimi SECURITY_CREDENTIALS. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

XMSC_IC_SECURITY_PRINCIPAL**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

InitialContext

Java Bağlam arabirimi SECURITY_PRINCIPAL değerine dayalı olarak. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

XMSC_IC_SECURITY_PROTOCOL**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

InitialContext

Java Bağlam arabirimi SECURITY_PROTOCOL değerine dayalı Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

XMSC_IC_URL**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

InitialContext

LDAP ve FileSystem bağlamları için, yönetilen nesnelere içeren havuzun adresi.

LDAP ve FileSystem bağlamları için, yönetilen nesnelere içeren havuzun adresi.

XMSC_IS_SUBSCRIPTION_MULTICAST**Veri tipi:**

System.Boolean

Bu özelliğin özelliği:

MessageConsumer

İletilerin WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim edilip edilmediğini gösterir. Bu özellik salt okunur.

İletiler WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim ediliyorsa özelliğin değeri true olur. Ters durumda, değeri false olur.

Bu özellik yalnızca bir aracıya gerçek zamanlı bağlantı için geçerlidir.

XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST**Veri tipi:**

System.Boolean

Bu özelliğin özelliği:

MessageConsumer

İletilerin güvenilir bir hizmet kalitesiyle WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim edilip edilmediğini gösterir. Bu özellik salt okunur.

İletiler, güvenilir bir hizmet kalitesiyle WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim ediliyorsa özelliğin değeri true olur. Ters durumda, değer false olur.

Bu özellik yalnızca bir aracıya gerçek zamanlı bağlantı için geçerlidir.

XMSC_JMS_MAJOR_VERSION

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionMetaVerileri

XMS belirtiminin temel aldığı JMS belirtiminin ana sürüm numarası. Bu özellik salt okunur.

XMSC_JMS_MINOR_VERSION

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionMetaVerileri

XMS belirtiminin temel aldığı JMS belirtiminin ikincil sürüm numarası. Bu özellik salt okunur.

XMSC_JMS_VERSION

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionMetaVerileri

XMS belirtiminin temel aldığı JMS belirtiminin sürüm tanıtıcısı. Bu özellik salt okunur.

XMSC_ANKESÖR_SÜRÜMÜ

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionMetaVerileri

XMS istemcisinin sürüm numarası. Bu özellik salt okunur.

XMSC_MINOR_VERSION

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionMetaVerileri

XMS istemcisinin yayın numarası. Bu özellik salt okunur.

XMSC_PASSWORD

Veri tipi:

Bayt dizisi

Bu özelliğin özelliği:

ConnectionFactory

Bir ileti sistemi sunucusuna bağlanmaya çalışıldığında uygulamanın kimliğini doğrulamak için kullanılacak parola. Parola, [XMSC_USERID](#) özelliğiyle kullanılır.

Varsayılan olarak özellik ayarlanmaz.

Multi

Çoklu platformlar üzerinde IBM MQ ' e bağlanıyorsanız ve bağlantı üreticisinin XMSC_USERID özelliğini ayarlıyorsanız, bu özellik oturum açmış kullanıcının **userid** ile eşleşmelidir. Bu özellikleri ayarlamazsanız, kuyruk yöneticisi varsayılan olarak oturum açmış kullanıcının **userid** değerini kullanır. Tek tek kullanıcılar için daha fazla bağlantı düzeyinde kimlik doğrulaması gerekiyorsa, IBM MQ içinde yapılandırılan bir istemci kimlik doğrulama çıkışı yazabilirsiniz. İstemci kimlik doğrulama çıkışı oluşturma hakkında daha fazla bilgi için bkz. [İstemci uygulaması için kimlik doğrulamasını planlama](#).

z/OS

IBM MQ for z/OS ile bağlantı kurarken kullanıcının kimliğini doğrulamak için bir güvenlik çıkışı kullanmanız gerekir.

XMSC_PRIORITY

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Hedef

URI ' de kullanılan ad:

öncelik

Hedefe gönderilen iletilerin önceliği.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

0 aralığındaki bir tamsayı, en düşük öncelik, 9 en yüksek öncelik

Anlamı

Hedefe gönderilen bir ileti belirtilen önceliğe sahip. İleti üreticisinin varsayılan önceliği ya da Gönderme çağrısında belirtilen herhangi bir öncelik yoksa, Hedef bir IBM MQ kuyruğuysa, **DefPriority** kuyruk özniteliğinin değeri de yoksa, Hedefe gönderilen bir ileti, Gönderme çağrısında belirtilen önceliğe sahip. Gönderme çağrısı öncelik belirtmezse, bunun yerine ileti üreticisinin varsayılan önceliği kullanılır. Hedef bir IBM MQ kuyruğuysa, **DefPriority** kuyruk özniteliğinin değeri yoksa, Hedef bir IBM MQ kuyruğuysa, kuyruğa konan bir ileti, **DefPriority** kuyruk özniteliğinin değeriyle belirtilen önceliğe sahiptir. İleti üreticisinin varsayılan önceliği ya da Gönderme çağrısında belirtilen herhangi bir öncelik yoksa, Hedef bir IBM MQ kuyruğu değilse, anlamı XMSC_PRIORITY_AS_APP ile aynıdır.

XMSC_PRIORITY_AS_APP

Hedefe gönderilen bir ileti, Gönderme çağrısında belirtilen önceliğe sahip. Gönderme çağrısı öncelik belirtmezse, bunun yerine ileti üreticisinin varsayılan önceliği kullanılır. Hedef bir IBM MQ kuyruğuysa, **DefPriority** kuyruk özniteliğinin değeri yoksa, Hedef bir IBM MQ kuyruğuysa, kuyruğa konan bir ileti, **DefPriority** kuyruk özniteliğinin değeriyle belirtilen önceliğe sahiptir. İleti üreticisinin varsayılan önceliği ya da Gönderme çağrısında belirtilen herhangi bir öncelik yoksa, Hedef bir IBM MQ kuyruğu değilse, anlamı XMSC_PRIORITY_AS_APP ile aynıdır.

XMSC_PRIORITY_AS_DEST

Hedef bir IBM MQ kuyruğuysa, kuyruğa konan bir ileti, **DefPriority** kuyruk özniteliğinin değeriyle belirtilen önceliğe sahiptir. İleti üreticisinin varsayılan önceliği ya da Gönderme çağrısında belirtilen herhangi bir öncelik yoksa, Hedef bir IBM MQ kuyruğu değilse, anlamı XMSC_PRIORITY_AS_APP ile aynıdır.

Varsayılan değer: XMSC_PRIORITY_AS_APP.

WebSphere MQ Real-Time Transport ve WebSphere MQ Multicast Transport , bir iletinin önceliğine dayalı olarak herhangi bir işlem gerçekleştirmez.

XMSC_PROVIDER_NAME

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionMetaVerileri

XMS istemcisinin sağlayıcısı. Bu özellik salt okunur.

XMSC_RTT_BROKER_PING_INTERVAL

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

XMS .NET ' in milisaniye cinsinden zaman aralığı, herhangi bir etkinliği saptamak için Gerçek Zamanlı ileti sistemi sunucusuna olan bağlantıyı denetler. Bir etkinlik algılanmazsa, istemci bir ping işlemi başlatır; ping komutuna yanıt algılanmazsa bağlantı kapanır.

Özelliğin varsayılan değeri 30000 'dir.

XMSC_RTT_CONNECTION_PROTOCOL

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Bir aracıya gerçek zamanlı bağlantı için kullanılan iletişim protokolü.

Özelliğin değeri, TCP/IP üzerinden bir aracıya gerçek zamanlı bağlantı anlamına gelen XMSC_RTT_CP_TCP olmalıdır. Varsayılan değer XMSC_RTT_CP_TCP 'dir.

XMSC_RTT_HOST_NAME

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Aracının çalıştığı sistemin anasistem adı ya da IP adresi.

Bu özellik, aracıyı tanımlamak için [XMSC_RTT_PORT](#) özelliğiyle birlikte kullanılır.

Varsayılan olarak özellik ayarlanmaz.

XMSC_RTT_LOCAL_ADDRESS

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bir aracıya gerçek zamanlı bağlantı için kullanılacak yerel ağ arabiriminin anasistem adı ya da IP adresi.

Bu özellik, uygulamanın çalıştığı sistemde iki ya da daha çok ağ arabirimi varsa ve gerçek zamanlı bağlantı için hangi arabirimin kullanılması gerektiğini belirtmeniz gerekiyorsa yararlı olur. Sistemde yalnızca bir ağ arabirimi varsa, yalnızca bu arabirim kullanılabilir. Sistemde iki ya da daha çok ağ arabirimi varsa ve özellik ayarlanmamışsa, arabirim rasgele seçilir.

Varsayılan olarak özellik ayarlanmaz.

XMSC_RTT_MULTICAST

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory ve Hedef

URI ' de kullanılan ad:

Çok hedefli

Bir bağlantı üreticisi ya da hedefi için çoklu yayın ayarı. Yalnızca konu olan bir hedef bu özelliğe sahip olabilir.

Bir uygulama, bir aracıya gerçek zamanlı bir bağlantıyla ilişkili olarak çoklu yayını etkinleştirmek ve çoklu yayın etkinleştirildiyse, aracından iletileri bir ileti tüketicisine teslim etmek için çoklu yayının nasıl kullanılacağını kesin olarak belirtmek için bu özelliği kullanır. Özellik, ileti üreticisinin aracıya ileti gönderme şekliyle etkilenmez.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_RTT_MULTICAST_DISABLED

Anlamı

İletiler, WebSphere MQ Multicast Transport kullanılarak bir ileti tüketicisine teslim edilmez. Bu değer, bir ConnectionFactory nesnesi için varsayılan değerdir.

XMSC_RTT_MULTICAST_ASCF

İletiler, ileti tüketicisiyle ilişkili bağlantı üreticisine ilişkin çoklu yayın ayarına göre bir ileti tüketicisine teslim edilir. Bağlantı üreticisine ilişkin çoklu yayın ayarı, bağlantının oluşturulduğu sırada belirtilir. Bu değer yalnızca Hedef nesne için geçerlidir ve Hedef nesne için varsayılan değerdir.

XMSC_RTT_MULTICAST_ENABLED

Konu, aracıda çoklu yayın için yapılandırıldıysa, iletiler WebSphere MQ Multicast Transport kullanılarak bir ileti tüketicisine teslim edilir. Konu güvenilir çoklu yayın için yapılandırıldıysa, güvenilir bir hizmet kalitesi kullanılır.

XMSC_RTT_MULTICAST_RELIABLE

Konu, aracıda güvenilir çoklu yayın için yapılandırıldıysa, iletiler güvenilir hizmet kalitesiyle WebSphere MQ Multicast Transport kullanılarak bir ileti tüketicisine teslim edilir. Konu güvenilir çoklu yayın için yapılandırılmadıysa, konu için bir ileti tüketicisi oluşturamazsınız.

XMSC_RTT_MULTICAST_NOT_RELIABLE

Konu, aracıda çoklu yayın için yapılandırıldıysa, iletiler WebSphere MQ Multicast Transport kullanılarak bir ileti tüketicisine teslim edilir. Konu güvenilir çoklu yayın için yapılandırılmış olsa da güvenilir bir hizmet kalitesi kullanılmaz.

XMSC_RTT_PORT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Bir aracının gelen istekleri dinlediği kapının numarası. Aracıda, bu kapıyı dinleyecek bir Real-timeInput ya da Real-timeOptimizedAkış iletisi işleme düğümü yapılandırmanız gerekir.

Bu özellik, aracıyı tanımlamak için [XMSC_RTT_HOST_NAME](#) özelliğiyle birlikte kullanılır.

Özelliğin varsayılan değeri XMSC_RTT_DEFAULT_PORT ya da 1506 'dır.

XMSC_TIME_TO_LIVE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Hedef

URI ' de kullanılan ad:

süre bitimi (IBM MQ hedefi için)

timeToLive (WebSphere varsayılan ileti sistemi sağlayıcısı hedefi için)

Hedefe gönderilen iletiler için yaşam süresi.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

0

Artı bir tamsayı

Anlamı

Hedefe gönderilen bir iletinin süresi hiçbir zaman dolmaz.

Hedefe gönderilen bir ileti, milisaniye cinsinden belirtilen yaşam süresini içeriyor. İleti üreticisinin etkin olduğu varsayılan süre ya da Gönderme çağrısında belirtilen herhangi bir zaman yoksayılr.

XMSC_TIME_TO_LIVE_AS_APP

Hedefe gönderilen bir ileti, Gönder çağrısında belirtilen etkin kalma süresini içeriyor. Gönder çağrısı, etkin olmak için zaman belirtmezse, bunun yerine ileti üreticisinin varsayılan yaşam süresi kullanılır.

Varsayılan değer XMSC_TIME_TO_LIVE_AS_APP 'dir.

XMSC_USERID

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bir ileti sistemi sunucusuna bağlanmayı denediğinde uygulamanın kimliğini doğrulamak için kullanılacak kullanıcı kimliği. Kullanıcı kimliği [XMSC_PASSWORD](#) özelliğiyle birlikte kullanılır.

Varsayılan olarak özellik ayarlanmaz.

Multi IBM MQ for Multiplatforms' e bağlanıyorsanız ve bağlantı üreticisinin XMSC_USERID özelliğini ayarlarsanız, bu özellik oturum açmış kullanıcının **userid** ile eşleşmelidir. Bu özellikleri ayarlamazsanız, kuyruk yöneticisi varsayılan olarak oturum açmış kullanıcının **userid** değerini kullanır. Tek tek kullanıcılar için daha fazla bağlantı düzeyinde kimlik doğrulaması gerekiyorsa, IBM MQ içinde yapılandırılmış bir istemci kimlik doğrulama çıkışı yazabilirsiniz. İstemci kimlik doğrulama çıkışı oluşturma hakkında daha fazla bilgi için bkz. [İstemci uygulaması için kimlik doğrulamasını planlama](#).

z/OS IBM MQ for z/OS ile bağlantı kurarken kullanıcının kimliğini doğrulamak için bir güvenlik çıkışı kullanmanız gerekir.

XMSC_Sürümü

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionMetaVerileri

cliXMS' nin sürüm tanıtıcısı. Bu özellik salt okunur.

V 9.3.0 XMSC_WMQ_BALANCING_APPLICATION_TYPE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_WMQ_BALANCING_APPLICATION_TYPE_SIMPLE

Anlamı

Basit dengeleme; Tek biçimli kümelerde uygulama yeniden dengelemesi konusunda anlatılanlara ek olarak belirli bir kural uygulanmaz. Bu varsayılan değerdir.

XMSC_WMQ_BALANCING_APPLICATION_TYPE_REQUEST_REPLY

İstek-Yanıt dengeleme; her MQPUT çağrısından sonra, yanıt iletisi için eşleşen bir MQGET çağrısı beklenir. Böyle bir ileti alınıncaya ya da istek iletisi EXPIRY aşıncaya kadar dengeleme geciktirildi

Buna ek olarak, bu özellik `client.ini` dosyasında ayarlanabilir. Seçiş sırası:

1. Uygulamada ayarlanan özellikler
2. `mqclient.ini` dosyasında Application stanza adlı eşleştirme.
3. `mqclient.ini` dosyasındaki Application defaults stanza (Uygulama varsayılan kısmı).

V 9.3.0 XMSC_WMQ_BALANCING_OPTIONS

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_WMQ_BALANCING_OPTIONS_NONE

İlgili değer

Seçenek belirlenmedi. Bu varsayılan değerdir

XMSC_WMQ_BALANCING_OPTIONS_IGNORE_TRANSACTION

Bu seçeneğin ayarlanması, bir hareketin ortasında bile uygulamaların yeniden dengelenmesini sağlar.

Buna ek olarak, bu özellik `client.ini` dosyasında ayarlanabilir. Seçiş sırası:

1. Uygulamada ayarlanan özellikler
2. `mqclient.ini` dosyasında Application stanza adlı eşleştirme.
3. `mqclient.ini` dosyasındaki Application defaults stanza (Uygulama varsayılan kısmı).

V 9.3.0 XMSC_WMQ_BALANCING_TIMEOUT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Özelliğın geçerli değeri şunlardır:

Geçerli değeri	Anlamı
XMSC_WMQ_BALANCING_TIMEOUT_IMMEDIATE	Anında zamanaşımı oluştı
XMSC_WMQ_BALANCING_TIMEOUT_AS_DEFAULT	Ayarlanan varsayılan zamanaşımı değeri. Bu varsayılan değeri
XMSC_WMQ_BALANCING_TIMEOUT_ASLA	Zamanaşımı oluşmuyor

Not: Yalnızca tanımlı değerilerden bir değeri ya da 0-999999999 saniye arasında bir değeri belirtmeniz gerekir.

Buna ek olarak, bu özellik `client.ini` dosyasında ayarlanabilir. Seçiş sırası:

1. Uygulamada ayarlanan özellikler
2. `mqclient.ini` dosyasında Application stanza adlı eşleştirme.
3. `mqclient.ini` dosyasındaki Application defaults stanza (Uygulama varsayılan kısmı).

XMSC_WMQ_BROKER_CONTROLQ

Veri tipi:

Dizgi

Bu özelliğın özelliğı:

ConnectionFactory

Bir aracı tarafından kullanılan denetim kuyruğının adı.

Özelliğın varsayılan değeri SYSTEM.BROKER.CONTROL.QUEUE.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

XMSC_WMQ_BROKER_PUBQ

Veri tipi:

Dizgi

Bu özelliğın özelliğı:

ConnectionFactory

Uygulamaların yayınladıkları iletileri gönderdikleri bir aracı tarafından izlenen kuyruğın adı.

Özelliğın varsayılan değeri SYSTEM.BROKER.DEFAULT.STREAM.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

XMSC_WMQ_BROKER_QMGR

Veri tipi:

Dizgi

Bu özelliğın özelliğı:

ConnectionFactory

Aracının bağılı olduğı kuyruk yöneticisinin adı.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

XMSC_WMQ_BROKER_SUBQ

Veri tipi:

Dizgi

Bu özelliğın özelliğı:

ConnectionFactory

Sürekli olmayan bir ileti tüketicisine ilişkin abone kuyruğunun adı.

Abone kuyruğunun adı aşağıdaki karakterlerle başlamalıdır:

SYSTEM.JMS.ND.

Sürekli olmayan tüm ileti tüketicilerinin bir abone kuyruğunu paylaşmasını istiyorsanız, paylaşılan kuyruğun tam adını belirtin. Bir uygulamanın kalıcı olmayan bir ileti tüketicisi oluşturabilmesi için, belirtilen ada sahip bir kuyruk var olmalıdır.

Sürekli olmayan her ileti tüketicisinin iletileri kendi dışlayıcı abone kuyruğundan almasını istiyorsanız, yıldız işaretiyle (*) biten bir kuyruk adı belirleyin. Daha sonra, bir uygulama dayanıklı olmayan bir ileti tüketicisi oluşturduğunda, XMS istemcisi ileti tüketicisinin özel kullanımını için dinamik bir kuyruk oluşturur. XMS istemcisi, dinamik kuyruğu yaratmak için kullanılan nesne tanımlayıcıdaki **DynamicQName** alanının içeriğini ayarlamak için özelliğin değerini kullanır.

Özelliğin varsayılan değeri SYSTEM.JMS.ND.SUBSCRIBER.QUEUE, XMS ' un varsayılan olarak paylaşılan kuyruk yaklaşımını kullandığı anlamına gelir.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

XMSC_WMQ_BROKER_VERSION

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory ve Hedef

URI ' de kullanılan ad:

brokerVersion

Uygulama tarafından bir bağlantı ya da hedef için kullanılan aracının tipi. Yalnızca konu olan bir hedef bu özelliğe sahip olabilir.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_WMQ_BROKER_V1

Anlamı

Uygulama bir IBM MQ yayınlama/abone olma aracı kullanıyor.

IBM MQ yayınlama/abone olma seçeneğinden WebSphere Message Broker ' a geçiş yaptıysanız, ancak uygulamayı değiştirmediyse de uygulama bu değeri kullanabilir.

XMSC_WMQ_BROKER_V2

Uygulama IBM Integration Bus aracısını kullanıyor.

XMSC_WMQ_BROKER_BELIRTILMEDI

Aracı geçireldikten sonra bu özelliği, RFH2 üstbilgilerinin artık kullanılmayacak şekilde ayarlayın. Geçişten sonra bu özellik artık ilgili değildir.

Bir connectionfactory için varsayılan değer XMSC_WMQ_BROKER_UNDEFINDIR, ancak varsayılan olarak özellik bir hedef için ayarlanmaz. Bir hedef için özelliğin ayarlanması, bağlantı üreticisi özelliği tarafından belirtilen herhangi bir değeri geçersiz kılar.

XMSC_WMQ_CCDTURL

Veri tipi:

System.String

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: CCDTURL

JMS yönetim aracı kısa adı: CCDT

İstemci kanal tanımlama çizelgesini içeren dosyanın adını ve yerini tanıtan ve dosyaya nasıl erişilebileceğini belirten URL.

Varsayılan olarak bu özellik ayarlanmaz.

XMSC_WMQ_CCSID

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Hedef

URI ' de kullanılan ad:

CCSID

XMS istemcisi iletiyi hedefe ilettiğinde, iletinin gövdesindeki karakter verileri dizgilerinin içinde olduğu kodlanmış karakter takımının ya da kod sayfasının tanıtıcısı (CCSID). Tek bir ileti için ayarlanırsa, JMS_IBM_CHARACTER_SET özelliği bu özellik tarafından hedef için belirtilen CCSID ' yi geçersiz kılar.

Özelliğin varsayılan değeri 1208 'dir.

Bu özellik, hedeften alınan iletilerle değil, yalnızca hedefe gönderilen iletilerle ilgilidir.

XMSC_WMQ_CHANNEL

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: CHANNEL

JMS yönetim aracı kısa adı: CHAN

Bağlantı için kullanılacak kanalın adı.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında geçerlidir.

XMSC_WMQ_CLIENT_RECONNECT_OPTIONS

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: CLIENTRECONNECTOPTIONS

JMS yönetim aracı kısa adı: CROPT

Bu özellik, bu üretici tarafından yaratılan yeni bağlantılara ilişkin istemci yeniden bağlanma seçeneklerini belirler. XMSC ' de bulunur ve aşağıdakilerden biridir:

- WMQ_CLIENT_RECONNECT_AS_DEF (varsayılan). `mqcClient.ini` dosyasında belirtilen değeri kullanın. Değeri, Kanal kısmı içindeki **DefRecon** özelliğini kullanarak ayarlayın. Aşağıdakilerden birine ayarlanabilir:
 1. EVET. WMQ_CLIENT_RECONNECT seçeneği olarak davranır
 2. No. Varsayılan. Herhangi bir yeniden bağlanma seçeneği belirtmez
 3. QMGR. WMQ_CLIENT_RECONNECT_Q_MGR seçeneği olarak davranır
 4. Geçersiz kılındı. WMQ_CLIENT_RECONNECT_DISABLED seçeneği olarak davranır

- WMQ_CLIENT_RECONNECT. Bağlantı adı listesinde belirtilen kuyruk yöneticilerinden herhangi birine yeniden bağlanın.
- WMQ_CLIENT_RECONNECT_Q_MGR. Başlangıçta bağlı olduğu aynı kuyruk yöneticisine yeniden bağlanır. Bağlanmaya çalıştığı kuyruk yöneticisinin (bağlantı adı listesinde belirtilir), başlangıçta bağlı olduğu kuyruk yöneticisiyle farklı bir QMID 'si varsa MQRC_RECONNECT_QMID_MISMATCH değerini döndürür.
- WMQ_CLIENT_RECONNECT_DISABLED. Yeniden bağlantı devre dışı bırakıldı.

XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: CLIENTRECONNECTTIMEOUT

JMS yönetim aracı kısa adı: CRT

XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT özelliği yalnızca Yönetilen XMS .NET istemcisi için geçerlidir.

Bu " zellik, bir istemci başlantsnn yeniden başlanmaya ıalıt ş sreyi saniye cinsinden belirler.

Bu süre boyunca yeniden bağlanma girişiminde bulunulduktan sonra, istemci MQRC_RECONNECT_FAILED ile başarısız olur. Bu özelliğin varsayılan ayarı XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT_DEFAULT.

Bu özelliğin varsayılan değeri 1800 'dür.

XMSC_WMQ_CONNECTION_MODE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Bir uygulamanın kuyruk yöneticisine bağlanma kipi.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer	Anlamı
XMSC_WMQ_CM_BINDINGS	En iyi başarıml için bağ tanımlama kipinde bir kuyruk yöneticisine yönelik bağlantı. Bu değer, C/C + + için varsayılan değerdir.
XMSC_WMQ_CM_CLIENT	Tam olarak yönetilen bir yığın sağlamak için istemci kipinde bir kuyruk yöneticisiyle bağlantı. Bu değer, .NET için varsayılan değerdir.
XMSC_WMQ_CM_CLIENT_UNMANAGED (yalnızca .NET için)	Yönetilmeyen bir istemci yığınına zorlayan bir kuyruk yöneticisine yönelik bağlantı.

XMSC_WMQ_CONNECTION_NAME_LIST

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: CONNECTIONNAMELIST

JMS yönetim aracı kısa adı: CNLIST

Bu özellik, istemcinin bağlantısı kesildikten sonra yeniden bağlanmaya çalıştığı anasistemleri belirtir.

Bağlantı adı listesi, anasistem/ip kapı çiftlerinin virgülle ayrılmış listesidir. Bu özelliğin varsayılan ayarı WMQ_CONNECTION_NAME_LIST_DEFAULT değeridir.

Örneğin,127.0.0.1 (1414) ,host2.example.com(1400)

Bu özelliğin varsayılan ayarı localhost (1414)' tur.

XMSC_WMQ_DUR_SUBQ

Veri tipi:

Dizgi

Bu özelliğin özelliği:

Hedef

Hedeften ileti alan sürekli bir aboneye ilişkin abone kuyruğunun adı. Yalnızca konu olan bir hedef bu özelliğe sahip olabilir.

Abone kuyruğunun adı aşağıdaki karakterlerle başlamalıdır:

SYSTEM.JMS.D.

Tüm sürekli abonelerin bir abone kuyruğunu paylaşmasını istiyorsanız, paylaşılan kuyruğun tam adını belirtin. Bir uygulamanın kalıcı bir abone yaratabilmesi için, belirtilen ada sahip bir kuyruk var olmalıdır.

Her sürekli abonenin kendi dışlayıcı abone kuyruğundan ileti almasını istiyorsanız, yıldız işaretiyle (*) biten bir kuyruk adı belirleyin. Daha sonra, bir uygulama sürekli bir abone yarattığında, XMS istemcisi kalıcı abone tarafından özel olarak kullanılmak üzere dinamik bir kuyruk yaratır. XMS istemcisi, dinamik kuyruğu yaratmak için kullanılan nesne tanımlayıcıdaki **DynamicQName** alanının içeriğini ayarlamak için özelliğin değerini kullanır.

Özelliğin varsayılan değeri SYSTEM.JMS.D.SUBSCRIBER.QUEUE, XMS ' un varsayılan olarak paylaşılan kuyruk yaklaşımını kullandığı anlamına gelir.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

XMSC_WMQ_ENCODING

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Hedef

XMS istemcisi iletiyi hedefe ilettiğinde, iletinin gövdesindeki sayısal verilerin nasıl gösterileceği. Tek bir ileti için ayarlanırsa, JMS_IBM_ENCODING özelliği bu özellik tarafından hedef için belirtilen kodlamayı geçersiz kılar. Özellik, ikili tamsayıların, paketlenmiş ondalık tamsayıların ve kayan noktalı sayıların gösterimini belirtir.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **Encoding** alanında belirtilebilir değerlerle aynıdır.

Bir uygulama, özelliği ayarlamak için aşağıdaki adlandırılmış değişmezleri kullanabilir:

Adlandırılmış değişmez

MQENC_INTEGER_NORMAL

MQENC_INTEGER_REVERSED

MQENC_DECIMAL_NORMAL

MQENC_DECIMAL_REVERSED

MQENC_FLOAT_IEEE_NORMAL

MQENC_FLOAT_IEEE_REVERSED

MQENC_FLOAT_S390

Anlamı

Normal tamsayı kodlaması

Tersine çevrilmiş tamsayı kodlaması

Normal paketlenmiş ondalık kodlama

Paketlenmiş onlu kodlama tersine çevrildi

Olağan IEEE kayan nokta kodlaması

Ters IEEE kayan nokta kodlaması

z/OS mimarisi kayan nokta kodlaması

Adlandırılmış değişmez**Anlamı**

MQENC_NATIVE

Yerli makine kodlaması

Özelliğe ilişkin bir değer oluşturmak için, uygulama bu sabitlerden üçünü aşağıdaki gibi ekleyebilir:

- İkili tamsayıların gösterimini belirtmek için adı MQENC_INTEGER ile başlayan bir değişmez
- Paketlenmiş ondalık tamsayıların gösterimini belirtmek için adı MQENC_DECIMAL ile başlayan bir değişmez
- Kayar noktalı sayıların gösterimini belirtmek için adı MQENC_FLOAT ile başlayan bir değişmez

Diğer bir seçenek olarak, uygulama özelliği değeri ortama bağımlı olan MQENC_NATIVE olarak ayarlayabilir.

Özelliğin varsayılan değeri MQENC_NATIVE değeridir.

Bu özellik, hedeften alınan iletilerle değil, yalnızca hedefe gönderilen iletilerle ilgilidir.

XMSC_WMQ_FAIL_IF_QUIESCE**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

ConnectionFactory ve Hedef

URI ' de kullanılan ad:

failIfSusturma

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: FAILIFQUIESCE

JMS yönetim aracı kısa adı: FIQ

Uygulamanın bağlı olduğu kuyruk yöneticisi susturma durumundaysa, belirli yöntemlere yapılan çağrılar başarısız olup olmadığını belirler.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer**Anlamı**

XMSC_WMQ_FIQ_YES

Kuyruk yöneticisi susturma durumundaysa, belirli yöntemlere yönelik çağrılar başarısız olur. Uygulama kuyruk yöneticisinin durmakta olduğunu saptadığında, uygulama hemen görevini tamamlayabilir ve bağlantıyı kapatarak kuyruk yöneticisinin durmasına izin verebilir.

XMSC_WMQ_FIQ_NO

Kuyruk yöneticisi susturma durumunda olduğundan yöntem çağrıları başarısız olmaz. Bu değeri belirtirseniz, uygulama kuyruk yöneticisinin susturulduğunu saptayamaz. Uygulama, kuyruk yöneticisine ilişkin işlemleri gerçekleştirmeye devam edebilir ve bu nedenle kuyruk yöneticisinin durmasını önleyebilir.

Bir bağlantı üreticisinin varsayılan değeri XMSC_WMQ_FIQ_YES 'dir, ancak varsayılan olarak özellik bir hedef için ayarlanmaz. Bir hedef için özelliğin ayarlanması, bağlantı üreticisi özelliği tarafından belirtilen herhangi bir değeri geçersiz kılar.

XMSC_WMQ_MESSAGE_BODY**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

Hedef

Bu özellik, bir XMS uygulamasının bir IBM MQ iletisinin MQRFH2 ' sini ileti bilgi yükünün (ileti gövdesinin bir parçası olarak) bir parçası olarak işleyip işlemediğini belirler.

Not: Bir hedefe ileti gönderirken, XMSC_WMQ_MESSAGE_BODY özelliği var olan XMS Hedef özelliği XMSC_WMQ_TARGET_CLIENT ' in yerini alır.

Bu özellik için geçerli değerler şunlardır:

XMSC_WMQ_MESSAGE_BODY_JMS

Alma: Gelen XMS ileti tipi ve gövdesi, alınan IBM MQ iletisinde MQRFH2 (varsa) ya da MQMD ' nin (MQRFH2 yoksa) içeriğine göre belirlenir.

Gönder: Giden XMS ileti gövdesi, XMS İleti özelliklerini ve üstbilgi alanlarını temel alan önekli ve otomatik oluşturulan MQRFH2 üstbilgisini içerir.

XMSC_WMQ_MESSAGE_BODY_MQ

Al: Gelen XMS ileti tipi, alınan IBM MQ iletisinin içeriğinden ya da alınan MQMD ' nin biçim alanından bağımsız olarak her zaman ByteMessage olur. XMS ileti gövdesi, temel ileti alışverişi sağlayıcısı API çağrısının döndürdüğü, değiştirilmemiş ileti verileridir. İleti gövdesindeki verilerin karakter kümesi ve kodlaması, MQMD ' nin CodedCharSetId ve Encoding alanlarından saptanır. İleti gövdesindeki verilerin biçimi, MQMD ' nin Biçim alanı tarafından belirlenir.

Gönder: Giden XMS ileti gövdesi, uygulama bilgi yükünü olduğu gibi içerir ve gövdeye otomatik oluşturulan IBM MQ üstbilgisi eklenmez.

XMSC_WMQ_MESSAGE_BODY_BELIRTIHMEDI

Al: XMS istemcisi bu özellik için uygun bir değer saptar. Alma yolunda bu değer WMQ_MESSAGE_BODY_JMS özellik değeridir.

Gönder: XMS istemcisi bu özellik için uygun bir değer saptar. Gönderme yolunda bu değer XMSC_WMQ_TARGET_CLIENT özellik değeridir.

Varsayılan olarak bu özellik XMSC_WMQ_MESSAGE_BODY_BELIRTIHMEDI olarak ayarlanır.

XMSC_WMQ_MQMD_MESSAGE_CONTEXT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Hedef

XMS uygulaması tarafından ayarlanacak ileti bağlamı düzeyini belirler. Bu özelliğin yürürlüğe girmesi için uygulamanın uygun bağlam yetkisiyle çalışması gerekir.

Bu özellik için geçerli değerler şunlardır:

XMSC_WMQ_MDCTX_DEFAULT

Giden iletiler için, MQOPEN API çağrısı ve MQPMO yapısı belirttik ileti bağlamı seçenekleri belirtmiyor.

XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT

MQOPEN API çağrısı MQOO_SET_IDENTITY_CONTEXT ileti bağlamı seçeneğini belirtiyor ve MQPMO yapısı MQPMO_SET_IDENTITY_CONTEXT ögesini belirtiyor.

XMSC_WMQ_MDCTX_SET_ALL_CONTEXT

MQOPEN API çağrısı MQOO_SET_ALL_CONTEXT ileti bağlamı seçeneğini belirtiyor ve MQPMO yapısı MQPMO_SET_ALL_CONTEXT ögesini belirtiyor.

Varsayılan olarak bu özellik XMSC_WMQ_MDCTX_DEFAULT olarak ayarlanır.

Not: Bir uygulama WebSphere Application Server service integration bus' a bağlandığında bu özellik ilgili değildir.

Aşağıdaki özellikler, istenmesi için bir ileti gönderilirken XMSC_WMQMD_MESSAGE_CONTEXT özelliğinin XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT özellik değerine ya da XMSC_WMQ_MDCTX_SET_ALL_CONTEXT özellik değerine ayarlanmasını gerektirir:

- JMS_IBM_MQMD_USERIDENTIFIER
- JMS_IBM_MQMD_ACCOUNTINGTOKEN

- JMS_IBM_MQMD_APPLIDENTITYDATA

Aşağıdaki özellikler, istenmesi için bir ileti gönderilirken XMSC_WMQ_MQMD_MESSAGE_CONTEXT özelliğinin XMSC_WMQ_MDCTX_SET_ALL_CONTEXT özellik değerine ayarlanmasını gerektirir:

- 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

Veri tipi:

System.Int32

Bu özelliğın özelliğı:

Hedef

Bu özellik, bir XMS uygulamasının MQMD alanlarının değerlerini alıp almayacağını belirler.

Bu özellik için geçerli değerler şunlardır:

XMSC_WMQ_READ_ENABLED_NO

İleti gönderilirken, gönderilen bir iletideki JMS_IBM_MQMD* özellikleri MQMD ' deki güncellenen alan değerlerini yansıtacak şekilde güncellenmez.

İleti alınırken, JMS_IBM_MQMD* özelliklerinin bazıları ya da tümü gönderen tarafından ayarlanmış olsa da, alınan bir iletide hiçbiri kullanılamaz.

XMSC_WMQ_READ_ENABLED_YES

İleti gönderilirken, gönderilen bir iletideki tüm JMS_IBM_MQMD* özellikleri, gönderenin belirttik olarak ayarlamadığı özellikler de içinde olmak üzere, MQMD ' deki güncellenen alan değerlerini yansıtacak şekilde güncellenir.

İleti alınırken, gönderenin belirttik olarak ayarlamadığı özellikler de içinde olmak üzere, alınan bir iletide tüm JMS_IBM_MQMD* özellikleri kullanılabilir.

Varsayılan olarak bu özellik XMSC_WMQ_READ_ENABLED_NO olarak ayarlanır.

XMSC_WMQ_MQMD_WRITE_ENABLED

Veri tipi:

System.Int32

Bu özelliğın özelliğı:

Hedef

Bu özellik, bir XMS uygulamasının MQMD alanlarının değerlerini ayarlayıp ayarlayamayacağını belirler.

Bu özellik için geçerli değerler şunlardır:

XMSC_WMQ_WRITE_ENABLED_NO

Tüm JMS_IBM_MQMD* özellikleri yoksayılr ve değerleri temel MQMD yapısına kopyalanmaz.

XMSC_WMQ_WRITE_ENABLED_YES

JMS_IBM_MQMD* özellikleri işlendi. Değerleri temeldeki MQMD yapısına kopyalanır.

Varsayılan olarak bu özellik XMSC_WMQ_WRITE_ENABLED_NO olarak ayarlanır.

XMSC_WMQ_PUT_ASYNC_ALLOWED

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Hedef

Bu özellik, ileti üreticilerinin bu hedefe ileti göndermek için zamanuyumsuz girişler kullanmalarına izin verilip verilmeyeceğini belirler.

Bu özellik için geçerli değerler şunlardır:

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_DEST

Kuyruğa ya da konu tanımlamasına gönderme yaparak, zamanuyumsuz girişlere izin verilip verilmediğini saptayın.

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_Q_DEF

Kuyruk tanımlamasına gönderme yaparak, zamanuyumsuz girişlere izin verilip verilmediğini saptayın.

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_TOPIC_DEF

Konu tanımlamasına başvurarak zamanuyumsuz girişlere izin verilip verilmediğini belirleyin.

XMSC_WMQ_PUT_ASYNC_ALLOWED_DISABLED

Zamanuyumsuz girişlere izin verilmez.

XMSC_WMQ_PUT_ASYNC_ALLOWED_ENABLED

Zamanuyumsuz girişlere izin verilir.

Varsayılan olarak bu özellik XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_DEST olarak ayarlanır.

Not: Bir uygulama WebSphere Application Server service integration bus' a bağlanırken bu özellik ilgili değildir.

XMSC_WMQ_READ_AHEAD_ALLOWED**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

Hedef

Bu özellik, ileti tüketicilerinin ve kuyruk tarayıcılarının, bu hedeften kalıcı olmayan, işlemsel olmayan iletileri almadan önce iç arabelleğe almak için önden okuma özelliğini kullanıp kullanamayacağını belirler.

Bu özellik için geçerli değerler şunlardır:

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_Q_DEF

Kuyruk tanımlamasına başvurarak önden okumaya izin verilip verilmediğini belirleyin.

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_KONU_DEF

Konu tanımlamasına başvurarak önden okumaya izin verilip verilmediğini belirleyin.

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST

Kuyruğa ya da konu tanımlamasına başvurarak önden okumaya izin verilip verilmediğini belirleyin.

XMSC_WMQ_READ_AHEAD_ALLOWED_DISABLED

İletiler tüketilirken ya da iletilere göz atılırken önden okumaya izin verilmez.

XMSC_WMQ_READ_AHEAD_ALLOWED_ENABLED

Önden okumaya izin verilir.

Varsayılan olarak bu özellik XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST olarak ayarlanır.

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

Hedef

Bu özellik, zamanuyumsuz bir ileti dinleyicisine teslim edilen iletiler için, ileti tüketicisi kapatıldığında iç önden okuma arabelleğindeki iletilere ne olacağını belirler.

Bu "zellik, bir hedeften ileti alırken kapanan kuyruk seçeneklerinin belirlenmesinde ve bir hedefe ileti g" rntlerken uygulanamaz.

Kuyruk tarayıcıları için bu özellik yoksayılr; iletilere göz atma işlemi sırasında kuyruklar kullanılabilir.

Bu özellik için geçerli değerler şunlardır:

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT

Yalnızca yürürlükteki ileti dinleyici çağrısı dönmenden önce tamamlanır; iletiler iç önden okuma arabelleğinde bırakılır ve daha sonra atılır.

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_ALL

İç önden okuma arabelleğindeki tüm iletiler, geri dönmenden önce uygulama iletilisi dinleyicisine teslim edilir.

Varsayılan olarak bu özellik XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT olarak ayarlanır.

Not:**Olağandışı uygulama sonlandırması**

Bir XMS uygulaması aniden sona erdiğinde, önden okuma arabelleğindeki tüm iletiler kaybolur.

İşlemlere ilişkin çıkarımlar

Uygulamalar hareketleri kullandığında önden okuma devre dışı bırakılır. Bu nedenle, uygulama işlemli oturumları kullandığında davranışta herhangi bir fark görmez.

Oturum bilgisi kiplerinin etkileri

Alındı bildirim kipleri XMSC_AUTO_ALINDI bildirim ya da XMSC_DUPS_OK_ALINDI bildirim olduğunda, işlem yapılmayan bir oturumda önden okuma etkinleştirilir. Oturum alındı bildirim kipi, işlemden geçirilmiş ya da işlemden geçirilmemiş oturumlardan bağımsız olarak XMSC_CLIENT_ALINDI bildirim ise, önden okuma geçersiz kılınır.

Kuyruk tarayıcıları ve kuyruk tarayıcısı seçicileri için sonuçlar

XMS uygulamalarında kullanılan kuyruk tarayıcıları ve kuyruk tarayıcısı seçicileri, önden okuma özelliğinden performans avantajını elde eder. Kuyruk tarayıcısının kapatılması başarıyı düşürmez; ileti kuyrukta daha fazla işlem için kullanılabilir. Kuyruk tarayıcıları ve kuyruk tarayıcısı seçicileri için, önden okuma başarımının sağladığı avantajlar dışında başka bir etki yoktur.

XMSC_WMQ_ANASISTEMI_ADI**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: HOSTNAME

JMS yönetim aracı kısa adı: HOST

Kuyruk yöneticisinin çalıştığı sistemin anasistem adı ya da IP adresi.

Bu özellik yalnızca, bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında kullanılır. Bu özellik, kuyruk yöneticisini tanımlamak için [XMSC_WMQ_PORT](#) özelliğiyle birlikte kullanılır.

Özelliğin varsayılan değeri localhostdeğeridir.

XMSC_WMQ_LOCAL_ADDRESS**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: LOCALADDRESS

JMS yönetim aracı kısa adı: LA

Kuyruk yöneticisine yönelik bir bağlantı için bu özellik, kullanılacak yerel ağ arabirimini ya da kullanılacak yerel kapı ya da yerel kapı aralığını ya da her ikisini de belirtir.

Özelliğin değeri, aşağıdaki biçime sahip bir dizedir:

[*anasistem_adi*] [(*alt_kapı*) [,*yüksek_kapı*]]

Değişkenlerin anlamları şunlardır:

anasistem_adi

Bağlantı için kullanılacak yerel ağ arabiriminin anasistem adı ya da IP adresi.

Bu bilgilerin sağlanması, uygulamanın çalıştığı sistemde iki ya da daha çok ağ arabirimi varsa ve bağlantı için hangi arabirimin kullanılması gerektiğini belirleyebilmeniz gerekiyorsa gereklidir.

Sistemde yalnızca bir ağ arabirimi varsa, yalnızca bu arabirim kullanılabilir. Sistemde iki ya da daha çok ağ arabirimi varsa ve hangi arabirimin kullanılması gerektiğini belirlemezseniz, arabirim rasgele seçilir.

alt_kapı

Bağlantı için kullanılacak yerel kapının numarası.

high_port da belirtilirse, *low_port* bir kapı numarası aralığındaki en düşük kapı numarası olarak yorumlanır.

yüksek_kapı

Bir kapı numarası aralığındaki en yüksek kapı numarası. Bağlantı için belirtilen aralıktaki kapılardan biri kullanılmalıdır.

Dizginin uzunluk üst sınırı 48 karakterdir.

Aşağıda, özelliğin geçerli değerlerine ilişkin bazı örnekler verilmiştir:

JUPITER

9.20.4.98

JUPITER (1000)

9.20.4.98(1000,2000)

(1000)

(1000,2000)

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında geçerlidir.

XMSC_WMQ_MESSAGE_SELECTION**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

İleti seçiminin XMS istemcisi tarafından mı, yoksa aracı tarafından mı yapılacağını belirler.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer	Anlamı
XMSC_WMQ_MSEL_CLIENT	İleti seçimi XMS istemcisi tarafından yapılır.
XMSC_WMQ_MSEL_BROKER	İleti seçimi aracı tarafından yapılır.

Varsayılan değer XMSC_WMQ_MSEL_CLIENT değeridir.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir. XMSC_WMQ_BROKER_VERSION özelliği XMSC_WMQ_BROKER_V1 olarak ayarlanırsa, aracı tarafından ileti seçimi desteklenmez.

XMSC_WMQ_MSG_BATCH_SIZE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Zamanuyumsuz ileti teslimi kullanılırken bir toplu işte kuyruktan alınacak ileti sayısı üst sınırı.

Bir uygulama, belirli koşullar altında zamanuyumsuz ileti teslimini kullanırken, XMS istemcisi her iletiyi ayrı olarak uygulamaya iletmeden önce kuyruktan bir ileti kümesi alır. Bu özellik, toplu işte bulunabilecek ileti sayısı üst sınırını belirtir.

Özelliğin değeri pozitif bir tamsayıdır ve varsayılan değer 10 'dur. Yalnızca ele almanız gereken belirli bir performans sorununuz varsa özelliği farklı bir değere ayarlayın.

Bir uygulama bir ağ üzerinden bir kuyruk yöneticisine bağlıysa, bu özelliğin değerini yükseltmek ağ genel giderlerini ve yanıt sürelerini azaltabilir, ancak iletileri istemci sisteminde saklamak için gereken bellek miktarını artırabilir. Ters durumda, bu özelliğin değerinin düşürülmesi ağ ek yüklerini ve yanıt sürelerini artırabilir, ancak iletileri saklamak için gereken bellek miktarını azaltabilir.

XMSC_WMQ_POLLING_INTERVAL

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Bir oturum içindeki her ileti dinleyicinin kuyruğunda uygun bir ileti yoksa, bu değer, her ileti dinleyicisinin kuyruğundan ileti almaya yeniden çalışmadan önce geçmesi gereken milisaniye cinsinden aralık üst sınırınıdır.

Sık sık oturumdaki ileti dinleyicilerden herhangi biri için uygun bir ileti yoksa, bu özelliğin değerini artırmayı düşünebilirsiniz.

Özelliğin değeri pozitif bir tamsayıdır. Varsayılan değer 5000 'dir.

XMSC_WMQ_PORT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: PORT

JMS yönetim aracı kısa adı: PORT

Kuyruk yöneticisinin gelen istekleri dinlediği kapının numarası.

Bu özellik yalnızca, bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında kullanılır. Bu özellik, kuyruk yöneticisini tanımlamak için XMSC_WMQ_HOST_NAME özelliğiyle birlikte kullanılır.

Özelliğin varsayılan değeri XMSC_WMQ_DEFAULT_CLIENT_PORT ya da 1414 'tür.

XMSC_WMQ_PROVIDER_VERSION

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulamanın bağlanmayı planladığı kuyruk yöneticisinin sürümü, yayını, değişiklik düzeyi ve düzeltme paketi. Bu özellik için geçerli değerler şunlardır:

- Belirtilmedi

Ya da aşağıdaki biçimlerden birindeki bir dizgiyi

- V.R.M.F
- V.R.M
- V.R
- V

Burada V, R, M ve F sıfırdan büyük ya da sıfıra eşit tamsayı değerleridir.

Varsayılan olarak bu özellik "belirtilmemiş" olarak ayarlanır.

IBM MQ Client sürümü, bir XMS istemci uygulamasının IBM MQ ' e özgü özellikleri kullanıp kullanamayacağı konusunda da önemli bir rol oynar.

Not: XMSC_WMQ_OVERRIDEPROVIDERVERSION sistem özelliği, XMSC_WMQ_PROVIDER_VERSION özelliğini geçersiz kılar. Bağlantı üreticisi ayarını değiştiremiyorsanız bu özellik kullanılabilir.

XMSC_WMQ_PUB_ACK_INTERVAL**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

XMS istemcisi aracıdan bir alındı bildirimini istemeden önce bir yayınlayıcı tarafından yayınlanan iletilerin sayısı.

Bu özelliğin değerini azaltırsanız, istemci daha sık onay ister ve bu nedenle yayınlayıcının performansı düşer. Değeri yükseltirseniz, aracı başarısız olursa istemcinin kural dışı durum yayınması daha uzun sürer.

Özelliğin değeri pozitif bir tamsayıdır. Varsayılan değer 25 'tir.

XMSC_WMQ_QMGR_CCSID**Veri tipi:**

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

İleti Kuyruğu Arabirimi 'nde (MQI) tanımlanan karakter verileri alanlarının XMS istemcisi ile IBM MQ istemcisi arasında değiş tokuş edildiği, kodlanmış karakter takımının ya da kod sayfasının tanıtıcısı (CCSID). Bu özellik, iletilerin gövdelerindeki karakter verisi dizgileri için geçerli değildir.

XMS uygulaması istemci kipinde bir kuyruk yöneticisine bağlandığında, XMS istemcisi IBM MQ istemcisine bağlanır. İki istemci arasında değiş tokuş edilen bilgiler, MQI 'da tanımlı karakter verileri alanlarını içerir. Olağan koşullarda, IBM MQ istemcisi bu alanların istemcilerin çalıştığı sistemin kod sayfasında olduğunu varsayar. XMS istemcisi bu alanları farklı bir kod sayfasında sağlar ve almayı bekliyorsa, IBM MQ istemcisini bilgilendirmek için bu özelliği ayarlamanız gerekir.

IBM MQ istemcisi bu karakter verileri alanlarını kuyruk yöneticisine ilettiğinde, gerekiyorsa bu alanlardaki veriler kuyruk yöneticisi tarafından kullanılan kod sayfasına dönüştürülmelidir. Benzer şekilde, IBM MQ istemcisi kuyruk yöneticisinden bu alanları aldığı anda, gerekirse bu alanlardaki veriler XMS istemcisinin verileri almayı beklediği kod sayfasına dönüştürülmelidir. IBM MQ istemcisi bu veri dönüştürmelerini gerçekleştirmek için bu özelliği kullanır.

Varsayılan olarak özellik ayarlanmaz.

Bu özelliğin ayarlanması, yerli IBM MQ istemci uygulamalarını destekleyen bir IBM MQ istemcisi için MQCCSID ortam değişkeninin ayarlanmasıyla eşdeğerdir. Bu ortam değişkeniyle ilgili daha fazla bilgi için bkz. MQCCSID.

XMSC_WMQ_QUEUE_MANAGER

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: QMANAGER

JMS yönetim aracı kısa adı: QMGR

Bağlanılacak kuyruk yöneticisinin adı.

Varsayılan olarak özellik ayarlanmaz.

XMSC_WMQ_RECEIVE_CCSID

Kuyruk yöneticisi ileti dönüştürmesi için hedef CCSID ' yi belirleyen hedef özellik. XMSC_WMQ_RECEIVE_CONVERSION_QMGR değeri WMQ_RECEIVE_CONVERSION_QMGR olarak ayarlanmadıkça değer yoksayılr.

Veri tipi:

Tamsayı

Değer:

Herhangi bir pozitif tamsayı.

Varsayılan değer 1208 'dir.

Bir iletide GMO_CONVERT değerinin belirtilmesi isteğe bağlıdır. Bir GMO_CONVERT değeri belirtilirse, dönüştürme belirtilen değere göre gerçekleşir.

XMSC_WMQ_RECEIVE_CONVERSION

Veri dönüştürme işleminin kuyruk yöneticisi tarafından gerçekleştirilip gerçekleştirilmeyeceğini belirleyen hedef özellik.

Veri tipi:

Tamsayı

Değerler:

XMSC_WMQ_RECEIVE_CONVERSION_CLIENT_MSG (DEFAULT): Yalnızca XMS istemcisinde veri dönüştürme gerçekleştirir. Dönüştürme her zaman 1208 kod sayfası kullanılarak yapılır.

XMSC_WMQ_RECEIVE_CONVERSION_QMGR: XMS istemcisine ileti göndermeden önce kuyruk yöneticisinde veri dönüştürme işlemi gerçekleştirin.

XMSC_WMQ_RECEIVE_EXIT

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

çalıştırılacak kanal alma çıkışını tanımlar.

Özelliğin değeri, bir kanal alma çıkışını tanıtan ve aşağıdaki biçimde olan bir dizedir:

libraryName(entryPointAd)

Burada,

- **libraryName** yönetilen çıkışın tam yoludur .dll

- `entryPointAd` Alanı , ad alanı tarafından nitelenen sınıf adıdır.

Örneğin, `C:\MyReceiveExit.dll(MyReceiveExitNameSpace.MyReceiveExitClassName)`

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipinde bir kuyruk yöneticisine bağlandığında geçerlidir. Ayrıca, yalnızca yönetilen çıkışlar desteklenir.

XMSC_WMQ_RECEIVE_EXIT_INIT

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bir kanala geçirilen kullanıcı verileri, çağrıldığında çıkış alır.

Özelliğin değeri bir dizgi. Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipinde bir kuyruk yöneticisine bağlandığında ve [“XMSC_WMQ_RECEIVE_EXIT” sayfa 2045](#) özelliği ayarlandığında geçerlidir.

XMSC_WMQ_RESOLVED_QUEUE_MANAGER

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bu özellik, bağlı olduğu kuyruk yöneticisinin adını almak için kullanılır.

Bir CCDT (Client Channel Definition Table; İstemci Kanal Tanımlama Çizelgesi) ile kullanıldığında, bu ad Connection Factory 'de belirtilen kuyruk yöneticisi adından farklı olabilir.

XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bu özelliğe, bağlantıdan sonra kuyruk yöneticisinin tanıtıcısı yerleştirilir.

XMSC_WMQ_SECURITY_EXIT

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bir kanal güvenlik çıkışını tanımlar.

Özelliğin değeri, bir kanal güvenlik çıkışını tanıtan ve aşağıdaki biçimde olan bir dizedir:

libraryName(entryPointAd)

Burada,

- **libraryName** , yönetilen .dll çıkışının tam yoludur.
- `entryPointAd` Alanı , ad alanı tarafından nitelenen sınıf adıdır.

Örneğin, `C:\MySecurityExit.dll(MySecurityExitNameSpace.MySecurityExitClassName)`

Dizginin uzunluk üst sınırı 128 karakterdir.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipinde bir kuyruk yöneticisine bağlandığında geçerlidir. Ayrıca, yalnızca yönetilen çıkışlar desteklenir.

XMSC_WMQ_SECURITY_EXIT_INIT

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bir kanal güvenlik çıkışına çağrıldığında geçirilen kullanıcı verileri.

Kullanıcı verileri dizgisinin uzunluk üst sınırı 32 karakterdir.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipinde bir kuyruk yöneticisine bağlandığında ve [“XMSC_WMQ_SECURITY_EXIT” sayfa 2046](#) özelliği ayarlandığında geçerlidir.

XMSC_WMQ_SEND_EXIT

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bir kanal gönderme çıkışını tanımlar.

Özelliğin değeri bir dizgi. Kanal gönderme çıkışının biçimi şöyledir:

libraryName(entryPointAd)

Burada,

- **libraryName** , yönetilen .dll çıkışının tam yoludur.
- **entryPointAd** Alanı , ad alanı tarafından nitelenen sınıf adıdır.

Örneğin, C:\MySendExit.dll(MySendExitNameSpace.MySendExitClassName)

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipinde bir kuyruk yöneticisine bağlandığında geçerlidir. Ayrıca, yalnızca yönetilen çıkışlar desteklenir.

XMSC_WMQ_SEND_EXIT_INIT

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Çağrıldığında kanal gönderme çıkışlarına geçirilen kullanıcı verileri.

Özelliğin değeri, virgülle ayrılmış olarak kullanıcı verilerinin bir ya da daha çok öğeden oluşan bir dizedir. Varsayılan olarak özellik ayarlanmaz.

Kanal gönderme çıkışları sırasına geçirilen kullanıcı verilerini belirlemeye ilişkin kurallar, bir kanal alma çıkışları sırasına geçirilen kullanıcı verilerini belirlemeye ilişkin kurallarla aynıdır. Bu nedenle kurallar için bkz. [“XMSC_WMQ_RECEIVE_EXIT_INIT” sayfa 2046](#).

Bu özellik yalnızca, bir uygulama yönetilen istemci kipinde bir kuyruk yöneticisine bağlandığında ve [“XMSC_WMQ_SEND_EXIT” sayfa 2047](#) özelliği ayarlandığında geçerlidir.

XMSC_WMQ_SEND_CHECK_COUNT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Tek bir hareket etmeyen XMS oturumunda, zamanuyumsuz koyma hatalarının denetlenmesi arasında izin verilecek gönderme çağrılarının sayısı.

Varsayılan olarak bu özellik 0 olarak ayarlanır.

XMSC_WMQ_SHARE_CONV_ALLOWED

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Uygulanabilir nesnelere:

JMS yönetim aracı uzun adı: SHARECONVAL,

JMS yönetim aracı kısa adı: SCALD

Bir istemci bağlantısının, kanal tanımlamaları eşleşiyorsa, aynı işlemde aynı kuyruk yöneticisine yapılan diğer üst düzey XMS bağlantılarıyla yuvasını paylaşıp paylaşamayabileceğini belirler. Bu özellik, uygulama geliştirme, bakım ya da işletim nedenleriyle gerektiğinde Connections 'ın ayrı yuvalarda tam olarak yalıtılmasına olanak sağlamak için sağlanır. Bu özelliğin ayarlanması, temeldeki yuvanın paylaşılmasını XMS ' e gösterir. Bu, tek bir yuvayı paylaşan bağlantı sayısını göstermez. Yuva paylaşan bağlantı sayısı, IBM MQ istemcisi ile IBM MQ sunucusu arasında karşılaştırılan SHARECNV değeri tarafından belirlenir.

Bir uygulama, özelliği ayarlamak için aşağıdaki adlandırılmış değişmezleri ayarlayabilir:

- XMSC_WMQ_SHARE_CONV_ALLOWED_FALSE-Bağlantılar bir yuva paylaşmıyor.
- XMSC_WMQ_SHARE_CONV_ALLOWED_TRUE-Bağlantılar bir yuva paylaşıyor.

Varsayılan olarak özellik XMSC_WMQ_SHARE_CONV_ALLOWED_ENABLED olarak ayarlanır.

Bu özellik yalnızca, bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında geçerlidir.

XMSC_WMQ_SSL_CERT_STORES

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Kuyruk yöneticisine SSL bağlantısında kullanılacak sertifika iptal listelerini (CRL) bulduran sunucuların yerleri.

Özelliğin değeri, virgülle ayrılmış bir ya da daha çok URL ' nin listesidir. Her URL aşağıdaki biçimdedir:

```
[user[/password]@]ldap://[serveraddress][:portnum][, ...]
```

Bu biçim, temel MQJMS biçimiyle uyumludur, ancak bu biçimden genişletilmiştir.

Boş bir serveraddress olması geçerlidir. Bu durumda XMS , değerin "localhost" dizgisi olduğunu varsayar.

Örnek bir liste:

```
myuser/mypassword@ldap://server1.mycom.com:389  
ldap://server1.mycom.com
```


ldap://
ldap://:389

Yalnızca .NET için: IBM MQ 8.0 için, IBM MQ (WMQ_CM_CLIENT) ile yönetilen bağlantılar ve IBM MQ (WMQ_CM_CLIENT_UNMANAGED) ile yönetilmeyen bağlantılar TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

İlgili kavramlar

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

XMSC_WMQ_SSL_CIPHER_SPEC

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bir kuyruk yöneticisine güvenli bir bağlantıda kullanılacak CipherSpec ' in adı.

IBM MQ TLS desteği ile kullanabileceğiniz şifre belirteçleri aşağıdaki tabloda listelenmiştir. Kişisel sertifika isteğinde bulunduğunuzda, genel ve özel anahtar çifti için bir anahtar boyutu belirtirsiniz. SSL el sıkışması sırasında kullanılan anahtar boyutu, çizelgede belirtildiği şekilde CipherSpectarafından belirlenmedikçe sertifikada saklanan boyuttur. Varsayılan olarak bu özellik ayarlanmaz.

CipherSpec adı	Kullanılan protokol	Hash algoritması	Şifreleme Algoritması	Şifreleme bitleri	FIPS ¹	Suite B 128 bit	Takım B 192 bit
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	AES.	128	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_256_CBC_SHA ²	TLS 1.0	SHA-1	AES.	256	Evet	Hayır	Hayır
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	Hayır	Hayır	Hayır
TLS_RSA_WITH_3DES_EDE_CBC_SHA ⁴	TLS 1.0	SHA-1	3DES	168	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES.	128	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES.	256	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES.	128	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	SHA-256	AES.	256	Evet	Hayır	Hayır
ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Hayır	Hayır	Hayır
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Evet	Hayır	Hayır
ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Hayır	Hayır	Hayır
ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Evet	Hayır	Hayır
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES.	128	Evet	Hayır	Hayır

CipherSpec adı	Kullanılan protokol	Hash algoritması	Şifreleme Algoritması	Şifreleme bitleri	FIPS ¹	Suite B 128 bit	Takım B 192 bit
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES.	256	Evet	Hayır	Hayır
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES.	128	Evet	Hayır	Hayır
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES.	256	Evet	Hayır	Hayır
ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES.	128	Evet	Evet	Hayır
ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES.	256	Evet	Hayır	Evet
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES.	128	Evet	Hayır	Hayır
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES.	256	Evet	Hayır	Hayır
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	Yok	0	Hayır	Hayır	Hayır
ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-256	Yok	0	Hayır	Hayır	Hayır
ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-256	Yok	0	Hayır	Hayır	Hayır
TLS_RSA_WITH_NULL_NULL	TLS 1.2	Yok	Yok	0	Hayır	Hayır	Hayır
TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Hayır	Hayır	Hayır

Notlar:

1. CipherSpec ' in Federal Bilgi İşleme Standartları (FIPS) 140-2 ile uyumlu olup olmadığını belirtir. FIPS 'nin açıklaması ve IBM MQ ' in FIPS 140-2 uyumlu işlem için nasıl yapılandırılacağı hakkında bilgi için bkz. [Federal Information Processing Standards \(FIPS\)](#).
2. Bu CipherSpec , IBM MQ Explorer tarafından kullanılan JRE 'ye uygun kısıtlamasız ilke dosyaları uygulanmadıkça, IBM MQ Explorer ' den bir kuyruk yöneticisine bağlantı korumak için kullanılamaz.
3. Bu CipherSpec , 19 Mayıs 2007 'den önce FIPS 140-2 sertifikalıydı.
4. IBM MQ , FIPS 140-2 uyumlu işlem için yapılandırıldığında, bu CipherSpec , bağlantı AMQ9288 hatasıyla sonlandırılmadan önce 32 GB ' ye kadar veri aktarmak için kullanılabilir. Bu hatayı önlemek için, üçlü DES kullanmaktan kaçının (kullanımdan kaldırılan) ya da bir FIPS 140-2 yapılandırmasında bu CipherSpec ' i kullanırken gizli anahtar sıfırlamayı etkinleştirin.

İlgili kavramlar

[İletilerin veri bütünlüğü](#)

İlgili görevler

[güvenlik](#)

[CipherSpecs ' nin Belirtilmesi](#)

XMSC_WMQ_SSL_CIPHER_SUITE

Veri tipi:

Dize

Bu özelliğin özelliği:

ConnectionFactory

Bir kuyruk yöneticisine TLS bağlantısında kullanılacak CipherSuite ' in adı. Güvenli bağlantının kararlaştırılmasında kullanılan protokol, belirtilen CipherSuite' e bağlıdır.

Bu özellik aşağıdaki kurallı değerleri içerir:

- 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

Bu değer, XMSC_WMQ_SSL_CIPHER_SPEC için alternatif olarak sağlanabilir.

XMSC_WMQ_SSL_CIPHER_SPEC için boş olmayan bir değer belirtilirse, bu değer XMSC_WMQ_SSL_CIPHER_SUITE ayarını geçersiz kılar. XMSC_WMQ_SSL_CIPHER_SPEC için değer yoksa, IBM Global Security Kit (GSKit)' e verilecek şifreleme takımı olarak XMSC_WMQ_SSL_CIPHER_SUITE değeri kullanılır. Bu durumda değer, Bir IBM MQ kuyruk yöneticisine XMS bağlantıları için CipherSuite ve CipherSpec ad eşlemeleri içinde açıklandığı gibi eşdeğer CipherSpec değeriyle eşlenir.

Hem XMSC_WMQ_SSL_CIPHER_SPEC hem de XMSC_WMQ_SSL_CIPHER_SUITE boşsa, pChDef->SSLCipherSpec alanı boşluklarla doldurulur.

Yalnızca .NET için: IBM MQ 8.0 için, IBM MQ (WMQ_CM_CLIENT) ile yönetilen bağlantılar ve IBM MQ (WMQ_CM_CLIENT_UNMANAGED) ile yönetilmeyen bağlantılar TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

İlgili kavramlar

Yönetilmeyen .NET istemcisi için SSL ve TLS desteği

Yönetilen .NET istemcisi için SSL ve TLS desteği

XMSC_WMQ_SSL_CRYPTO_HW

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

İstemci sistemine bağlı şifreleme donanımına ilişkin yapılandırma ayrıntıları.

Bu özellik aşağıdaki kurallı değerleri içerir:

- GSK_ACCELERATOR_RAINBOW_CS_OFF
- GSK_ACCELERATOR_RAINBOW_CS_ON
- GSK_ACCELERATOR_NCIPHER_NF_OFF
- GSK_ACCELERATOR_NCIPHER_NF_ON

PKCS11 şifreleme donanımı için özel bir biçim vardır (burada DriverPath, TokenLabel ve TokenPassword kullanıcı tanımlı dizgilerdir):

```
GSK_PKCS11=PKCS#11 DriverPath; PKCS#11 TokenLabel;PKCS#11 TokenPassword
```

XMS , dizginin içeriğini yorumlamaz ya da değiştirmez. 256 tek baytlık karakter sınırına kadar sağlanan değeri MQSCO.CryptoHardware (Şifreleme Donanımı) alanı.

Yalnızca .NET için: IBM MQ 8.0 için, IBM MQ (WMQ_CM_CLIENT) ile yönetilen bağlantılar ve IBM MQ (WMQ_CM_CLIENT_UNMANAGED) ile yönetilmeyen bağlantılar TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

İlgili kavramlar

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

XMSC_WMQ_SSL_FIPS_REQUIRED

Veri tipi:

Boole

Bu özelliğin özelliği:

ConnectionFactory

Bu özelliğin değeri, bir uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanıp kullanamayacağını belirler. Bu özellik true olarak ayarlanırsa, istemci-sunucu bağlantısı için yalnızca FIPS algoritmaları kullanılır.

Bu özellik, MQSCO.FipsRequired:

Çizelge 880. MQSCO.FlipsRequired özelliği		
Değer	Açıklama	MQSCO.FipsRequired
yanlış	Herhangi bir CipherSpec kullanılabilir.	MQSSL_FIPS_NO (varsayılan)
doğru	Bu istemci bağlantısına uygulanan CipherSpec içinde yalnızca FIPS onaylı şifreleme algoritmaları kullanılabilir.	MQSSL_FIPS_YES

XMS , ilgili değeri MQSCO.FipsRequired .

Yalnızca .NET için: IBM MQ 8.0 için, IBM MQ (WMQ_CM_CLIENT) ile yönetilen bağlantılar ve IBM MQ (WMQ_CM_CLIENT_UNMANAGED) ile yönetilmeyen bağlantılar TLS/SSL bağlantılarını destekler.

İlgili kavramlar

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

XMSC_WMQ_SSL_KEY_REPOSITORY

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Anahtarların ve sertifikaların saklandığı anahtar veritabanı dosyasının konumu.

XMS , 256 tek baytlık karakter sınırını aşacak şekilde dizgiyi MQSCO.KeyRepository alanı. IBM MQ , bu dizgiyi tam yol da içinde olmak üzere bir dosya adı olarak yorumlar.

Yalnızca .NET için: IBM MQ 8.0 için, IBM MQ (WMQ_CM_CLIENT) ile yönetilen bağlantılar ve IBM MQ (WMQ_CM_CLIENT_UNMANAGED) ile yönetilmeyen bağlantılar TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

İlgili kavramlar

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

XMSC_WMQ_SSL_KEY_RESETCOUNT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

KeyResetSayısı, gizli anahtar yeniden anlaşılmadan önce SSL etkileşimi içinde gönderilen ve alınan toplam şifrelenmemiş bayt sayısını temsil eder. Bayt sayısı, MCA tarafından gönderilen denetim bilgilerini içerir.

XMS , bu özellik için sağladığınız değeri MQSCO.KeyResetCount .

MQSCO.KeyRestCount yalnızca IBM MQ sürüm 6 'da kullanılabilir. IBM MQ sürüm 5.3ise, bu özellik ayarlanırsa, XMS kuyruk yöneticisiyle bağlantı kurmayı denemez ve bunun yerine uygun bir kural dışı durum verir.

Yalnızca .NET için: IBM MQ 8.0için, IBM MQ (WMQ_CM_CLIENT) ile yönetilen bağlantılar ve IBM MQ (WMQ_CM_CLIENT_UNMANAGED) ile yönetilmeyen bağlantılar TLS/SSL bağlantılarını destekler.

Bu özelliğin varsayılan değeri sıfırdır; bu, gizli anahtarların hiçbir zaman yeniden anlaşılmadığı anlamına gelir.

İlgili kavramlar

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

XMSC_WMQ_SSL_PEER_NAME

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Kuyruk yöneticisine SSL bağlantısında kullanılacak eşdüzey ad.

Bu özellik için kurallı değerlerin listesi yok. Bunun yerine, bu dizgiyi SSLPEERkurallarına göre oluşturmanız gerekir.

Eşdüzey adı örneği:

```
"CN=John Smith, O=IBM ,OU=Test , C=GB"
```

XMS , dizgiyi doğru tek byte kod sayfasına kopyalar ve MQCONNX çağrılmadan önce doğru değerleri MQCD.SSLPeerNamePtr ve MQCD.SSLPeerNameLength içine yerleştirir.

Bu özellik, uygulamanın istemci kipinde bir kuyruk yöneticisine bağlanması durumunda geçerlidir.

Yalnızca .NET için: IBM MQ 8.0için, IBM MQ (WMQ_CM_CLIENT) ile yönetilen bağlantılar ve IBM MQ (WMQ_CM_CLIENT_UNMANAGED) ile yönetilmeyen bağlantılar TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

İlgili kavramlar

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

İlgili başvurular

[SSLPEERNAME](#)

XMSC_WMQ_SYNCPOINT_ALL_TR

Veri tipi:

System.Boolean

Bu özelliğin özelliği:

ConnectionFactory

Eşitleme noktası denetimi içindeki kuyruklardan tüm iletilerin alınıp alınmayacağı.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

yanlış

doğru

Anlamı

Koşullar uygun olduğunda, XMS istemcisi eşitleme noktası denetimi dışındaki kuyruklardan ileti alabilir.

XMS istemcisi, eşitleme noktası denetimi içindeki kuyruklardan tüm iletileri almalıdır.

Varsayılan değer: false.

XMSC_WMQ_TARGET_CLIENT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Hedef

URI ' de kullanılan ad:

targetClient

Hedefe gönderilen iletilerin bir MQRFH2 üstbilgisi içerip içermediğini belirler.

Bir uygulama MQRFH2 üstbilgisi içeren bir ileti gönderirse, alan uygulamanın üstbilgiyi işleyebilmesi gerekir.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_WMQ_TARGET_DEST_JMS

XMSC_WMQ_TARGET_DEST_MQ

Anlamı

Hedefe gönderilen iletiler bir MQRFH2 üstbilgisi içeriyor. Uygulama iletileri başka bir XMS uygulamasına, IBM MQ classes for JMS uygulamasına ya da MQRFH2 üstbilgisini işlemek üzere tasarlanmış bir yerel IBM MQ uygulamasına gönderiyorsa bu değeri belirtin.

Hedefe gönderilen iletiler MQRFH2 üstbilgisi içermiyor. Uygulama, iletileri MQRFH2 üstbilgisini işlemek üzere tasarlanmamış bir yerel IBM MQ uygulamasına gönderiyorsa bu değeri belirtin.

Varsayılan değer XMSC_WMQ_TARGET_DEST_JMS 'dir.

XMSC_WMQ_TEMP_Q_PREFIX

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulama bir XMS geçici kuyruğu oluşturduğunda oluşturulan IBM MQ dinamik kuyruğunun adını oluşturmak için kullanılan önek.

Öneki oluşturmaya ilişkin kurallar, nesne tanımlayıcıda **DynamicQName** alanının içeriğini oluşturmaya ilişkin kurullarla aynıdır, ancak son boşluk olmayan karakter bir yıldız işareti (*) olmalıdır. Özellik

ayarlanmazsa, kullanılan deęer dięer platformlarda CSQ . * on z/OS ve AMQ . * olur. Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca noktadan noktaya etki alanında geçerlidir.

XMSC_WMQ_TEMP_TOPIC_PREFIX

Veri tipi:

Dizgi

Bu özelliğın özelliğı:

ConnectionFactory, Hedef

Geçici konular oluştururken XMS , "TEMP/TEMPTOPICPREFIX/unique_id" biçiminde bir konu dizgisi oluşturur ya da bu özellik varsayılan deęeri içeriyorsa, bu dizgi ("TEMP/unique_id") oluşturulur. Boş olmayan bir deęer belirtilmesi, abonelerin bu bağlantı altında yaratılan geçici konulara ilişkin yönetilen kuyruklarının yaratılması için belirli model kuyruklarının tanımlanmasına olanak sağlar.

Yalnızca bir IBM MQ konu dizesi için geçerli karakterlerden oluşan boş olmayan dizgiler, bu özellik için geçerli bir deęerdir.

Varsayılan olarak bu özellik "" (boş dizgi) olarak ayarlanır.

Not: Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

XMSC_WMQ_TEMPORARY_MODEL

Veri tipi:

Dizgi

Bu özelliğın özelliğı:

ConnectionFactory

Uygulama bir XMS geçici kuyruğı yarattığında dinamik kuyruğın yaratıldığı IBM MQ model kuyruğının adı.

Özelliğın varsayılan deęeri SYSTEM.DEFAULT.MODEL.QUEUE.

Bu özellik yalnızca noktadan noktaya etki alanında geçerlidir.

XMSC_WMQ_WILDCARD_FORMAT

Veri tipi:

System.Int32

Bu özelliğın özelliğı:

ConnectionFactory, Hedef

Bu özellik, genel arama karakteri sözdiziminin hangi sürümünün kullanılacağını belirler.

IBM MQ '*' ve '?' ile yayınlama/abone olma özelliğini kullanırken genel arama karakteri olarak deęerlendirilir. IBM Integration Bus ile yayınlama aboneliğı kullanılırken '#' ve '+' joker karakter olarak deęerlendirilir. Bu özellik, XMSC_WMQ_BROKER_VERSION özelliğının yerini alır.

Bu özellik için geçerli deęerler şunlardır:

XMSC_WMQ_WILDCARD_TOPIC_ONLY

Yalnızca konu düzeyi genel arama karakterlerini tanır; örn. '#' ve '+' genel arama karakteri olarak kabul edilir. Bu deęer XMSC_WMQ_BROKER_V2 ile aynıdır.

XMSC_WMQ_WILDCARD_CHAR_ONLY

Yalnızca genel arama karakterlerini tanır, örn. '*' ve '?' genel arama karakteri olarak deęerlendirilir. Bu deęer XMSC_WMQ_BROKER_V1 ile aynıdır.

Varsayılan olarak bu özellik XMSC_WMQ_WILDCARD_TOPIC_ONLY olarak ayarlanır.

XMSC_WPM_BUS_NAME

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory ve Hedef

URI ' de kullanılan ad:

busName

Bir bağlantı üreticisi için, uygulamanın bağlandığı hizmet tümleştirme veriyolunun adı ya da hedef için, hedefin bulunduğu hizmet tümleştirme veriyolunun adı.

Konu olan bir hedef için bu özellik, ilişkili konu alanının bulunduğu hizmet tümleştirme veriyolunun adıdır. Bu konu alanı XMSC_WPM_TOPIC_SPACE özelliğiyle belirtilir.

Özellik bir hedef için ayarlanmamışsa, uygulamanın bağlandığı hizmet tümleştirme veriyolunda kuyruk ya da ilişkili konu alanının var olduğu varsayılır.

Varsayılan olarak özellik ayarlanmaz.

XMSC_WPM_CONNECTION_PROTOCOL

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Bağlantı

İleti alışverişi altyapısına bağlantı için kullanılan iletişim protokolü. Bu özellik salt okunur.

Özelliğin olası değerleri şunlardır:

Değer	Anlamı
XMSC_WPM_CP_HTTP	Bağlantı TCP/IP üzerinden HTTP kullanır.
XMSC_WPM_CP_TCP	Bağlantı TCP/IP ' yi kullanır.

XMSC_WPM_CONNECTION_YAKINLIĞI

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Bağlantı için bağlantı yakınlığı ayarı. Bu özellik, uygulamanın bağlandığı ileti alışverişi altyapısının önyükleme sunucusuna ne kadar yakın olması gerektiğini belirler.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer	Bağlantı yakınlığı ayarı
XMSC_WPM_CONNECTION_PROXIMITY_BUS	Veriyolu
XMSC_WPM_CONNECTION_PROXIMITY_CLUSTER	Küme
XMSC_WPM_CONNECTION_PROXIMITY_HOST	Anasistem
XMSC_WPM_CONNECTION_PROXIMITY_SERVER	Sunucu

Varsayılan değer: XMSC_WPM_CONNECTION_PROXIMITY_BUS.

XMSC_WPM_DUR_SUB_HOME

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

URI ' de kullanılan ad:

durableSubscriptionAna Sayfası

Bir bağlantıya ya da hedefe ilişkin tüm sürekli aboneliklerin yönetildiği ileti alışverişi altyapısının adı. Sürekli abonelere teslim edilecek iletiler, aynı ileti alışverişi altyapısının yayın noktasında saklanır.

Bir uygulamanın bağlantıyı kullanan sürekli bir abone yaratabilmesi için, bağlantı için kalıcı bir abonelik ana dizini belirtilmelidir. Hedef için belirtilen herhangi bir değer, bağlantı için belirtilen değeri geçersiz kılar.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

XMSC_WPM_ANASISTEM_ADI**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

Bağlantı

Uygulamanın bağlı olduğu ileti alışverişi altyapısını içeren sistemin anasistem adı ya da IP adresi. Bu özellik salt okunur.

XMSC_WPM_LOCAL_ADDRESS**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Bir hizmet bütünleştirme veriyolu bağlantısı için bu özellik, kullanılacak yerel ağ arabirimini ya da kullanılacak yerel kapı ya da yerel kapı aralığını ya da her ikisini de belirtir.

Özelliğin değeri, aşağıdaki biçime sahip bir dizedir:

[*anasistem_adi*] [(*alt_kapı*) [,*yüksek_kapı*]]

Değişkenlerin anlamları şunlardır:

anasistem_adi

Bağlantı için kullanılacak yerel ağ arabiriminin anasistem adı ya da IP adresi.

Bu bilgilerin sağlanması, uygulamanın çalıştığı sistemde iki ya da daha çok ağ arabirimi varsa ve bağlantı için hangi arabirimin kullanılması gerektiğini belirleyebilmeniz gerekiyorsa gereklidir. Sistemde yalnızca bir ağ arabirimi varsa, yalnızca bu arabirim kullanılabilir. Sistemde iki ya da daha çok ağ arabirimi varsa ve hangi arabirimin kullanılması gerektiğini belirlemezseniz, arabirim rasgele seçilir.

alt_kapı

Bağlantı için kullanılacak yerel kapının numarası.

high_port da belirtilirse, *low_port* bir kapı numarası aralığındaki en düşük kapı numarası olarak yorumlanır.

yüksek_kapı

Bir kapı numarası aralığındaki en yüksek kapı numarası. Bağlantı için belirtilen aralıktaki kapılardan biri kullanılmalıdır.

Aşağıda, özelliğin geçerli değerlerine ilişkin bazı örnekler verilmiştir:

JÜPITER

9.20.4.98

JÜPITER (1000)

9.20.4.98(1000,2000)
(1000)
(1000,2000)

Varsayılan olarak özellik ayarlanmaz.

XMSC_WPM_ME_NAME

Veri tipi:

Dizgi

Bu özelliğin özelliği:

Bağlantı

Uygulamanın bağlı olduğu ileti alışverişi altyapısının adı. Bu özellik salt okunur.

XMSC_WPM_NON_PERSISTENT_MAP

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Bağlantı kullanılarak gönderilen kalıcı olmayan iletilerin güvenilirlik düzeyi.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_WPM_MAPPING_AS_DESTINATION

Güvenilirlik düzeyi

Hizmet tümleştirme veriyolundaki kuyruk ya da konu alanı için belirtilen varsayılan güvenilirlik düzeyi tarafından belirlenir

XMSC_WPM_MAPPING_BEST_EFFORT_NON_Kalıcı

En iyi çalışma kalıcı değil

XMSC_WPM_MAPPING_EXPRESS_NON_Kalıcı

Express kalıcı olmayan

XMSC_WPM_MAPPING_RELIABLE_NON_Kalıcı

Güvenilir kalıcı olmayan

XMSC_WPM_MAPPING_RELIABLE_PERSISTENT

Güvenilir kalıcı

XMSC_WPM_MAPPING_ASSURED_PERSISTENT

Güvenli kalıcı

Varsayılan değer XMSC_WPM_MAPPING_EXPRESS_NON_PERSISTENT değeridir.

XMSC_WPM_PERSISTENT_MAP

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

Bağlantı kullanılarak gönderilen kalıcı iletilerin güvenilirlik düzeyi.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_WPM_MAPPING_AS_DESTINATION

XMSC_WPM_MAPPING_BEST_EFFORT_NON_
Kalıcı

XMSC_WPM_MAPPING_EXPRESS_NON_
Kalıcı

XMSC_WPM_MAPPING_RELIABLE_NON_
Kalıcı

XMSC_WPM_MAPPING_RELIABLE_PERSISTENT

XMSC_WPM_MAPPING_ASSURED_PERSISTENT

Varsayılan değer XMSC_WPM_MAPPING_RELIABLE_PERSISTENT değeridir.

Güvenilirlik düzeyi

Hizmet tümleştirme veriyolundaki kuyruk ya da konu alanı için belirtilen varsayılan güvenilirlik düzeyi tarafından belirlenir

En iyi çalışma kalıcı değil

Express kalıcı olmayan

Güvenilir kalıcı olmayan

Güvenilir kalıcı

Güvenli kalıcı

XMSC_WPM_PORT

Veri tipi:

System.Int32

Bu özelliğin özelliği:

Bağlantı

Uygulamanın bağlı olduğu ileti alışverişi altyapısı tarafından dinlenen kapının sayısı. Bu özellik salt okunur.

XMSC_WPM_PROVIDER_ENDPOINTS

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Önyükeme sunucularının bir ya da daha çok uç noktası adresi sırası. Uç nokta adresleri virgülle ayrılır.

Önyükeme sunucusu, uygulamanın bağlandığı ileti alışverişi altyapısını seçmekten sorumlu bir uygulama sunucusudur. Önyükeme sunucusunun uç noktası adresi aşağıdaki biçimde olur:

anasistem_adı:kapı_no:zincir_adı

Bir uç nokta adresinin bileşenlerinin anlamları aşağıdaki gibidir:

anasistem_adı

Önyükeme sunucusunun bulunduğu sistemin anasistem adı ya da IP adresi. Anasistem adı ya da IP adresi belirtilmezse, varsayılan değer localhost olur.

kapı_numarası

Önyükeme sunucusunun gelen istekleri dinlediği kapının numarası. Kapı numarası belirtilmezse, varsayılan değer 7276 olur.

zincir_adı

Önyükeme sunucusu tarafından kullanılan önyükeme iletim zincirinin adı. Geçerli değerler şunlardır:

Geçerli değer

XMSC_WPM_BOOTSTRAP_HTTP

XMSC_WPM_BOOTSTRAP_HTTPS

Önyükeme iletim zincirinin adı

BootstrapTunneledİleti Sistemi

BootstrapTunneledSecureMessaging

Geçerli değer

XMSC_WPM_BOOTSTRAP_SSL

XMSC_WPM_BOOTSTRAP_TCP

Önyükleme iletim zincirinin adı

BootstrapSecureİleti Sistemi

BootstrapBasicİleti Sistemi

Ad belirtilmezse, varsayılan değer XMSC_WPM_BOOTSTRAP_TCP olur.

Uç nokta adresi belirtilmezse, varsayılan değer localhost:7276:BootstrapBasicMessagingdeğeridir.

XMSC_WPM_SSL_CIPHER_SUITE**Veri tipi:**

Dize

Bu özelliğin özelliği:

ConnectionFactory

Bir WebSphere Application Server service integration bus ileti alışverişi altyapısına TLS bağlantısında kullanılacak CipherSuite ' in adı. Güvenli bağlantının kararlaştırılmasında kullanılan protokol, belirtilen CipherSuite' e bağlıdır.

Çizelge 881. WebSphere Application Server service integration bus ileti alışverişi altyapısına bağlantı için CipherSuite seçenekleri

şifre paketi	Kullanılan protokol
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

Notlar:

- Windows** TLS_RSA_WITH_AES_128_CBC_SHA ve TLS_RSA_WITH_AES_256_CBC_SHA CipherSuites yalnızca Windows üzerinde desteklenir. (Bu, GSKitarafından dikte edilir.)
- Deprecated** TLS_RSA_WITH_3DES_EDE_CBC_SHA kullanımdan kaldırıldı. Ancak, bağlantı AMQ9288hatasıyla sonlandırılmadan önce 32 GB ' ye kadar veri aktarmak için yine de kullanılabilir. Bu hatayı önlemek için üçlü DES kullanmaktan kaçınmanız ya da bu CipherSpec' i kullanırken gizli anahtar sıfırlamasını etkinleştirmeniz gerekir.

Bu özellik için varsayılan değer yoktur. SSL ya da TLS kullanmak istiyorsanız, bu özellik için bir değer belirtmeniz gerekir; tersi durumda, uygulamanız sunucuya başarıyla bağlanamaz.

XMSC_WPM_SSL_FIPS_REQUIRED

Not: AIX, Linux, and Windows işletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerinde aranarak görüntülenebilir.

Veri tipi:

Boole

Bu özelliğin özelliği:

ConnectionFactory

Bu özelliğin değeri, bir uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanıp kullanamayacağını belirler. Bu özellik true olarak ayarlanırsa, istemci-sunucu bağlantısı için yalnızca FIPS

algoritmaları kullanılır. Bu özelliğin değerinin TRUE olarak ayarlanması, uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanmasını önler.

Varsayılan olarak, özellik FALSE (FIPS kipi kapalı) olarak ayarlanır.

XMSC_WPM_SSL_KEY_REPOSITORY

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Güvenli bağlantıda kullanılacak genel ya da özel anahtarları içeren anahtarlık dosyası olan dosyanın yolu.

Anahtarlık dosyası özelliğinin XMSC_WPM_SSL_MS_CERTIFICATE_STORE özel değerine ayarlanması, Microsoft Windows anahtar veritabanının kullanılacağını belirtir. **Denetim Masası > Internet Seçenekleri > İçerik > Sertifikalar** altında bulunan Microsoft Windows anahtar veritabanının kullanılması, ayrı bir anahtar dosyası veritabanı gereksinmesini ortadan kaldırır. Bu değışmezin Windows x64 üzerinde ve diğer platformlarda kullanılmasına izin verilmez.

Varsayılan olarak özellik ayarlanmaz.

XMSC_WPM_SSL_KEYRING_LABEL

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Sunucuyla kimlik doğrulaması yaparken kullanılacak sertifika. Değer belirtilmezse, varsayılan sertifika kullanılır.

Varsayılan olarak özellik ayarlanmaz.

XMSC_WPM_SSL_KEYRING_PW

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Anahtarlık dosyasının parolası.

Bu özellik, anahtarlık dosyasının parolasını yapılandırmak için XMSC_WPM_SSL_KEYRING_STASH_FILE dosyasının kullanılmasına alternatif olarak kullanılabilir.

Varsayılan olarak özellik ayarlanmaz.

XMSC_WPM_SSL_KEYRING_STASH_FILE

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Anahtar havuzu dosyasının parolasını içeren ikili dosyanın adı.

Bu özellik, anahtarlık dosyasına ilişkin parolayı yapılandırmak için XMSC_WPM_SSL_KEYRING_PW değerinin kullanılmasına bir alternatif olarak kullanılabilir.

Varsayılan olarak özellik ayarlanmaz.

XMSC_WPM_TARGET_GROUP

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

İleti alışverişi altyapılarının hedef grubunun adı. Hedef grubun türü XMSC_WPM_TARGET_TYPE özelliği tarafından belirlenir.

Bir ileti alışverişi altyapısına ilişkin aramayı, hizmet tümleştirme veriyolundaki ileti alışverişi altyapılarının bir alt grubuyla sınırlamak istiyorsanız bu özelliği ayarlayın. Uygulamanızın hizmet tümleştirme veriyolundaki herhangi bir ileti alışverişi altyapısına bağlanabilmesini istiyorsanız, bu özelliği ayarlamayın.

Varsayılan olarak özellik ayarlanmaz.

XMSC_WPM_TARGET_ANLAM

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

İleti alışverişi altyapılarının hedef grubunun önemi.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_WPM_TARGET_ANLAMLI_
Tercih Edilen

XMSC_WPM_TARGET_ANLAMLI_
ZORUNLU

Anlamı

Varsa, hedef gruptaki bir ileti alışverişi altyapısı seçilir. Ters durumda, aynı hizmet tümleştirme veriyolunda olması koşuluyla, hedef grubun dışındaki bir ileti alışverişi altyapısı seçilir.

Seçilen ileti alışverişi altyapısı hedef grupta olmalıdır. Hedef gruptaki bir ileti alışverişi altyapısı kullanılamıyorsa, bağlantı işlemi başarısız olur.

Özelliğin varsayılan değeri XMSC_WPM_TARGET_ANLAMANCE_PREFERRED değeridir.

XMSC_WPM_TARGET_TRANSPORT_CHAIN

Veri tipi:

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulamanın bir ileti alışverişi altyapısına bağlanmak için kullanması gereken gelen iletim zincirinin adı.

Özelliğin değeri, ileti alışverişi altyapısını barındıran uygulama sunucusunda var olan herhangi bir gelen iletim zincirinin adı olabilir. Önceden tanımlanmış gelen iletim zincirlerinden biri için aşağıdaki adlandırılmış değişmez sağlanmıştır:

Adlandırılmış değişmez

XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC

İletim zincirinin adı

InboundBasicİleti Sistemi

Özelliğin varsayılan değeri şudur: XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC.

XMSC_WPM_TARGET_TYPE

Veri tipi:

System.Int32

Bu özelliğin özelliği:

ConnectionFactory

İleti alışverişi altyapılarının hedef grubunun tipi. Bu özellik, XMSC_WPM_TARGET_GROUP özelliğinin tanıttığı hedef grubun niteliğini belirler.

Özelliğin geçerli değerleri şunlardır:

Geçerli değer

XMSC_WPM_TARGET_TYPE_BUSMEMBER

XMSC_WPM_TARGET_TYPE_CUSTOM

XMSC_WPM_TARGET_TYPE_ME

Anlamı

Hedef grubun adı, bir veriyolu üyesinin adıdır. Hedef grup, veriyolu üyesindeki tüm ileti alışverişi altyapıları.

Hedef grubun adı, ileti alışverişi altyapılarının kullanıcı tanımlı grubunun adıdır. Hedef grup, kullanıcı tanımlı gruba kayıtlı tüm ileti alışverişi altyapılarıdır.

Hedef grubun adı, ileti alışverişi altyapısının adıdır. Hedef grup, belirtilen ileti alışverişi altyapısıdır.

Varsayılan olarak özellik ayarlanmaz.

XMSC_WPM_TEMP_Q_PREFIX**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulama bir XMS geçici kuyruğu yarattığında hizmet tümleştirme veriyolunda yaratılan geçici kuyruğun adını oluşturmak için kullanılan önek. Önek en çok 12 karakter içerebilir.

Geçici bir kuyruğun adı, önekin izlediği "_Q" karakterleriyle başlar. Adın geri kalanı, sistem tarafından oluşturulan karakterlerden oluşur.

Varsayılan olarak, özellik ayarlanmaz; bu, geçici bir kuyruk adının öneki olmadığı anlamına gelir.

Bu özellik yalnızca noktadan noktaya etki alanında geçerlidir.

XMSC_WPM_TEMP_TOPIC_PREFIX**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

ConnectionFactory

Uygulama tarafından oluşturulan geçici bir konunun adını oluşturmak için kullanılan önek. Önek en çok 12 karakter içerebilir.

Geçici bir konunun adı, önekten sonra gelen "_T" karakterleriyle başlar. Adın geri kalanı, sistem tarafından oluşturulan karakterlerden oluşur.

Varsayılan olarak özellik ayarlanmaz; bu, geçici bir konunun adının önek olmadığı anlamına gelir.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

XMSC_WPM_TOPIC_SPACE**Veri tipi:**

Dizgi

Bu özelliğin özelliği:

Hedef

URI ' de kullanılan ad:

topicSpace

Konuyu içeren konu alanının adı. Yalnızca konu olan bir hedef bu özelliğe sahip olabilir.

Varsayılan olarak özellik ayarlanmaz; bu, varsayılan konu alanının varsayıldığı anlamına gelir.

Bu özellik yalnızca yayınlama/abone olma etki alanında geçerlidir.

Managed File Transfer uygulama başvurusu geliştirilmesi

Managed File Transfer için uygulama geliştirmenize yardımcı olacak başvuru bilgileri.

Programları başlatmak için fteCreateTransfer yönteminin kullanılmasına ilişkin örnekler

Bir aktarmadan önce ya da sonra çalıştırılacak programları belirlemek için **fteCreateTransfer** komutunu kullanabilirsiniz.

fteCreateTransfer komutunu kullanmanın yanı sıra, bir programı aktarımdan önce ya da sonra başlatmanın başka yolları da vardır. Daha fazla bilgi için [MFT ile çalıştırılacak programları belirtme](#) başlıklı konuya bakın.

Bu örneklerin tümü, bir program belirtmek için aşağıdaki sözdizimini kullanır:

```
[type:]commandspec[, [retrycount][, [retrywait][, successsrc]]]
```

Bu sözdizimi hakkında daha fazla bilgi için bkz. **fteCreateTransfer**: yeni bir dosya aktarımı başlatma.

Yürütülebilir bir programın çalıştırılması

Aşağıdaki örnek, mycommand adlı yürütülebilir bir programı belirtir ve programa iki bağımsız değişken (a ve b) geçirir.

```
mycommand(a,b)
```

Bu programı aktarım başlamadan önce AGENT1 kaynak aracısında çalıştırmak için aşağıdaki komutu kullanın:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -presrc mycommand(a,b)  
destinationSpecification sourceSpecification
```

Yürütülebilir bir programın çalıştırılması ve yeniden denemesi

Aşağıdaki örnek, herhangi bir bağımsız değişken almayan simple adlı yürütülebilir bir programı belirtir. retrycount için 1 değeri ve retrywait için 5 değeri belirtilir. Bu değerler, beş saniye bekledikten sonra başarılı bir dönüş kodu döndürmezse programın bir kez yeniden deneneceği anlamına gelir. successsrc için değeri belirtilmedi, bu nedenle başarılı olan tek dönüş kodu 0 varsayılan değeridir.

```
executable:simple,1,5
```

Aktarım tamamlandıktan sonra bu programı AGENT1 kaynak aracısında çalıştırmak için aşağıdaki komutu kullanın:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc executable:simple,1,5  
destinationSpecification sourceSpecification
```


Ant komut dosyasının çalıştırılması ve başarılı dönüş kodlarının belirtilmesi


Aşağıdaki örnek, `myscript` adlı bir Ant komut dosyasını belirtir ve komut dosyasına iki özellik iletir. Komut dosyası, **fteAnt** komutu kullanılarak çalıştırılır. `successrc` değeri, 0, 3, 4, 6 ve 14 dönüş kodlarının başarıyı gösterdiğini belirten `>2&<7&!5|0|14` olarak belirtilir.

```
antscript:myscript(prop1=fred,prop2=bob),,,>2&<7&!5|0|14
```

Bu programı aktarım başlamadan önce AGENT2 hedef aracısında çalıştırmak için aşağıdaki komutu kullanın:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -predst  
"antscript:myscript(prop1=fred,prop2=bob),,,>2&<7&!5|0|14" destinationSpecification sourceSpecification
```

Ant komut dosyasının çalıştırılması ve çağrılan hedeflerin belirtilmesi

Aşağıdaki örnek, `script2` adlı bir Ant komut dosyasını ve çağrılacak iki hedefi (`target1` ve `target2`) belirtir. `prop1` özelliği de `recmfm(F,B)` değeriyle iletilir. Parantezler, virgüller (,) ve ters eğik çizgiler (\) MFT komutlarında özel karakterlerdir ve ters eğik çizgi (\) karakteri kullanılmalıdır.  Windows Windows 'ta dosya yolları, ayrıncı olarak çift ters eğik çizgi (\\) kullanılarak ya da tek eğik çizgi (/) kullanılarak belirtilebilir. Aşağıdaki örnekte, virgül (,) ve parantezlere ters eğik çizgi karakteri (\) kullanılarak çıkış karakteri eklenmektedir.

```
antscript:script2(target1,target2,prop1=recmfm(F,B\)),,,>2&<7&!5|0|14
```

Bu programı aktarım tamamlandıktan sonra AGENT2 hedef aracısında çalıştırmak için aşağıdaki komutu kullanın:

```
fteCreateTransfer -sa AGENT1 -da AGENT2  
-postdst "antscript:script2(target1,target2,prop1=recmfm(F,B\)),,,>2&<7&!5|0|14"  
destinationSpecification sourceSpecification
```

Ant komut dosyasında meta verileri kullanma

Bir Ant görevini, aktarım için aşağıdaki çağrılardan herhangi biri olarak belirtebilirsiniz:

- ön kaynak
- gönderi kaynağı
- önhedef
- gönderi hedefi

Ant görevi çalıştırıldığında, aktarımın kullanıcı meta verileri ortam değişkenleri kullanılarak kullanılabilir hale getirilir. Bu verilere aşağıdaki kodu kullanarak erişebilirsiniz:

```
<property environment="environment" />  
<echo>${environment.mymetadata}</echo>
```

Burada `mymetadata`, aktarıma eklenen bazı meta verilerin adıdır.

JCL komut dosyasının çalıştırılması

Aşağıdaki örnek, ZOSBATCH adlı bir JCL komut dosyasını belirtir. `retrycount` için 3 değeri, `retrywait` için 30 değeri ve `successrc` için 0 değeri belirtilir. Bu değerler, başarılı bir dönüş kodu 0 döndürmezse komut dosyasının üç kez yeniden denendiği ve her deneme arasında otuz saniye beklediği anlamına gelir.

```
jcl:ZOSBATCH,3,30,0
```

Burada ZOSBATCH, MYSYS.JCL ve `agent.properties` dosyası şu satırı içerir
`commandPath=.....:/'MYSYS.JCL':....`

Aktarım tamamlandıktan sonra bu programı AGENT1 kaynak aracısında çalıştırmak için aşağıdaki komutu kullanın:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc jcl:ZOSBATCH,3,30,0  
destinationSpecification sourceSpecification
```

İlgili görevler

MFT ile çalıştırılacak programları belirtme

İlgili başvurular

fteCreateTransfer: yeni bir dosya aktarımı başlatır

fteAnt: Ant görevlerini MFT içinde çalıştırma

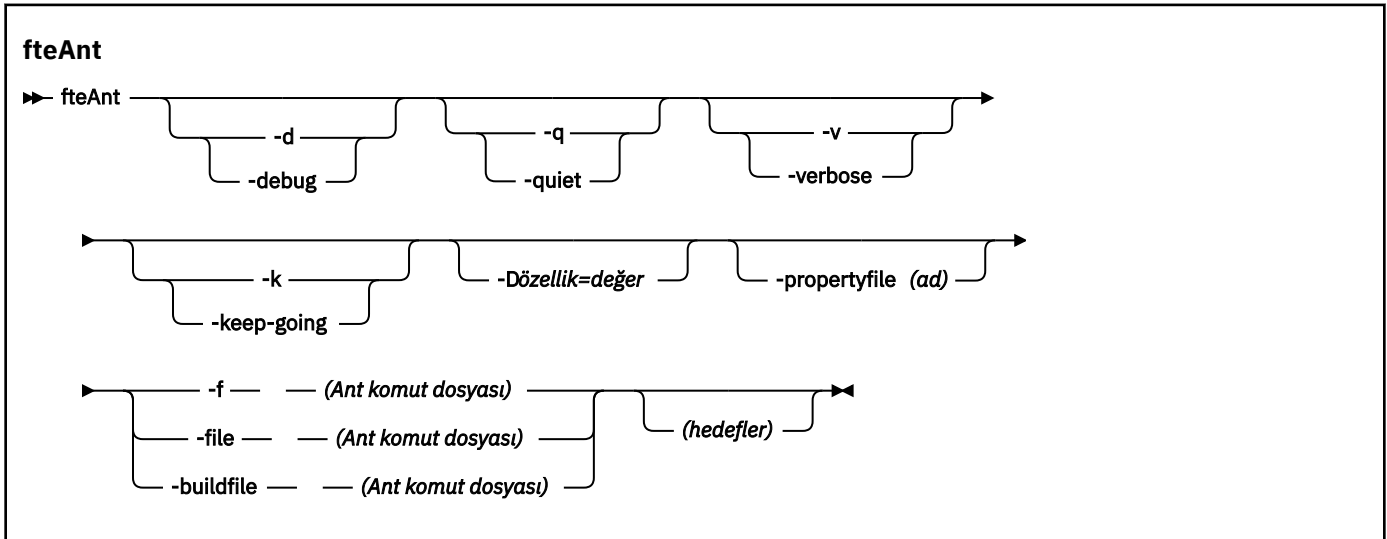
fteAnt komutu, kullanılabilir Managed File Transfer Ant görevleri olan bir ortamda Ant komut dosyalarını çalıştırır. Standart **ant** komutunun tersine, **fteAnt** bir komut dosyası tanımlamanızı gerektirir.

MFT Ant görevleri ve iç içe geçmiş parametreleri

Managed File Transfer , dosya aktarma yeteneklerine erişmek için kullanabileceğiniz bir dizi Ant görevi sağlar. Ayrıca, iç içe yerleştirilmiş bir değiştirge kümesi de vardır; bu değiştirgeler, verilen Ant görevlerinin bazılarında ortak olan iç içe yerleştirilmiş öge kümelerini açıklar.

fteAnt komutunun sözdizimi, parametreleri, kullanım örneği ve dönüş kodları bu konunun geri kalanında açıklanmıştır. MFT tarafından sağlanan Ant görevlerine ve iç içe geçmiş parametrelere ilişkin ayrıntılar için, alt başlıklara bakın.

fteAnt sözdizimi



Parametreler

-debug ya da -d

İsteğe Bağlı. Hata ayıklama çıkışı oluşturur.

-sessiz ya da -q

İsteğe Bağlı. Minimum çıkış üret.

-verbose ya da -v

İsteğe Bağlı. Ayrıntılı çıkış oluştur.

-devam et ya da -k

İsteğe Bağlı. Başarısız hedeflere bağlı olmayan tüm hedefleri yürütün.

-D özellik=değer

İsteğe Bağlı. Belirli bir *özelliği* için *değer* değerini kullanın. **-D** ile ayarlanan özellikler, bir özellikler dosyasında ayarlanan özelliklerden önceliklidir.

Ant görevleri için kullanılan yapılandırma seçenekleri kümesini belirtmek için

com.ibm.wmqfte.propertyset özelliğini kullanın. Bu özelliğin değeri olarak varsayılan olmayan bir eşgüdüm kuyruğu yöneticisinin adını kullanın. Ant görevleri daha sonra, bu varsayılan olmayan eşgüdüm kuyruğu yöneticisiyle ilişkilendirilmiş yapılanış seçenekleri kümesini kullanır. Bu özelliği belirtmezseniz, varsayılan eşgüdüm kuyruğu yöneticisine dayalı varsayılan yapılanış seçenekleri kümesi kullanılır. Bir Ant görevi için **cmdqm** özniteliğini belirtirseniz, bu öznitelik **fteAnt** komutu için belirtilen yapılandırma seçenekleri kümesinden önceliklidir. Bu davranış, varsayılan yapılandırma seçenekleri kümesini kullanıyor ya da **com.ibm.wmqfte.propertyset** özelliğiyle bir küme belirtip belirtmediğinizden bağımsız olarak geçerlidir.

-propertyfile (ad)

İsteğe Bağlı. **-D** özellikleri öncelikli olan bir dosyadan tüm özellikleri yükleyin.

-f (Ant komut dosyası), -file (Ant komut dosyası) ya da -buildfile (Ant komut dosyası)

Gereklidir. Çalıştırılacak Ant komut dosyasının adını belirtir.

hedefler

İsteğe Bağlı. Ant komut dosyasından çalıştırılacak bir ya da daha fazla hedefin adı. Bu parametre için bir değer belirtmezseniz, komut dosyası için varsayılan hedef çalıştırılır.

-version

İsteğe Bağlı. Managed File Transfer komutunu ve Ant sürümlerini görüntüler.

-Ne? ya da -h

İsteğe Bağlı. Komut sözdizimini görüntüler.

Örnek

In this example, the target **copy** in Ant script `fte_script.xml` is run and the command writes debugging output to standard out.

```
fteAnt -d -f fte_script.xml copy
```

Dönüş kodları

0

Komut başarıyla tamamlandı.

1

Komut başarısız oldu.

Diğer durum dönüş kodları Ant komut dosyalarından da belirtilebilir; örneğin, Ant başarısız görevi kullanılarak.

Daha fazla bilgi için bkz. [Başarısız](#) .

İlgili kavramlar

[MFT ile Ant komut dosyalarını kullanmaya başlama](#)

İlgili görevler

[Apache Ant 'yi MFT ile kullanma](#)

İlgili başvurular

[MFT için örnek Ant görevleri](#)

fte: bekleme sonucu Ant görevi

Bir **fte:filecopy**, **fte:filemove** ya da **fte:call** işleminin tamamlanmasını bekler.

Öznitelikler

tanıtıcı

Gereklidir. Sonucun bekleneceği aktarımı tanımlar. Genellikle bu, `fte:filecopy`, `fte:filetaşımaya` da `fte:call` görevlerinin `idProperty` özneliği tarafından ayarlanan bir özelliktir.

rcproperty

Gereklidir. `fte:awaitoutcome` görevinin dönüş kodunun saklanacağı bir özelliği adlandırıyor.

zaman aşımı

İsteğe Bağlı. İşlemin tamamlanması için beklenecek süre üst sınırı (saniye). Zamanaşımı alt sınırı bir saniyedir. Bir zamanaşımı değeri belirtmezseniz, `fte:awaitoutcome` görevi işlemin sonucunun belirlenmesini sonsuza kadar bekler.

Örnek

Bu örnekte bir dosya kopyası başlatılır ve tanıtıcısı `copy.id` özelliğinde saklanır. Kopyalama işlemi devam ederken diğer işlemler de gerçekleştirilebilir. `fte:awaitoutcome` deyimi, kopyalama işleminin tamamlanmasını beklemek için kullanılır. `fte:awaitoutcome` deyimi, `copy.id` özelliğinde saklanan tanıtıcıyı kullanmak için beklenecek işlemi tanımlar. `fte:awaitoutcome`, kopyalama işleminin sonucunu gösteren bir dönüş kodunu `copy.result` adlı bir özelliğe saklar.

```
<-- 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>
```

İlgili görevler

[Apache Ant 'yi MFT ile kullanma](#)

fte: call Ant görevi

Komut dosyalarını ve programları uzaktan çağırmak için `fte:call` görevini kullanabilirsiniz.

Bu görev, bir aracıya `fte:call` isteği göndermenizi sağlar. Aracı, bir komut dosyası ya da program çalıştırarak ve sonucu döndürerek bu isteği işler. Çağrılacak komutlar aracı tarafından erişilebilir olmalıdır. `agent.properties` dosyasındaki `commandPath` özellik değerinin, çağrılması gereken komutların konumunu içerdiğinden emin olun. Komut iç içe geçmiş öğesi tarafından belirtilen yol bilgileri, `commandPath` özelliği tarafından belirtilen konumlarla görece olmalıdır. Varsayılan olarak `commandPath`, aracının herhangi bir komutu çağıramayacağı için boştur. Bu özellik hakkında daha fazla bilgi için bkz. [commandPath MFT özellik](#).

`agent.properties` dosyası hakkında daha fazla bilgi için bkz. [MFT agent.properties dosyası](#).

Öznitelikler

aracı

Gereklidir. `fte:call` isteğinin gönderileceği aracıyı belirtir. Aracı bilgilerini şu biçimde belirtin: `agentname@qmgrname`; burada `agentname`, aracının adıdır ve `qmgrname`, bu aracının doğrudan bağlı olduğu kuyruk yöneticisinin adıdır.

cmdqm

İsteğe Bağlı. İsteğin gönderileceği komut kuyruğu yöneticisi. Bu bilgileri `qmgrname@host@port@channel` biçiminde belirtin; burada:

- *qmgrname* , kuyruk yöneticisinin adıdır
- *host* , kuyruk yöneticisinin çalıştığı sistemin isteğe bağlı anasistem adıdır
- *port* , kuyruk yöneticisinin dinlediği isteğe bağlı kapı numarasıdır
- *channel* , kullanılacak isteğe bağlı SVRCONN kanalıdır

Komut kuyruğu yöneticisine ilişkin *host*, *port* ya da *channel* bilgilerini atlarsanız, `command.properties` dosyasında belirtilen bağlantı bilgileri kullanılır.



Uyarı: Aşağıdakiler için değer belirtilmezse:

- *program* değişkeni, bağ tanımlama kipi kullanılır
- *kapı* değişkeni, 1414 değeri kullanılır
- *kanal* değişkeni, SYSTEM.DEF.SVRCONN kullanılır.

Daha fazla bilgi için bkz. [MFT command.properties dosyası](#) .

Ancak, ortadaki öznitelikleri atlayamazsınız; örneğin, *qmgrname@host@@channel*. Örneğin, *qmgrname@host*, *qmgrname@host@port* ya da *qmgrname@hostport@@channel*.

MFT , @ sınırlayıcısını kullanarak belirtilen özniteliği böler. Bulunan simge sayısına bağlı olarak, ilk simgeyi *qmgrname*, ikincisi *anasistem*, üçüncüsü *kapı* ve son olarak *kanal* olarak alır.

Daha fazla bilgi için bkz. [MFT command.properties dosyası](#).

Hangi `command.properties` dosyasının kullanılacağını belirtmek için

com.ibm.wmqfte.propertySet özelliğini kullanabilirsiniz. Daha fazla bilgi için bkz. [com.ibm.wmqfte.propertySet](#).

`cmdqm` özniteliğini kullanmazsanız, bu özellik ayarlanırsa, görev

varsayılan olarak `com.ibm.wmqfte.ant.commandQueueManager` özelliğini kullanır.

`com.ibm.wmqfte.ant.commandQueueManager` özelliği ayarlanmazsa, `command.properties` dosyasında tanımlı olan varsayılan kuyruk yöneticisine bağlanma girişiminde bulunur.

`com.ibm.wmqfte.ant.commandQueueManager` özelliğinin biçimi, `cmdqm` özniteliğiyle (yani, *qmgrname@host@port@channel*) aynıdır.

idproperty

`outcome / defer` belirtmediğiniz sürece isteğe bağlıdır. Aktarma tanıtıcısının atanacağı özelliğin adını belirler. Aktarma tanıtıcıları, aktarma isteğinin gönderildiği noktada oluşturulur ve aktarma işleminin ilerleyişini izlemek, aktarma sorunlarını tanılamak ve aktarma işlemi iptal etmek için aktarma tanıtıcılarını kullanabilirsiniz.

`ignore outcome` özelliğini de belirttiyseniz, bu özelliği belirtebilirsiniz. Ancak, `defer outcome` özelliğini de belirttiyseniz, `idproperty` belirtmeniz gerekir.

jobName

İsteğe Bağlı. **fte:call** isteğine bir iş adı atar. Mantıksal aktarım grupları oluşturmak için iş adlarını kullanabilirsiniz. Sözde benzersiz iş adları oluşturmak için "[fte: uuid Ant görevi](#)" sayfa 2081 görevini kullanın. `jobname` özniteliğini kullanmazsanız, bu özellik ayarlanırsa, görev varsayılan olarak `com.ibm.wmqfte.ant.jobName` özellik değerini kullanır. Bu özelliği ayarlamazsanız, iş adı **fte:call** isteğiyle ilişkilendirilmez.

yönlendir

İsteğe Bağlı. **fte:call** isteğiyle ilişkilendirilecek kaynak kullanıcı kimliğini belirtir. `origuser` özniteliğini kullanmazsanız, görev varsayılan olarak Ant komut dosyasını çalıştırmak için kullanılan kullanıcı kimliğini kullanır.

sonuç

İsteğe Bağlı. Görevin, denetimi Ant komut dosyasına döndürmeden önce **fte:call** işleminin tamamlanmasını bekleyip beklemeyeceğini belirler. Aşağıdaki seçeneklerden birini belirleyin:

bekleme

Görev, **fte:call** işleminin geri dönmeden önce tamamlanmasını bekler. Bir `outcome / await` belirtildiğinde `idproperty` özniteliği isteğe bağlıdır.

Ertele

Görev, **fte:call** isteği gönderilir sunulmaz geri döner ve çağrı işleminin sonucunun daha sonra `awaitoutcome` ya da `ignoreoutcome` görevleri kullanılarak ele alındığını varsayar. Bir `outcome / defer` belirtildiğinde `idproperty` özneliği gereklidir.

yoksay

fte:call işleminin sonucu önemli değilse, `ignoredeğeri`ni belirtebilirsiniz. Daha sonra görev, **fte:call** isteği sunulduğu anda, komutun sonucunu izlemek için kaynak ayırmadan geri döner. Bir `outcome / ignore` belirtildiğinde `idproperty` özneliği belirtilemez.

kazanım özneliğini belirtmezseniz, görev varsayılan olarak `awaitdeğeri`ni kullanır.

rcözellik

İsteğe Bağlı. **fte:call** isteğinin sonuç kodunun atanacağı özelliğin adını belirtir. Sonuç kodu, **fte:call** isteğinin genel sonucunu yansıtır.

`ignore` ya da `defer outcome` özelliğini de belirttiyseniz, bu özelliği belirtebilirsiniz. Ancak, `awaitsonucu` belirttiyseniz `rcproperty` belirtmeniz gerekir.

İç içe geçmiş öğeler olarak belirtilen parametreler

fte: komut

Aracı tarafından çağrılacak komutu belirtir. Tek bir `fte:command` öğesini belirli bir **fte:call** işlemiyle ilişkilendirebilirsiniz. Çağrılacak komut, aracının `agent.properties` dosyasında `commandPath` özelliği tarafından belirtilen yolda bulunmalıdır.

fte: meta veri

Arama işlemiyle ilişkilendirilecek meta verileri belirtebilirsiniz. Bu meta veriler, çağrı işlemi tarafından oluşturulan günlük iletilerine kaydedilir. Tek bir meta veri öbeğini belirli bir aktarım öğesiyle ilişkilendirebilirsiniz; ancak bu öbek birçok meta veri parçası içerebilir.

Örnek

Bu örnek, QM1kuyruk yöneticisinde çalışan AGENT1 ' de bir komutun nasıl çağrılacağını gösterir. Çağrılacak komut dosyası `command.sh` komut dosyasıdır ve komut dosyası tek bir `xyz` bağımsız değişkeniyle çağrılır. `command.sh` komutu, aracının `agent.properties` dosyasında `commandPath` özelliği tarafından belirtilen yolda bulunur.

```
<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>
```

İlgili görevler

[Apache Ant ' yi MFT ile kullanma](#)

fte: Ant görevini iptal et

Managed File Transfer tarafından yönetilen ya da yönetilen bir aramayı iptal eder. **fte:filecopy** ya da **fte:filemove** görevleri kullanılarak yönetilen bir aktarım yaratılmış olabilir. **fte:call** görevi kullanılarak yönetilen bir çağrı oluşturulmuş olabilir.

Öznitelikler

aracı

Gereklidir. **fte:cancel** isteğinin gönderileceği aracıyı belirtir. Değer şu biçimdedir: *agentname@qmgrname* ; burada *agentname* , aracının adı ve *qmgrname* , bu aracının doğrudan bağlı olduğu kuyruk yöneticisinin adıdır.

cmdqm

İsteğe Bağlı. İsteğin gönderileceği komut kuyruğu yöneticisi. Bu bilgileri *qmgrname@host@port@channel* biçiminde belirtin; burada:

- *qmgrname* , kuyruk yöneticisinin adıdır
- *host* , kuyruk yöneticisinin çalıştığı sistemin isteğe bağlı anasistem adıdır
- *port* , kuyruk yöneticisinin dinlediği isteğe bağlı kapı numarasıdır
- *channel* , kullanılacak isteğe bağlı SVRCONN kanalıdır

Komut kuyruğu yöneticisine ilişkin *host* , *port* ya da *channel* bilgilerini atlarsanız, `command.properties` dosyasında belirtilen bağlantı bilgileri kullanılır.



Uyarı: Aşağıdakiler için değer belirtilmezse:

- *program* değişkeni, bağ tanımlama kipi kullanılır
- *kapı* değişkeni, 1414 değeri kullanılır
- *kanal* değişkeni, SYSTEM.DEF.SVRCONN kullanılır.

Daha fazla bilgi için bkz. [MFT command.properties](#) dosyası .

Ancak, ortadaki öznitelikleri atlayamazsınız; örneğin, *qmgrname@host@@channel*. Örneğin, *qmgrname@host*, *qmgrname@host@port* ya da *qmgrname@hostport@@channel*.

MFT , @ sınırlayıcısını kullanarak belirtilen özniteliği böler. Bulunan simge sayısına bağlı olarak, ilk simgeyi *qmgrname*, ikincisi *anasistem*, üçüncüsü *kapı* ve son olarak *kanal* olarak alır.

Daha fazla bilgi için bkz. [MFT command.properties](#) dosyası.

Hangi `command.properties` dosyasının kullanılacağını belirtmek için

com.ibm.wmqfte.propertySet özelliğini kullanabilirsiniz. Daha fazla bilgi için bkz. [com.ibm.wmqfte.propertySet](#).

`cmdqm` özniteliğini kullanmazsanız, bu özellik ayarlanırsa, görev varsayılan olarak `com.ibm.wmqfte.ant.commandQueueManager` özelliğini kullanır. `com.ibm.wmqfte.ant.commandQueueManager` özelliği ayarlanmazsa, `command.properties` dosyasında tanımlı olan varsayılan kuyruk yöneticisine bağlanma girişiminde bulunur. `com.ibm.wmqfte.ant.commandQueueManager` özelliğinin biçimi, `cmdqm` özniteliğiyle (yani, *qmgrname@host@port@channel*) aynıdır.

tanıtıcı

Gereklidir. İptal edilecek aktarma işleminin aktarma tanıtıcısını belirler. Aktarma tanıtıcıları, hem [fte:filecopy](#) , hem de [fte:filetaşıma](#) görevleri tarafından bir aktarma isteği gönderildiği noktada oluşturulur.

yönlendir

İsteğe Bağlı. **cancel** isteğiyle ilişkilendirilecek kaynak kullanıcı kimliğini belirtir. `origuser` özniteliği kullanılmazsa, görev varsayılan olarak `Ant` komut dosyasını çalıştırmak için kullanılan kullanıcı kimliğini kullanır.

Örnek

Bu örnek, `qm0` komut kuyruğu yöneticisine bir **fte:cancel** isteği gönderir. **fte:cancel** isteği, `transfer.id` değişkeninin veri yerleştiği aktarma tanıtıcısı için `agent1` kuyruk yöneticisinde `qm1` hedeflenmiştir. İstek, "bob" kullanıcı kimliği kullanılarak çalıştırılır.

```
<fte:cancel cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
agent="agent1@qm1"
```

```
id="{transfer.id}"
origuser="bob"/>
```

İlgili görevler

Apache Ant 'yi MFT ile kullanma

fte: filecopy Ant görevi

fte:filecopy görevi, dosyaları Managed File Transfer araçları arasında kopyalar. Dosya kaynak araçtan silinmedi.

Öznitelikler

cmdqm

İsteğe Bağlı. İsteğin gönderileceği komut kuyruğu yöneticisi. Bu bilgileri `qmgrname@host@port@channel` biçiminde belirtin; burada:

- `qmgrname` , kuyruk yöneticisinin adıdır
- `host` , kuyruk yöneticisinin çalıştığı sistemin isteğe bağlı anasistem adıdır
- `port` , kuyruk yöneticisinin dinlediği isteğe bağlı kapı numarasıdır
- `channel` , kullanılacak isteğe bağlı SVRCONN kanalıdır

Komut kuyruğu yöneticisine ilişkin `host`, `port` ya da `channel` bilgilerini atlarsanız, `command.properties` dosyasında belirtilen bağlantı bilgileri kullanılır.



Uyarı: Aşağıdakiler için değer belirtilmezse:

- `program` değişkeni, bağ tanımlama kipi kullanılır
- `kapı` değişkeni, 1414 değeri kullanılır
- `kanal` değişkeni, SYSTEM.DEF.SVRCONN kullanılır.

Daha fazla bilgi için bkz. [MFT command.properties dosyası](#) .

Ancak, ortadaki öznitelikleri atlayamazsınız; örneğin, `qmgrname@host@@channel`. Örneğin, `qmgrname@host`, `qmgrname@host@port` ya da `qmgrname@hostport@@channel`.

MFT , @ sınırlayıcısını kullanarak belirtilen özniteliği böler. Bulunan simge sayısına bağlı olarak, ilk simgeyi `qmgrname`, ikincisi `anasistem`, üçüncüsü `kapı` ve son olarak `kanal` olarak alır.

Daha fazla bilgi için bkz. [MFT command.properties dosyası](#).

Hangi `command.properties` dosyasının kullanılacağını belirtmek için

com.ibm.wmqfte.propertySet özelliğini kullanabilirsiniz. Daha fazla bilgi için bkz. [com.ibm.wmqfte.propertySet](#).

`cmdqm` özniteliğini kullanmazsanız, bu özellik ayarlanırsa, görev

varsayılan olarak `com.ibm.wmqfte.ant.commandQueueManager` özelliğini kullanır.

`com.ibm.wmqfte.ant.commandQueueManager` özelliği ayarlanmazsa, `command.properties` dosyasında tanımlı olan varsayılan kuyruk yöneticisine bağlanma girişiminde bulunur.

`com.ibm.wmqfte.ant.commandQueueManager` özelliğinin biçimi, `cmdqm` özniteliğiyle (yani, `qmgrname@host@port@channel`) aynıdır.

dst

Gereklidir. Kopyalama işlemine ilişkin hedef aracıyı belirtir. Bu bilgileri şu biçimde belirtin:

`agentname@qmgrname` ; burada `agentname` , hedef aracının adı ve `qmgrname` , bu aracının doğrudan bağlı olduğu kuyruk yöneticisinin adıdır.

idproperty

`outcome / defer` belirtmediğiniz sürece isteğe bağlıdır. Aktarma tanıtıcısının atanacağı özelliğin adını belirler. Aktarma tanıtıcıları, aktarma isteğinin gönderildiği noktada oluşturulur ve aktarma işleminin ilerleyişini izlemek, aktarma sorunlarını tanılamak ve aktarma işlemini iptal etmek için aktarma tanıtıcılarını kullanabilirsiniz.

ignore outcome özelliğini de belirttiyseniz, bu özelliği belirtebilirsiniz. Ancak, defer outcome özelliğini de belirttiyseniz, idproperty belirtmeniz gerekir.

jobName

İsteğe Bağlı. Kopyalama isteğine bir iş adı atar. Mantıksal aktarım grupları oluşturmak için iş adlarını kullanabilirsiniz. Sözde benzersiz iş adları oluşturmak için "[fte: uuid Ant görevi](#)" sayfa 2081 görevini kullanın. jobname özniteliğini kullanmazsanız, bu özellik ayarlanırsa, görev varsayılan olarak com.ibm.wmqfte.ant.jobName özellik değerini kullanır. Bu özelliği ayarlamazsanız, kopyalama isteğiyle ilişkilendirilmiş bir iş adı yoktur.

yönlendir

İsteğe Bağlı. Kopyalama isteğiyle ilişkilendirilecek kaynak kullanıcı kimliğini belirler. origuser özniteliğini kullanmazsanız, görev varsayılan olarak Ant komut dosyasını çalıştırmak için kullanılan kullanıcı kimliğini kullanır.

sonuç

İsteğe Bağlı. Görevin, denetimi Ant komut dosyasına döndürmeden önce kopyalama işleminin tamamlanmasını bekleyip beklemeyeceğini belirler. Aşağıdaki seçeneklerden birini belirleyin:

bekleme

Görev, geri dönmeden önce kopyalama işleminin tamamlanmasını bekler. Bir outcome / await belirtildiğinde idproperty özniteliği isteğe bağlıdır.

Ertele

Görev, kopyalama isteği sunulduğu anda döndürülür ve kopyalama işleminin sonucunun daha sonra "[fte: bekleme sonucu Ant görevi](#)" sayfa 2067 ya da "[fte: ignoreoutcome Ant task](#)" sayfa 2079 görevleri kullanılarak ele alındığını varsayar. Bir outcome / defer belirtildiğinde idproperty özniteliği gereklidir.

yoksay

Kopyalama işleminin sonucu önemli değilse, ignoredeğerini belirtebilirsiniz. Bundan sonra görev, kopyalama isteği sunulduğu anda, aktarımın sonucunu izlemek için kaynak ayırmadan geri döner. Bir outcome / ignore belirtildiğinde idproperty özniteliği belirtilemez.

kazanım özniteliğini belirtmezseniz, görev varsayılan olarak awaitdeğerini kullanır.

öncelik

İsteğe Bağlı. Kopyalama isteğiyle ilişkilendirilecek önceliği belirtir. Genel olarak, yüksek öncelikli aktarma istekleri düşük öncelikli isteklere göre önceliklidir. Öncelik değeri 0-9 (bu değerler de içinde olmak üzere) aralığında olmalıdır. 0 öncelik değeri en düşük öncelik, 9 ise en yüksek öncelik değeridir. priority özniteliğini belirtmezseniz, aktarma varsayılan olarak 0 önceliğine ayarlanır.

rcözellik

İsteğe Bağlı. Kopyalama isteğinin sonuç kodunun atanacağı özelliğin adını belirtir. Sonuç kodu, kopyalama isteğinin genel sonucunu yansıtır.

ignore ya da defer outcome özelliğini de belirttiyseniz, bu özelliği belirtebilirsiniz. Ancak, awaitsonucu belirtirseniz rcproperty belirtmeniz gerekir.

transferRecoveryZamanaşımı

İsteğe Bağlı. Kaynak aracısının durmuş bir dosya aktarımını kurtarmaya çalıştığı süreyi saniye cinsinden ayarlar. Aşağıdaki seçeneklerden birini belirleyin:

-1

Aracı, aktarım tamamlanıncaya kadar durdurulmuş aktarımı kurtarmaya çalışmaya devam eder. Bu seçeneğin kullanılması, özellik ayarlanmadığında aracının varsayılan davranışının eşdeğeridir.

0

Aracı, kurtarma işlemine girer girmez dosya aktarımını durdurur.

>0

Aracı, belirtilen pozitif tamsayı değeriyle belirlendiği şekilde, saniye cinsinden süre boyunca durdurulmuş aktarımı kurtarma girişiminde bulunmaya devam eder. Örneğin,

```
<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>
```

Aracının, kurtarma işlemine girdiği andan itibaren 6 saat boyunca aktarımı kurtarmaya çalıştığını gösterir. Bu öznitelik için değer üst sınırı: 999999999.

Aktarma kurtarma zamanasını değerinin bu şekilde belirlenmesi, aktarma temelinde bu değeri belirler. Managed File Transfer ağındaki tüm aktarımlar için bir genel değer belirlemek üzere [Aktarma kurtarma zamanasını özellikleri](#)' ne bir özellik ekleyebilirsiniz. Daha fazla bilgi için bkz. [Kurtarma sırasında aktarımlar için zamanasını seçeneği](#).

src

Gereklidir. Kopyalama işlemi için kaynak aracıyı belirtir. Bu bilgileri şu biçimde belirtin: *agentname@qmgrname* ; burada *agentname* kaynak aracının adıdır ve *qmgrname* , bu aracının doğrudan bağlı olduğu kuyruk yöneticisinin adıdır.

İç içe geçmiş öğeler olarak belirtilen parametreler

fte: filShakespeare

Gereklidir. Kopyalanacak dosyaları tanıtan en az bir dosya belirtimi belirtmelisiniz. Gerekirse birden çok dosya belirtimi belirtebilirsiniz. Ek bilgi için bkz. [“fte: filShakespeare Ant içiçe yerleştirilmiş öğe” sayfa 2081](#) .

fte: meta veri

Kopyalama işlemiyle ilişkilendirilecek meta verileri belirtebilirsiniz. Bu meta veriler aktarımla birlikte taşınır ve aktarma tarafından oluşturulan günlük iletilerine kaydedilir. Tek bir meta veri öbeğini belirli bir aktarım ögesiyle ilişkilendirebilirsiniz; ancak bu öbek birçok meta veri parçası içerebilir. Daha fazla bilgi için [fte: metadata](#) başlıklı konuya bakın.

fte: presrc

Aktarım başlamadan önce kaynak aracıda gerçekleşmesi gereken program çağrısını belirtir. Tek bir *fte: presrc* ögesini belirli bir aktarımla ilişkilendirebilirsiniz. Ek bilgi için [program başlatma](#) başlıklı konuya bakın.

fte: predst

Aktarma başlamadan önce hedef aracıda gerçekleşecek program çağrısını belirtir. Tek bir *fte: predst* ögesini belirli bir aktarımla ilişkilendirebilirsiniz. Ek bilgi için [program başlatma](#) başlıklı konuya bakın.

fte: postsrc

Aktarım tamamlandıktan sonra kaynak aracıda gerçekleşecek program çağrısını belirtir. Tek bir *fte: postsrc* ögesini belirli bir aktarımla ilişkilendirebilirsiniz. Ek bilgi için [program başlatma](#) başlıklı konuya bakın.

fte: postdst

Aktarma tamamlandıktan sonra hedef aracıda gerçekleşecek program çağrısını belirtir. Tek bir *fte: postdst* ögesini belirli bir aktarımla ilişkilendirebilirsiniz. Ek bilgi için [program başlatma](#) başlıklı konuya bakın.

fte:presrc, *fte:predst*, *fte:postsrc*, *fte:postdst* ve çıkışlar bir başarı durumu döndürmezse, kurallar belirtilen sırada aşağıdaki gibidir:

1. Kaynak başlatma çıkışlarını çalıştırın. Kaynak başlatma çıkışları başarısız olursa, aktarma başarısız olur ve başka bir işlem çalıştırılmaz.
2. Kaynak öncesi çağrıyı çalıştırın (varsa). Kaynak öncesi çağrı başarısız olursa, aktarma başarısız olur ve başka bir şey çalıştırılmaz.
3. Hedef başlatma çıkışlarını çalıştırın. Hedef başlatma çıkışları başarısız olursa, aktarma başarısız olur ve başka bir işlem yürütülmez.
4. Hedef öncesi çağrıyı çalıştırın (varsa). Hedef öncesi çağrı başarısız olursa, aktarma başarısız olur ve başka hiçbir şey çalıştırılmaz.
5. Dosya aktarımlarını gerçekleştirin.

6. Hedef uç çıkışlarını çalıştırın. Bu çıkışlar için arıza durumu yok.
7. Aktarma başarılı olursa (bazı dosyalar başarıyla aktarılırsa, başarılı olarak kabul edilir), hedef sonrası çağrışı (varsa) yürütür. Hedef sonrası çağrı başarısız olursa, aktarma başarısız olur.
8. Kaynak uç çıkışlarını çalıştırın. Bu çıkışlar için arıza durumu yok.
9. Aktarma başarılı olursa, kaynak sonrası çağrışı (varsa) çalıştırın. Kaynak sonrası çağrı başarısız olursa aktarma başarısız olur.

Örnekler

Bu örnek, agent1 ile agent2 arasında temel bir dosya aktarımını gösterir. Dosya aktarımını başlatma komutu, istemci iletim kipi bağlantısı kullanılarak qm0, adlı bir kuyruk yöneticisine gönderilir. Dosya aktarma işleminin sonucu, copy.result adlı özelliğe atanır.

```
<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>
```

Bu örnek, aynı dosya aktarımını gösterir; ancak, meta veriler eklenerek ve aktarım tamamlandıktan sonra kaynak aracıda bir program başlatıldığında.

```
<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>
```

İlgili kavramlar

Kurtarmada [dosya aktarımları için zamanaşımı seçeneği](#)

İlgili görevler

[Apache Ant 'yi MFT ile kullanma](#)

fte: filetaşıma Ant görevi

fte:filemove görevi, dosyaları Managed File Transfer araçları arasında taşır. Bir dosya kaynak araçından hedef aracıya başarıyla aktarıldığında, dosya kaynak araçından silinir.

Öznitelikler

cmdqm

İsteğe Bağlı. İsteğin gönderileceği komut kuyruğu yöneticisi. Bu bilgileri *qmgrname@host@port@channel* biçiminde belirtin; burada:

- *qmgrname*, kuyruk yöneticisinin adıdır
- *host*, kuyruk yöneticisinin çalıştığı sistemin isteğe bağlı anasistem adıdır
- *port*, kuyruk yöneticisinin dinlediği isteğe bağlı kapı numarasıdır
- *channel*, kullanılacak isteğe bağlı SVRCONN kanalıdır

Komut kuyruğu yöneticisine ilişkin *host*, *port* ya da *channel* bilgilerini atlarsanız, `command.properties` dosyasında belirtilen bağlantı bilgileri kullanılır.



Uyarı: Aşağıdakiler için değer belirtilmezse:

- *program* değişkeni, bağ tanımlama kipi kullanılır
- *kapı* değişkeni, 1414 değeri kullanılır
- *kanal* değişkeni, SYSTEM.DEF.SVRCONN kullanılır.

Daha fazla bilgi için bkz. [MFT command.properties](#) dosyası .

Ancak, ortadaki öznitelikleri atlayamazsınız; örneğin, `qmgrname@host@@channel`. Örneğin, `qmgrname@host`, `qmgrname@host@port` ya da `qmgrname@hostport@@channel`.

MFT , @ sınırlayıcısını kullanarak belirtilen özneliği böler. Bulunan simge sayısına bağlı olarak, ilk simgeyi *qmgrname*, ikincisi *anasistem*, üçüncüsü *kapı* ve son olarak *kanal* olarak alır.

Daha fazla bilgi için bkz. [MFT command.properties](#) dosyası.

Hangi `command.properties` dosyasının kullanılacağını belirtmek için

com.ibm.wmqfte.propertySet özelliğini kullanabilirsiniz. Daha fazla bilgi için bkz. [com.ibm.wmqfte.propertySet](#).

`cmdqm` özneliğini kullanmazsanız, bu özellik ayarlanırsa, görev varsayılan olarak `com.ibm.wmqfte.ant.commandQueueManager` özelliğini kullanır. `com.ibm.wmqfte.ant.commandQueueManager` özelliği ayarlanmazsa, `command.properties` dosyasında tanımlı olan varsayılan kuyruk yöneticisine bağlanma girişiminde bulunur. `com.ibm.wmqfte.ant.commandQueueManager` özelliğinin biçimi, `cmdqm` özneliğiyle (yani, `qmgrname@host@port@channel`) aynıdır.

dst

Gereklidir. Kopyalama işlemine ilişkin hedef aracıyı belirtir. Bu bilgileri şu biçimde belirtin:

`agentname@qmgrname` ; burada *agentname* , hedef aracının adı ve *qmgrname* , bu aracının doğrudan bağlı olduğu kuyruk yöneticisinin adıdır.

idproperty

`outcome / defer` belirtmediğiniz sürece isteğe bağlıdır. Aktarma tanıtıcısının atanacağı özelliğin adını belirler. Aktarma tanıtıcıları, aktarma isteğinin gönderildiği noktada oluşturulur ve aktarma işleminin ilerleyişini izlemek, aktarma sorunlarını tanılamak ve aktarma işlemini iptal etmek için aktarma tanıtıcılarını kullanabilirsiniz.

`ignore outcome` özelliğini de belirttiyseniz, bu özelliği belirtebilirsiniz. Ancak, `defer outcome` özelliğini de belirttiyseniz, `idproperty` belirtmeniz gerekir.

jobName

İsteğe Bağlı. Taşıma isteğine bir iş adı atar. Mantıksal aktarım grupları oluşturmak için iş adlarını kullanabilirsiniz. Sözde benzersiz iş adları oluşturmak için `fte: uuid` görevini kullanın. `jobname` özneliğini kullanmazsanız, bu özellik ayarlanırsa, görev varsayılan olarak `com.ibm.wmqfte.ant.jobName` özellik değerini kullanır. Bu özelliği ayarlamazsanız, taşıma isteğiyle ilişkilendirilmiş bir iş adı yoktur.

yönlendir

İsteğe Bağlı. Taşıma isteğiyle ilişkilendirilecek kaynak kullanıcı kimliğini belirler. `origuser` özneliğini kullanmazsanız, görev varsayılan olarak Ant komut dosyasını çalıştırmak için kullanılan kullanıcı kimliğini kullanır.

sonuç

İsteğe Bağlı. Görevin, denetimi Ant komut dosyasına döndürmeden önce taşıma işleminin tamamlanmasını bekleyip beklemeyeceğini belirler. Aşağıdaki seçeneklerden birini belirleyin:

bekleme

Görev, geri dönmeden önce taşıma işleminin tamamlanmasını bekler. Bir `outcome / await` belirtildiğinde `idproperty` özneliği isteğe bağlıdır.

Ertele

Görev, taşıma isteği sunulduğu anda geri döner ve taşıma işleminin sonucunun daha sonra “fte: bekleme sonucu Ant görevi” sayfa 2067 ya da “fte: ignoreoutcome Ant task” sayfa 2079 görevi kullanılarak ele alındığını varsayar. Bir outcome / defer belirtildiğinde idproperty özneliği gereklidir.

yoksay

Taşıma işleminin sonucu önemli değilse, ignoredeğerini belirtebilirsiniz. Daha sonra, taşıma isteği sunulduğu anda, aktarma işleminin sonucunu izlemek için kaynak ayırmadan görev döndürülür. Bir outcome / ignore belirtildiğinde idproperty özneliği belirtilemez.

kazanım özneliğini belirtmezseniz, görev varsayılan olarak awaitdeğerini kullanır.

öncelik

İsteğe Bağlı. Taşıma isteğiyle ilişkilendirilecek önceliği belirtir. Genel olarak, yüksek öncelikli aktarma istekleri düşük öncelikli isteklere göre önceliklidir. Öncelik değeri 0-9 (bu değerler de içinde olmak üzere) aralığında olmalıdır. 0 öncelik değeri en düşük öncelik, 9 ise en yüksek öncelik değeridir. priority özneliğini belirtmezseniz, aktarma varsayılan olarak 0 önceliğine ayarlanır.

rcözellik

İsteğe Bağlı. Taşıma isteğinin sonuç kodunun atanacağı özelliğin adını belirtir. Sonuç kodu, taşıma isteğinin genel sonucunu yansıtır.

ignore ya da defer outcome özelliğini de belirttiyseniz, bu özelliği belirtebilirsiniz. Ancak, awaitsonucu belirttiyseniz rcproperty belirtmeniz gerekir.

transferRecoveryZamanaşımı

İsteğe Bağlı. Kaynak aracısının durmuş bir dosya aktarımını kurtarmaya çalıştığı süreyi saniye cinsinden ayarlar. Aşağıdaki seçeneklerden birini belirleyin:

-1

Aracı, aktarım tamamlanıncaya kadar durdurulmuş aktarımı kurtarmaya çalışmaya devam eder. Bu seçeneğin kullanılması, özellik ayarlanmadığında aracının varsayılan davranışının eşdeğeridir.

0

Aracı, kurtarma işlemine girer girmez dosya aktarımını durdurur.

>0

Aracı, belirtilen pozitif tamsayı değeriyle belirlendiği şekilde, saniye cinsinden süre boyunca durdurulmuş aktarımı kurtarma girişiminde bulunmaya devam eder. Örneğin,

```
<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
```

Aracının, kurtarma işlemine girdiği andan itibaren 6 saat boyunca aktarımı kurtarmaya çalıştığını gösterir. Bu öznelik için değer üst sınırı: 999999999.

Aktarma kurtarma zamanasını değerinin bu şekilde belirlenmesi, aktarma temelinde bu değeri belirler. Managed File Transfer ağındaki tüm aktarımlar için bir genel değer belirlemek üzere Aktarma kurtarma zamanasını özellikleri' ne bir özellik ekleyebilirsiniz. Daha fazla bilgi için bkz. Kurtarma sırasında aktarımlar için zamanasını seçeneği.

src

Gereklidir. Taşıma işlemine ilişkin kaynak aracıyı belirtir. Bu bilgileri şu biçimde belirtin: agentname@qmgrname ; burada agentname , kaynak aracının adı ve qmgrname , bu aracının doğrudan bağlı olduğu kuyruk yöneticisinin adıdır.

İç içe geçmiş öğeler olarak belirtilen parametreler

fte: filShakespeare

Gereklidir. Taşınacak dosyaları tanıtan en az bir dosya belirtimi belirtmelisiniz. Gerekirse birden çok dosya belirtimi belirtebilirsiniz. Ek bilgi için bkz. "[fte: filShakespeare Ant içiçe yerleştirilmiş öğe](#)" sayfa 2081 .

fte: meta veri

İsteğe Bağlı. Dosya taşıma işlemiyle ilişkilendirilecek meta verileri belirtebilirsiniz. Bu meta veriler aktarımla birlikte taşınır ve aktarma tarafından oluşturulan günlük iletilerine kaydedilir. Tek bir meta veri öbeğini belirli bir aktarım öğesiyle ilişkilendirebilirsiniz; ancak bu öbek birçok meta veri parçası içerebilir. Daha fazla bilgi için [fte: metadata](#) başlıklı konuya bakın.

fte: presrc

İsteğe Bağlı. Aktarım başlamadan önce kaynak araçıda gerçekleşmesi gereken program çağrısını belirtir. Tek bir `fte: presrc` öğesini belirli bir aktarımla ilişkilendirebilirsiniz. Ek bilgi için [program başlatma](#) başlıklı konuya bakın.

fte: predst

İsteğe Bağlı. Aktarma başlamadan önce hedef araçıda gerçekleşecek program çağrısını belirtir. Tek bir `fte: predst` öğesini belirli bir aktarımla ilişkilendirebilirsiniz. Ek bilgi için [program başlatma](#) başlıklı konuya bakın.

fte: postsrc

İsteğe Bağlı. Aktarım tamamlandıktan sonra kaynak araçıda gerçekleşecek program çağrısını belirtir. Tek bir `fte: postsrc` öğesini belirli bir aktarımla ilişkilendirebilirsiniz. Ek bilgi için [program başlatma](#) başlıklı konuya bakın.

fte: postdst

İsteğe Bağlı. Aktarma tamamlandıktan sonra hedef araçıda gerçekleşecek program çağrısını belirtir. Tek bir `fte: postdst` öğesini belirli bir aktarımla ilişkilendirebilirsiniz. Ek bilgi için [program başlatma](#) başlıklı konuya bakın.

`fte:presrc`, `fte:predst`, `fte:postsrc`, `fte:postdst` ve çıkışlar bir başarı durumu döndürmezse, kurallar belirtilen sırada aşağıdaki gibidir:

1. Kaynak başlatma çıkışlarını çalıştırın. Kaynak başlatma çıkışları başarısız olursa, aktarma başarısız olur ve başka bir işlem çalıştırılmaz.
2. Kaynak öncesi çağrısı çalıştırın (varsa). Kaynak öncesi çağrı başarısız olursa, aktarma başarısız olur ve başka bir şey çalıştırılmaz.
3. Hedef başlatma çıkışlarını çalıştırın. Hedef başlatma çıkışları başarısız olursa, aktarma başarısız olur ve başka bir işlem yürütülmez.
4. Hedef öncesi çağrısı çalıştırın (varsa). Hedef öncesi çağrı başarısız olursa, aktarma başarısız olur ve başka hiçbir şey çalıştırılmaz.
5. Dosya aktarımlarını gerçekleştirin.
6. Hedef uç çıkışlarını çalıştırın. Bu çıkışlar için arıza durumu yok.
7. Aktarma başarılı olursa (bazı dosyalar başarıyla aktarılırsa, aktarma başarılı olarak kabul edilir), hedef sonrası çağrısı (varsa) çalıştırın. Hedef sonrası çağrı başarısız olursa, aktarma başarısız olur.
8. Kaynak uç çıkışlarını çalıştırın. Bu çıkışlar için arıza durumu yok.
9. Aktarma başarılı olursa, kaynak sonrası çağrısı (varsa) çalıştırın. Kaynak sonrası çağrı başarısız olursa aktarma başarısız olur.

Örnekler

Bu örnek, `agent1` ile `agent2` arasında temel bir dosya taşıma işlemi gösterir. Dosya taşıma işlemi başlatma komutu, istemci iletim kipi bağlantısı kullanılarak `qm0`, adlı bir kuyruk yöneticisine gönderilir. Dosya aktarma işleminin sonucu, `move .result` adlı özelliğe atanır.

```
<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>
```

İlgili kavramlar

Kurtarmada dosya aktarımları için zamanaşımı seçeneği

İlgili görevler

[Apache Ant 'yi MFT ile kullanma](#)

fte: ignoreoutcome Ant task

fte:filecopy, **fte:filemove**ya da **fte:call** komutunun sonucunu yoksayın. **defer**sonucunun olması için bir **fte:filecopy**, **fte:filemove**ya da **fte:call** görevi belirttiğinizde, Ant görevi bu sonucu izlemek için kaynakları ayırır. Sonuçla artık ilgilenmiyorsanız, bu kaynakları serbest bırakmak için **fte:ignoreoutcome** görevini kullanabilirsiniz.

Öznitelikler

tanıtıcı

Gereklidir. Artık ilgilenmeyen sonucu tanımlar. Genellikle bu tanıtıcıyı, "[fte: filecopy Ant görevi](#)" sayfa 2072, "[fte: filetaşıma Ant görevi](#)" sayfa 2075ya da "[fte: call Ant görevi](#)" sayfa 2068 görevinin `idproperty` özniteliğini kullanarak ayarladığınız bir özellik kullanarak belirtirsiniz.

Örnek

Bu örnek, önceki "[fte: filecopy Ant görevi](#)" sayfa 2072 görevinin sonucunu izlemek üzere ayrılmış kaynakları serbest bırakmak için `fte: ignoreoutcome` görevini nasıl kullanabileceğinizi gösterir.

```
<!-- 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}"/>
```

İlgili görevler

[Apache Ant 'yi MFT ile kullanma](#)

fte: ping Ant görevi

Bu IBM MQ Managed File Transfer Ant görevi, bir aracıyı bir yanıt almak üzere görevlendirir ve aracının aktarımları işleyip işleyemeyeceğini belirler.

Not: IBM WebSphere MQ File Transfer Edition (FTE) artık desteklenen bir ürün değildir. IBM MQ'da FTE'den Managed File Transfer bileşenine geçiş için [Migrating Managed File Transfer](#) başlıklı konuya bakın.

Öznitelikler

aracı

Gereklidir. **fte:ping** isteğinin gönderileceği aracıyı belirtir. Değer şu biçimdedir: `agentname@qmgrname` ; burada `agentname` , aracının adı ve `qmgrname` , bu aracının doğrudan bağlı olduğu kuyruk yöneticisinin adıdır.

cmdqm

İsteğe Bağlı. İsteğin gönderileceği komut kuyruğu yöneticisi. Bu bilgileri `qmgrname@host@port@channel` biçiminde belirtin; burada:

- `qmgrname` , kuyruk yöneticisinin adıdır

- *host* , kuyruk yöneticisinin çalıştığı sistemin isteğe bağlı anasistem adıdır
- *port* , kuyruk yöneticisinin dinlediği isteğe bağlı kapı numarasıdır
- *channel* , kullanılacak isteğe bağlı SVRCONN kanalıdır

Komut kuyruğu yöneticisine ilişkin *host*, *port* ya da *channel* bilgilerini atlarsanız, `command.properties` dosyasında belirtilen bağlantı bilgileri kullanılır.



Uyarı: Aşağıdakiler için değer belirtilmezse:

- *program* değişkeni, bağ tanımlama kipi kullanılır
- *kapı* değişkeni, 1414 değeri kullanılır
- *kanal* değişkeni, SYSTEM.DEF.SVRCONN kullanılır.

Daha fazla bilgi için bkz. [MFT command.properties dosyası](#) .

Ancak, ortadaki öznitelikleri atlayamazsınız; örneğin, `qmgrname@host@@channel`. Örneğin, `qmgrname@host`, `qmgrname@host@port` ya da `qmgrname@hostport@@channel`.

MFT , @ sınırlayıcısını kullanarak belirtilen özniteliği böler. Bulunan simge sayısına bağlı olarak, ilk simgeyi *qmgrname*, ikincisi *anasistem*, üçüncüsü *kapı* ve son olarak *kanal* olarak alır.

Daha fazla bilgi için bkz. [MFT command.properties dosyası](#).

Hangi `command.properties` dosyasının kullanılacağını belirtmek için

com.ibm.wmqfte.propertySet özelliğini kullanabilirsiniz. Daha fazla bilgi için bkz. [com.ibm.wmqfte.propertySet](#).

`cmdqm` özniteliğini kullanmazsanız, bu özellik ayarlanırsa, görev

varsayılan olarak `com.ibm.wmqfte.ant.commandQueueManager` özelliğini kullanır.

`com.ibm.wmqfte.ant.commandQueueManager` özelliği ayarlanmazsa, `command.properties` dosyasında tanımlı olan varsayılan kuyruk yöneticisine bağlanma girişiminde bulunur.

`com.ibm.wmqfte.ant.commandQueueManager` özelliğinin biçimi, `cmdqm` özniteliğiyle (yani, `qmgrname@host@port@channel`) aynıdır.

rcproperty

Gereklidir. **ping** işleminin dönüş kodunun saklanacağı bir özelliği adlandırıyor.

zaman aşımı

İsteğe Bağlı. Görevin aracının yanıt vermesini beklemesi için saniye cinsinden süre üst sınırı. Zaman aşımı alt sınırı sıfır saniyedir; ancak, komutun aracının yanıt vermesini sonsuza kadar bekleyecek şekilde eksi bir zamanaşımı da belirtilebilir. `timeout` için herhangi bir değer belirtilmezse, varsayılan değer, aracının yanıt vermesi için 5 saniyeye kadar beklemektir.

Örnek

Bu örnek, `qm1` tarafından barındırılan `agent1` ' e bir **fte:ping** isteği gönderir. **fte:ping** isteği, aracının yanıt vermesi için 15 saniye bekler. **fte:ping** isteğinin sonucu, `ping.rc` adlı bir özellikte saklanır.

```
<fte:ping agent="agent1@qm1" rcproperty="ping.rc" timeout="15"/>
```

Dönüş kodları

0

Komut başarıyla tamamlandı.

2

Komut zamanaşımına uğradı.

İlgili görevler

[Apache Ant ' yi MFT ile kullanma](#)

fte: uuid Ant görevi

Sözde rasgele benzersiz bir tanıtıcı oluşturur ve bunu belirli bir özelliğe atar. Örneğin, diğer dosya aktarma işlemleri için iş adları oluşturmak üzere bu tanıtıcıyı kullanabilirsiniz.

Öznitelikler

uzunluk

Gereklidir. Oluşturulacak UUID ' nin sayısal uzunluğu. Bu uzunluk değeri, **prefix** değiştirilmesiyle belirtilen herhangi bir öneğin uzunluğunu içermez.

özellik

Gereklidir. Oluşturulan UUID ' nin atanacağı özelliğin adı.

Önek

İsteğe Bağlı. Oluşturulan UUID ' ye eklenecek bir önek. Bu önek, **length** parametresiyle belirtildiği gibi UUID uzunluğunun bir parçası olarak sayılmaz.

Örnek


Bu örnek, ABC harflerinin ardından 16 sözde rasgele onaltılı karakterle başlayan bir UUID ' yi tanımlar. UUID, `uuid` .property adlı bir özelliğe atanır.


```
<fte:uuid length="16" property="uuid.property" prefix="ABC"/>
```

İlgili görevler

[Apache Ant ' yi MFT ile kullanma](#)

fte: filShakespeare Ant içiçe yerleştirilmiş öge

fte: filespec parametresi, diğer görevlerde içiçe yerleştirilmiş bir öge olarak kullanılır. Bir ya da daha çok kaynak dosya, dizin  ya da veri kümesi ile hedef arasındaki bir eşlemeyi tanımlamak için

fte: filespec komutunu kullanın. Genellikle bu öge, bir dosya ya da dizin kümesini  ya da veri kümelerini taşımak ya da kopyalamak için ifade ederken kullanılır.

İçiçe yerleşimli:

- [fte: filecopy](#) görevi
- [fte: filetaşıma](#) görevi

Kaynak belirtimi öznitelikleri

`srcfilShakespeare` ya da `srcqueue` seçeneğinden birini belirtmelisiniz.

srcfilespec

Dosya işleminin kaynağını belirtir. Bu özniteliğin değeri bir genel arama karakteri içerebilir.

kaynak kuyruğu

Aktarma işleminin kaynağının bir kuyruk olduğunu belirler. Aktarma işlemi, verileri bu özniteliğin belirlediği kuyruksa saklanan iletilerden taşır. **fte: filespec** görevi **fte: filecopy** görevi içinde içiçe yerleştirildiyse bu özniteliği belirtemezsiniz.

Kaynak aracı bir protokol köprüsü aracısı olduğunda `srcqueue` özniteliği desteklenmez.

Hedef belirtimi öznitelikleri

Şunlardan birini belirtmelisiniz: `dstdir`, `dstds`, `dstfilespace`, `dstfile`, `dstqueue` ya da `dstpds`.

dstdir

Bir dosya işleminin hedefi olarak bir dizin belirler.

z/OS **dstds**

Bir dosya işleminin hedefi olarak bir veri kümesini belirtir.

Bu öznitelik yalnızca hedef aracı z/OS platformunda çalışıyorsa desteklenir.

dstfile

Dosya işleminin hedefi olarak bir dosyayı belirtir.

dstfilespace

Bir dosya işleminin hedefi olarak bir dosya alanını belirtir.

Bu öznitelik, hedef aracı web ağ geçidi dosya alanına erişimi olan bir IBM MQ 8.0 web aracısıysa geçerlidir.

z/OS **dstpds**

Bir dosya işleminin hedefi olarak bölümlenmiş bir veri kümesini belirtir.

Bu öznitelik yalnızca hedef aracı z/OS platformunda çalışıyorsa desteklenir.

dstqueue

Dosyadan iletiye işlemin hedefi olarak bir kuyruk belirler. İsteğe bağlı olarak, QUEUE@QUEUEMANAGER. biçimini kullanarak bu belirtim bir kuyruk yöneticisi adı ekleyebilirsiniz. Bir kuyruk yöneticisi adı belirtmezseniz, enableClusterQueueInputÇıkış aracı özelliğini true olarak ayarlamadıysanız, hedef aracı kuyruk yöneticisi kullanılır. enableClusterQueueInputOutput özelliği true olarak ayarlanırsa, hedef aracı kuyruğun nerede bulunduğunu saptamak için standart IBM MQ yordamlarını kullanır. Kuyruk yöneticisinde var olan geçerli bir kuyruk adı belirtmelisiniz.

dstqueue özniteliğini belirtirseniz, bu öznitelikler karşılıklı olarak birbirini dışladığından srcqueue özniteliklerini belirtmezsiniz.

Hedef aracı bir protokol köprüsü aracısı olduğunda dstqueue özniteliği desteklenmez.

Kaynak seçeneği öznitelikleri

kaynak kodlaması

İsteğe Bağlı. Aktarmak için dosya tarafından kullanılan karakter kümesi kodlaması.

Bu özniteliği yalnızca conversion özniteliği text . değerine ayarlandığında belirtebilirsiniz.

srcencoding özniteliğini belirtmezseniz, metin aktarımları için kaynak sistemin karakter kümesi kullanılır.

srceol

İsteğe Bağlı. Aktarılmakta olan dosya tarafından kullanılan satır sonu sınırlayıcısı. Geçerli değerler şunlardır:

- CRLF -Satır sonu sınırlayıcısı olarak satır başı karakteri ve ardından satır besleme karakteri kullanın. Bu kural, Windows sistemleri için tipik bir kuraldır.
- LF -Satır sınırlayıcı olarak bir satır besleme karakteri kullanın. Bu kural, UNIX sistemleri için tipik bir kuraldır.

Bu özniteliği yalnızca conversion özniteliği textdeğerine ayarlandığında belirtebilirsiniz. srceol özniteliğini belirtmezseniz, metin aktarımları kaynak aracının işletim sistemine dayalı olarak doğru değeri otomatik olarak belirler.

z/OS **srckeeptrailingspaces**

İsteğe Bağlı. Metin kipi aktarımının bir parçası olarak, değişmez uzunluklu veri kümesinden okunan kaynak kayıtlarda sondaki boşlukların tutulup tutulmayacağını belirler. Geçerli değerler şunlardır:

- true -sondaki boşluklar korunur.
- false -sondaki boşluklar soyulmuştur.

srckeeptrailingspaces özniteliğini belirtmezseniz, varsayılan değer olan false belirtilir.

Bu özniteliği yalnızca srcfilspec özniteliğini de belirtirseniz ve conversion özniteliğini text . değerine ayarlarsanız belirtebilirsiniz.

srcmsgdelimbytes

İsteğe Bağlı. Bir ikili dosyaya birden çok ileti eklenirken sınırlayıcı olarak eklenecek bir ya da daha çok bayt değerini belirler. Her değer, başına x eklenmiş olarak, 00-FF aralığında iki onaltılı basamak olarak belirlenmelidir. Birden çok byte virgülle ayrılmış olmalıdır. Örneğin, srcmsgdelimbytes="x08,xA4". srcmsgdelimbytes özniteliğini yalnızca srcqueue özniteliğini de belirttiyseniz belirtebilirsiniz. conversion özniteliği için text değerini de belirttiyseniz, srcmsgdelimbytes özniteliğini belirtebilirsiniz.

srcmsgdelimtext

İsteğe Bağlı. Bir metin dosyasına birden çok ileti eklenirken sınırlayıcı olarak eklenecek metin sırasını belirler. Sınırlayıcıya dizgi hazır bilgileri için Java çıkış dizilerini ekleyebilirsiniz. Örneğin, srcmsgdelimtext="\u007d\n". Metin sınırlayıcı, kaynak aracı tarafından her iletiden sonra eklenir. Metin sınırlayıcı, aktarımın kaynak kodlaması kullanılarak ikili biçimde kodlanır. Her ileti ikili biçimde okunur, kodlanmış sınırlayıcı iletinin sonuna ikili biçimde eklenir ve sonuç hedef aracıya ikili biçimde aktarılır. Kaynak aracı kod sayfası çift bayt dizilimi başlangıç ve bitiş durumlarını içeriyorsa, aracı her iletinin sonunda çift bayt dizilimi bitiş durumunda olduğunu varsayar. Hedef aracıda ikili veriler, dosyaya metin aktarımı ile aynı şekilde dönüştürülür. srcmsgdelimtext özniteliğini yalnızca srcqueue özniteliğini ve conversion özniteliği için text değerini belirttiyseniz belirtebilirsiniz.

srcmsgdelimposition

İsteğe Bağlı. Metin ya da ikili sınırlayıcının eklendiği konumu belirtir. Geçerli değerler şunlardır:

- prefix -sınırlayıcılar, her iletideki verilerden önce hedef dosyaya eklenir.
- postfix -sınırlayıcılar, her iletideki verilerden sonra hedef dosyaya eklenir.

srcmsgdelimposition özniteliğini yalnızca, srcmsgdelimbytes ya da srcmsgdelimtext özniteliklerinden birini de belirttiyseniz belirtebilirsiniz.

srcmsggroups

İsteğe Bağlı. İletilerin IBM MQ grup tanıtıcısına göre gruplandırıldığını belirtir. İlk tam grup hedef dosyaya yazılır. Bu öznitelik belirtilmezse, kaynak kuyruktaki tüm iletiler hedef dosyaya yazılır. srcmsggroups özniteliğini yalnızca srcqueue özniteliğini de belirttiyseniz belirtebilirsiniz.

srcqueuetimeout

İsteğe Bağlı. Aşağıdaki koşullardan birinin karşılanması için beklenecek süreyi saniye cinsinden belirtir:

- Kuyruğa yazılacak yeni bir ileti için.
- srcmsggroups özniteliği belirtildiyse, tam bir grubun kuyruğa yazılması için.

Bu koşulların hiçbiri srcqueuetimeoutdeğeriyle belirtilen süre içinde karşılanmazsa, kaynak aracı kuyruktan okumayı durdurur ve aktarımı tamamlar. srcqueuetimeout özniteliği belirtilmezse, kaynak kuyruk boşsa kaynak aracı kaynak kuyruktan okumayı hemen durdurur ya da kuyrukte tam grup yoksa, srcmsggroups özniteliği belirtilirse. srcqueuetimeout özniteliğini yalnızca srcqueue özniteliğini de belirttiyseniz belirtebilirsiniz.

srcqueuetimeout değerini ayarlama hakkında bilgi için bkz. [İletiden dosyaya aktarımda bekleme süresi belirtilmesi için kılavuz](#).

z/OS srcrecdelimbytes

İsteğe Bağlı. Kayıt yönelimli bir kaynak dosyadan ikili dosyaya birden çok kayıt eklenirken, sınırlayıcı olarak eklenecek bir ya da daha çok bayt değerini belirler. Her bir değeri, öneki x olan 00-FF aralığında iki onaltılı basamak olarak belirtmeniz gerekir. Birden çok byte virgülle ayrılmış olmalıdır. Örneğin:

```
srcrecdelimbytes="x08,xA4"
```

srcrecdelimbytes özniteliğini, aktarım kaynak dosyası kayıt odaklı (örneğin, bir z/OS veri kümesi) ve hedef dosya normal, kayıt odaklı olmayan bir dosyaysa belirtebilirsiniz. conversion özniteliği için text değerini de belirttiyseniz, srcrecdelimbytes özniteliğini belirtebilirsiniz.

srcrcdelimpos

İsteğe Bağlı. İkili sınırlayıcının eklendiği konumu belirtir. Geçerli değerler şunlardır:

- **prefix**-Sınırlayıcılar, her kaynak kayıt odaklı dosya kaydındaki verilerden önce hedef dosyaya eklenir.
- **postfix**-sınırlayıcılar, her kaynak kayıt odaklı dosya kaydından sonra hedef dosyaya eklenir.

srcrcdelimpos özniteliğini yalnızca srcrcdelimbytes özniteliğini de belirttiyseniz belirtebilirsiniz.

Hedef seçenek öznitelikleri

stencoding

İsteğe Bağlı. Aktarılan dosya için kullanılacak karakter kümesi kodlaması.

Bu özniteliği yalnızca conversion özniteliği **text** . değerine ayarlandığında belirtebilirsiniz.

dstencoding özniteliği belirtilmezse, metin aktarımları için hedef sistemin karakter kümesi kullanılır.

dsteol

İsteğe Bağlı. Aktarılan dosya için kullanılacak satır sonu sınırlayıcısı. Geçerli değerler şunlardır:

- **CRLF** -Satır sonu sınırlayıcısı olarak satır başı karakteri ve ardından satır besleme karakteri kullanın. Bu kural, Windows sistemleri için tipik bir kuraldır.
- **LF** -Satır sınırlayıcı olarak bir satır besleme karakteri kullanın. Bu kural, UNIX sistemleri için tipik bir kuraldır.

Bu özniteliği yalnızca conversion özniteliği **text** . değerine ayarlandığında belirtebilirsiniz.

dsteol özniteliğini belirtmezseniz, metin aktarımları hedef aracının işletim sistemine dayalı olarak doğru değeri otomatik olarak belirler.

dstmsgdelimbytes

İsteğe Bağlı. İkili bir dosyayı birden çok iletiye bölerken kullanılacak onaltılı sınırlayıcıyı belirler. Tüm iletiler aynı IBM MQ grup tanıtıcısına sahiptir; gruptaki son iletide IBM MQ **LAST_MSG_IN_GROUP** işareti ayarlanır. Sınırlayıcı olarak onaltılı byte 'ı belirtme biçimi şöyledir: **xNN**; burada N, 0-9 ya da a-f aralığındaki bir karakterdir. Onaltılı baytların virgülle ayrılmış listesini belirterek, sınırlayıcı olarak onaltılı baytların sırasını belirtebilirsiniz; örneğin: **x3e , x20 , x20 , xbf**.

dstmsgdelimbytes özniteliğini yalnızca dstqueue özniteliğini belirttiyseniz ve aktarma ikili kipte ise belirtebilirsiniz. dstmsgsize, dstmsgdelimbytesve dstmsgdelimpattern özniteliklerinden yalnızca birini belirtebilirsiniz.

dstmsgdelimpattern

İsteğe Bağlı. Bir metin dosyasını birden çok iletiye bölerken kullanılacak Java düzenli ifadesini belirtir. Tüm iletiler aynı IBM MQ grup tanıtıcısına sahiptir; gruptaki son iletide IBM MQ **LAST_MSG_IN_GROUP** işareti ayarlanır. Bir düzenli ifadeyi sınırlayıcı olarak belirtmeye ilişkin biçim, parantez içine alınmış (*regular_expression*) ya da çift tırnak içine alınmış "*regular_expression*" düzenli bir ifadedir. Daha fazla bilgi için bkz. [MFT tarafından kullanılan düzenli ifadeler](#).

Varsayılan olarak, düzenli ifadenin eşleştirebileceği dizginin uzunluğu hedef aracı tarafından beş karakterle sınırlıdır. **maxDelimiterMatchLength** aracı özelliğini kullanarak bu davranışı değiştirebilirsiniz. Daha fazla bilgi için bkz. [MFT gelişmiş aracı özellikleri](#).

dstmsgdelimpattern özniteliğini yalnızca dstqueue özniteliğini belirttiyseniz ve aktarma metin kipindeyse belirtebilirsiniz. dstmsgsize, dstmsgdelimbytesve dstmsgdelimpattern özniteliklerinden yalnızca birini belirtebilirsiniz.

dstmsgdelimposition

İsteğe Bağlı. Metin ya da ikili sınırlayıcının içinde olması beklenen konumu belirtir. Geçerli değerler şunlardır:

- **prefix** -Sınırlayıcılar her satırın başında beklenir.
- **postfix** -Sınırlayıcılar her satırın sonunda beklenir.

dstmsgdelimposition özniteliğini yalnızca dstmsgdelimpattern özniteliğini de belirttiyseniz belirtebilirsiniz.

dstmsgincludededelim

İsteğe Bağlı. Dosyayı iletilerde birden çok iletiye bölmek için kullanılan sınırlayıcının dahil edilip edilmeyeceğini belirtir. dstmsgincludedelim özniteliği belirtilirse, sınırlayıcı, sınırlayıcıdan önce gelen dosya verilerini içeren iletinin sonuna eklenir. Varsayılan olarak, sınırlayıcı iletilere dahil edilmez. dstmsgincludedelim özniteliğini, dstmsgdelimpattern ve dstmsgdelimbytes özniteliklerinden birini de belirttiyseniz belirtebilirsiniz.

dstmsgpersist

İsteğe Bağlı. Hedef kuyruğa yazılan iletilerin kalıcı olup olmadığını belirtir. Geçerli değerler şunlardır:

- `true` -Kalıcı iletileri hedef kuyruğa yazın. Bu varsayılan değerdir.
- `false` -Kalıcı olmayan iletileri hedef kuyruğa yazın.
- `qdef` -Kalıcı saklama değeri, hedef kuyruğun DefPersistence (DefPersistence) özniteliğinden alınır.

Bu özniteliği yalnızca dstqueue özniteliği de belirtildiğinde belirtebilirsiniz.

dstmsgprops

İsteğe Bağlı. Aktarma işlemi tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmadığını belirler. Olası değerler şunlardır:

- `true` -Aktarma tarafından yaratılan ilk iletide ileti özelliklerini ayarlayın.
- `false` -Aktarma tarafından yaratılan ilk iletide ileti özelliklerini ayarlamayın. Bu varsayılan değerdir.

Daha fazla bilgi için, hedef kuyruklara yazılan iletilerde MFT tarafından ayarlanan [MQ ileti özelliklerine bakın](#).

Bu özniteliği yalnızca dstqueue özniteliği de belirtildiğinde belirtebilirsiniz.

dstmsgsize

İsteğe Bağlı. Dosyanın birden çok sabit uzunluklu iletiye bölünip bölünmeyeceğini belirler. Tüm iletiler aynı IBM MQ grup tanıtıcısına sahiptir; gruptaki son iletide IBM MQ LAST_MSG_IN_GROUP işareti ayarlanır. İletilerin büyüklüğü dstmsgsize değeriyle belirtilir. dstmsgsize biçimi *uzunlukbirimleridir*; burada *uzunluk* artı bir tamsayı değeridir ve *birimler* aşağıdaki değerlerden biridir:

- `B` -Byte. İzin verilen alt sınır değeri, hedef iletilerin kod sayfasının karakter başına byte sayısı üst sınırının iki katıdır.
- `K` -Kibibayt. Bu 1024 bayta eşdeğerdir.
- `M` -Mebibayt. Bu, 1024 kibibayta eşdeğerdir.

Ktk metin kipinde aktarlrsa ve çift baytlk karakter takm ya da çok baytlk karakter takmndaysa, ktk, belirlenen ileti bykl ş ne en yakın karakter snrındaki iletilere ayrılır.

dstmsgsize özniteliğini yalnızca dstqueue özniteliğini de belirttiyseniz belirtebilirsiniz. dstmsgsize, dstmsgdelimbytesve dstmsgdelimpattern özniteliklerinden yalnızca birini belirtebilirsiniz.

dstunsupportedcodepage

İsteğe Bağlı. Hedef kuyruk yöneticisi, dstqueue özniteliğinde belirtildiği gibi, dosya verilerini bir kuyruğa metin aktarımı olarak aktarırken kullanılan kod sayfasını desteklemiyorsa yapılacak işlemi belirler. Bu özniteliğe ilişkin geçerli değerler şunlardır:

- `binary` -aktarmaya devam edin, ancak aktarılmakta olan verilere kod sayfası dönüştürmesi uygulamayın. Bu değer belirtilmesi, dönüştürme özniteliğinin textolarak ayarlanmamasına eşdeğerdir.
- `fail` -aktarma işlemine devam etme. Dosya aktarılamadı olarak kaydedildi. Bu varsayılandır.

Yalnızca, dstqueue özniteliğini ve conversion özniteliği için text değerini belirttiyseniz, dstunsupportedcodepage özniteliğini belirtebilirsiniz.

dsttruncaterecords

İsteğe Bağlı. LRECL veri kümesi özniteliğinden daha uzun hedef kayıtların kesildiğini belirtir. True olarak ayarlanırsa kayıtlar kesilir. False olarak ayarlanırsa kayıtlar kaydırılır. Varsayılan ayar false değeridir. Bu parametre yalnızca hedefin bir veri kümesi olduğu metin kipi aktarımları için geçerlidir.

Diğer öznitelikler

sağlama toplamı

İsteğe Bağlı. Aktarılan dosyaların toplamını sağlamak için kullanılan algoritmayı belirler.

- MD5 - MD5 hash algoritmasını kullanın.
- NONE -sağlama toplamı algoritması kullanmayın.

checksum özniteliğini belirtmezseniz, varsayılan değer olan MD5 kullanılır.

dönüştürme

İsteğe Bağlı. Aktarmakta olan ktk için uygulanacak d " nYtrme tipini belirler. Olası değerler şunlardır:

- binary -dönüştürme uygulanmaz.
- text -kaynak ve hedef sistemler arasında kod sayfası dönüştürmesi uygulayın. Satır sınırlayıcıların dönüştürülmesini de uygulayın. srcencoding, dstencoding, srceol ve dsteol öznitelikleri, uygulanan dönüştürmeyi etkiler.

conversion özniteliğini belirtmezseniz, varsayılan değer olan binary belirtilir.

Üzerine Yaz

İsteğe Bağlı. İşlem, var olan bir hedef dosyanın **z/OS** ya da veri kümesinin üzerine

yazılıp yazılamayacağını belirler. truedeğerini belirttiğinizde, var olan hedef dosyaların **z/OS** ya da veri kümelerinin üzerine yazılır. falsedeğerini belirttiğinizde, hedefte yinelenen bir

dosya **z/OS** ya da veri kümesinin varlığı işlemin başarısız olmasına neden olur. overwrite özniteliği belirtilmezse, varsayılan bir false değeri belirtilir.

özyineli

İsteğe Bağlı. Dosya aktarımının alt dizinlere yinelenip yinelenmeyeceğini belirler. truedeğerini belirlediğinizde, aktarma işlemi alt dizinlere yinelenir. falsedeğerini belirlediğinizde, aktarma işlemi alt dizinlere yinelenmez. recurse özniteliği belirtilmezse, varsayılan bir false değeri belirtilir.

Örnek

Bu örnek, kaynak dosyası file1.bin ve hedef dosyası file2.bin . olan bir fte: filespec dosyasını belirtir.

```
<fte:filespec srcfilespec="/home/fteuser/file1.bin" dstfile="/home/fteuser/file2.bin"/>
```

İlgili görevler

[Apache Ant 'yi MFT ile kullanma](#)

fte: meta veri Ant içiçe yerleştirilmiş öğeler

Meta veriler, bir dosya aktarma işlemiyle ek kullanıcı tanımlı bilgileri taşımak için kullanılır.

Managed File Transfer 'IN meta verileri nasıl kullandığıyla ilgili daha fazla bilgi için bkz. [“MFT kullanıcı çıkışları için meta veriler” sayfa 2090](#) .

İç içe yerleşimli:

- [fte: filecopy görevi](#)
- [fte: filetaşıma görevi](#)
- [fte: call görevi](#)

İç içe geçmiş öğeler olarak belirtilen parametreler

fte: giriş

fte:metadata içiçe yerleştirilmiş öğesinin içinde en az bir giriş belirtmeniz gerekir. Birden çok girdi belirtmeyi seçebilirsiniz. Girdiler bir anahtar adını bir değerle ilişkilendirir. Anahtarlar fte:metadata bloğunda benzersiz olmalıdır

Giriş öznitelikleri

NAME

Gereklidir. Bu girdiye ait anahtarın adı. Bu ad, bir fte: metadata öğesinin içine yerleştirilmiş tüm **entry** değıştirmelerinde benzersiz olmalıdır.

değer

Gereklidir. Bu girdiye atanacak değer.

Örnek

Bu örnek, iki giriş içeren bir fte:metadata tanımlamasını gösterir.

```
<fte:metadata>
  <fte:entry name="org.foo.partColor" value="red"/>
  <fte:entry name="org.foo.partSize" value="medium"/>
</fte:metadata>
```

İlgili görevler

[Apache Ant 'yi MFT ile kullanma](#)

Program çağırma iç içe geçmiş öğeleri

Programlar, iç içe yerleştirilmiş beş öğeden biri kullanılarak başlatılabilir: **fte:presrc**, **fte:predst**, **fte:postdst**, **fte:postsrvcve** **fte:command**. Bu iç içe geçmiş öğeler, bir aracıya bir dış programı işleminin bir parçası olarak çağırmasını bildirir. Bir programı başlatmadan önce, komutun, komutu çalıştıran aracının `agent.properties` dosyasında `commandPath` özelliği tarafından belirtilen bir konumda olduğundan emin olmanız gerekir.

Her program çağırma öğesinin farklı bir adı olsa da, aynı öznitelik kümesini ve aynı iç içe öğe kümesini paylaşırlar. Programlar **fte:filecopy**, **fte:filemove** ve **fte:command** Ant görevleri tarafından başlatılabilir.

Bir Connect:Direct köprü aracısından programları çağırabilirsiniz.

Programları çağırabilen Ant görevleri:

- **fte:filecopy** görevi, **fte:predst**, **fte:postdst**, **fte:presrvcve** **fte:postsrvc** iç içe geçmiş öğelerini kullanarak program çağırma parametrelerini iç içe yerleştirir.
- **fte:filetaşım**a görevi, **fte:predst**, **fte:postdst**, **fte:presrvcve** **fte:postsrvc** iç içe geçmiş öğelerini kullanarak program çağırma parametrelerini iç içe yerleştirir.
- **fte:call** görevi, **fte:command** iç içe yerleşimli öğesini kullanarak program çağırma değıştirmelerini iç içe yerleştirir.

Öznitelikler

DELETE ...

Gereklidir. Çağrılmaları gereken programı adlandırıyor. Aracının bir komutu çalıştırabilmesi için, komutun aracının `agent.properties` dosyasındaki `commandPath` özelliği tarafından belirtilen bir konumda olması gerekir. Daha fazla bilgi için bkz. [commandPath MFT property](#). `command` özniteliğinde belirtilen yol bilgileri, `commandPath` özelliği tarafından belirtilen bir konumla görelilik olarak kabul edilir. `type executable` olduğunda, yürütülebilir bir program beklenir; tersi durumda, çağrı tipine uygun bir komut dosyası beklenir.

yeniden sayım

İsteğe Bağlı. Program başarılı bir dönüş kodu döndürmezse, programı yeniden çağırma sayısı. `command` özniteliği tarafından adlandırılan program bu sayıya kadar çağrılır. Bu öznitelikğe atanan değer negatif olmamalıdır. `retrycount` özniteliğini belirtmezseniz, varsayılan değer olan sıfır kullanılır.

yeniden denemeyi bekleme

İsteğe Bağlı. Program çağrısını yeniden denemeden önce beklenecek süre (saniye olarak). `command` özneliği tarafından adlandırılan program bir başarı dönüş kodu döndürmezse ve `retrycount` özneliği sıfır dışında bir değer belirtirse, bu parametre yeniden denemeler arasında beklenecek süreyi belirler. Bu özneliğe atanan değer negatif olmamalıdır. `retrywait` özneliğini belirtmezseniz, varsayılan değer olan sıfır kullanılır.

başarılı

İsteğe Bağlı. Bu özneliğin değeri, program çağrısının başarılı bir şekilde ne zaman çalışacağını saptamak için kullanılır. Komuta ilişkin işlem dönüş kodu bu ifade kullanılarak değerlendirilir. Değer, Boole ORdeğerini belirtmek için dikey çubuk karakteriyle (|) birleştirilen bir ya da daha çok ifadeden ya da ve (&) işaretinden oluşabilir. Boole değerini belirtmek için ANDkarakterini kullanın. Her ifade aşağıdaki ifade tiplerinden biri olabilir:

- Süreç dönüş kodu ile sayı arasında bir eşitlik testini gösteren sayı.
- Sayı ile işlem dönüş kodu arasında büyük bir testi belirtmek için başına ">" karakteri eklenmiş bir sayı.
- Sayı ile işlem dönüş kodu arasında daha küçük bir testi belirtmek için başına "<" karakteri eklenmiş bir sayı.
- "!" öneklili bir sayı Sayı ile işlem dönüş kodu arasında eşit olmayan bir sınamayı gösteren karakter.

Örneğin: `>2&<7&!5|0|14` , şu dönüş kodlarının başarılı olduğu şeklinde yorumlanır: 0, 3, 4, 6, 14. Diğer tüm dönüş kodları başarısız olarak yorumlanır. `successrc` özneliğini belirtmezseniz, varsayılan değer olan sıfır kullanılır. Bu, komutun ancak ve ancak sıfır kodu döndürürse başarılı bir şekilde çalıştırılmış olduğu anlamına gelir.

tip

İsteğe Bağlı. Bu özneliğin değeri, hangi program tipinin çağrıldığını belirtir. Aşağıdaki seçeneklerden birini belirleyin:

Yürütülebilir

Görev yürütülebilir bir programı çağırır. `arg` içiçe yerleşimli öğesi kullanılarak ek bağımsız değişkenler belirtilebilir. Programın `commandPath` üzerinde erişilebilir olması ve uygulanabildiği durumlarda yürütme izni kümesi olması beklenir. UNIX komut dosyaları, bir kabuk programı belirttikleri sürece çağrılabilir (örneğin, kabuk komut dosyasının ilk satırı: `#!/bin/sh`). `stderr` ya da `stdout` komutuna yazılan komut çıkışı, çağrı için Managed File Transfer günlüğüne gönderilir. Ancak, veri çıkışı miktarı aracı yapılandırmasıyla sınırlıdır. Varsayılan değer 10K bayttır, ancak aracı özelliğini kullanarak bu varsayılan değeri geçersiz kılabilirsiniz: `maxCommandOutput`.

antscript

Görev, **fteAnt** komutunu kullanarak belirtilen Ant komut dosyasını çalıştırır. Özellikler, özellik içiçe yerleştirilmiş öğesi kullanılarak belirtilebilir. Ant hedefleri, hedef içiçe yerleştirilmiş öğesi kullanılarak belirtilebilir. Ant komut dosyasının `commandPath`komutunda erişilebilir olması beklenir. `stderr` ya da `stdout` 'a yazılan Ant çıkışı, çağrı için Managed File Transfer günlüğüne gönderilir. Ancak, veri çıkışı miktarı aracı yapılandırmasıyla sınırlıdır. Varsayılan değer 10K baytlık verilerdir, ancak aracı özelliğini kullanarak bu varsayılan değeri geçersiz kılabilirsiniz: `maxCommandOutput`.

z/OS jcl

`jcl` değeri yalnızca z/OS üzerinde desteklenir ve belirtilen z/OS JCL komut dosyasını çalıştırır. JCL iş olarak sunulur ve iş kartının var olmasını gerektirir. İş başarıyla sunulduğunda, Managed File Transfer günlüğüne yazılan JCL komutu çıkışı şu metni içerir: `JOB job_name(job_id)`, burada:

- `iş_adi` , JCL ' de iş kartı tarafından tanımlanan işin adıdır.
- `job_id` , z/OS sistemi tarafından oluşturulan iş tanıtıcısıdır.

İş başarıyla sunulamazsa, JCL komut dosyası komutu başarısız olur ve günlüğe hatanın nedenini belirten bir ileti yazar (örneğin, iş kartı yoktur). İşin başarıyla çalıştırılıp çalıştırılmadığını ya da tamamlandığını anlamak için SDSFgibi bir sistem hizmeti kullanın. Managed File Transfer , yalnızca işi sunduğundan bilgileri sağlamaz; sistem daha sonra işin ne zaman yürütüleceğini ve iş çıkışının nasıl sunulacağını belirler. Bir JCL komut dosyası toplu iş olarak sunulduğundan, yalnızca işin

başarıyla gönderildiğini ve aktarma başlamadan önce başarıyla tamamlanıp tamamlanmadığını bilmediğinizi bildiğinizden, presrc ya da predst içiçe yerleştirilmiş bir öge için jcl belirtilmesi önerilmez. jcl tipiyle geçerli içiçe yerleştirilmiş öge yok.

Aşağıdaki örnekte bir JCL işi gösterilmektedir:

```
//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
```

İç içe geçmiş öğeler olarak belirtilen parametreler

fte: arg

Yalnızca type özniteliğinin değeri yürütülürise geçerlidir. Program çağrısının bir parçası olarak çağrılmakta olan programa ilişkin bağımsız değişkenleri belirtmek için iç içe geçmiş fte: arg öğelerini kullanın. Program bağımsız değişkenleri, fte: arg öğeleri tarafından belirtilen değerlerden, fte: arg öğeleriyle karşılaşılan sırayla oluşturulur. Bir program çağrısının iç içe geçmiş öğeleri olarak sıfır ya da daha fazla fte: arg öğesi belirtmeyi seçebilirsiniz.

fte: özellik

Yalnızca type özniteliğinin değeri antscriptise geçerlidir. Ad-değer çiftlerini Ant komut dosyasına geçirmek için içiçe yerleşimli fte: property öğelerinin name ve value özniteliklerini kullanın. Bir program çağrısının iç içe geçmiş öğeleri olarak sıfır ya da daha fazla fte: property öğesi belirtmeyi seçebilirsiniz.

fte: hedef

Yalnızca type özniteliğinin değeri antscriptise geçerlidir. Ant komut dosyasında çağrılacak bir hedef belirtin. Bir program çağrısının iç içe geçmiş öğeleri olarak sıfır ya da daha fazla fte: target öğesi belirtmeyi seçebilirsiniz.

Arg öznitelikleri

değer

Gereklidir. Çağrılmakta olan programa geçirilecek bağımsız değişkenin değeri.

Özellik öznitelikleri

Ad

Gereklidir. Ant komut dosyasına geçirilecek bir özelliğin adı.

değer

Gereklidir. Ant komut dosyasına geçirilen özellik adıyla ilişkilendirilecek değer.

Örnekler

Bu örnek, bir fte: filecopy görevinin parçası olarak belirtilen fte: postsrc program çağrısını gösterir. Program çağırma, post.sh adlı bir program içindir ve /home/fteuser2/file.bin. bağımsız değişkeninin tek bir bağımsız değişkeni sağlar.

```
<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" successrc="1" >
  <fte:arg value="/home/fteuser2/file.bin"/>
</fte:postsrc>
</fte:filecopy>
```

Bu örnek, bir `fte:call` görevinin bir parçası olarak belirtilmekte olan bir `fte:command` program çağrısını gösterir. Program çağrısı, herhangi bir komut satırı bağımsız değişkeni iletilmeyen `command.sh` adlı yürütülür dosya içindir. `command.sh`, 1 başarı dönüş kodu döndürmezse, komut 30 saniye sonra yeniden denenecek.

```
<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>
```

Bu örnek, bir `fte:call` görevinin bir parçası olarak belirtilmekte olan bir `fte:command` program çağrısını gösterir. Program çağrısı, iki özellik geçirilen `script.xml` adlı bir Ant komut dosyasındaki kopyalama ve sıkıştırma hedefleri içindir.

```
<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>
```

İlgili görevler

[MFT ile çalıştırılacak programları belirtme](#)

[Apache Ant 'yi MFT ile kullanma](#)

Özelleştirme başvurusu için MFT kullanıcı çıktıları

Managed File Transfer için kullanıcı çıktıları yapılandırmanıza yardımcı olacak başvuru bilgileri.

İlgili kavramlar

[MFT kaynak ve hedef kullanıcı çıktıları](#)

MFT kullanıcı çıktıları için meta veriler

Managed File Transfer için kullanıcı çıkış yordamlarına sağlanabilecek üç farklı meta veri tipi vardır: ortam, aktarma ve dosya meta verileri. Bu meta veriler, Java anahtar-değer çiftlerinin eşlemeleri olarak sunulur.

Ortam meta verileri

Ortam meta verileri tüm kullanıcı çıkışı yordamlarına geçirilir ve kullanıcı çıkışı yordamının çağrılacağı aracı yürütme ortamını açıklar. Bu meta veriler salt okunur ve herhangi bir kullanıcı çıkışı yordamı tarafından güncellenemez.

Çizelge 882. Ortam meta verileri	
Kilit	Açıklama
AGENT_CONFIGURATION_DIRECTORY_KEY	Aracının yapılandırma bilgilerini içeren dizinin adı.
AGENT_PRODUCT_DIRECTORY_KEY	Aracı kodunun kurulduğu dizinin adı.
AGENT_VERSION_ANAHTARI	Çıkış yordamını çağırarak aracı yürütme ortamının sürüm numarası.

Çizelge 1 'de verilen anahtar adları ve değer adları, EnvironmentMetaDataConstants arabiriminde tanımlanan sabitlerdir.

Meta verileri aktar

Aktarma meta verileri tüm kullanıcı çıkış yordamlarına aktarılır. Meta veriler, sistem tarafından sağlanan değerler ve kullanıcı tarafından sağlanan değerlerden oluşur. Sistem tarafından sağlanan değerleri değiştirirseniz, bu değişiklikler dikkate alınmaz. Kaynak aktarma başlangıç kullanıcı çıkışına ilişkin kullanıcı tarafından sağlanan ilk değerler, aktarımı tanımladığınızda belirttiğiniz değerlere dayalıdır. Kaynak aracı, kaynak aktarma başlangıcı kullanıcı çıkışının işlenmesinin bir parçası olarak kullanıcı tarafından sağlanan değerleri değiştirebilir. Bu kullanıcı çıkışı, tüm dosya aktarımı başlamadan önce çağrılır. Bu değişiklikler, söz konusu aktarıma ilişkin diğer çıkış yordamlarına yapılan sonraki çağrılarda kullanılır. Aktarma meta verileri tüm aktarıma uygulanır.

Tüm kullanıcı çıkışları aktarma meta verilerinden değerleri okuyabilse de, yalnızca kaynak aktarma başlangıcı kullanıcı çıkışı aktarma meta verilerini değiştirebilir

Farklı dosya aktarımları arasında bilgi yaymak için aktarma meta verilerini kullanamazsınız.

Sistem tarafından sağlanan aktarma meta verileri Tablo 2 'de ayrıntılı olarak açıklanmıştır:

Çizelge 883. Meta verileri aktar	
Kilit	Açıklama
DESTINATION_AGENT_KEY	Aktarma hedefi olan aracının adı.
JOB_AD_ANAHTARI	Aktarma isteğiyle ilişkili iş adı
MQMD_USER_KEY	Aktarma isteğini sunmak için kullanılan iletideki MQMD kullanıcı alanı
KAYNAK ANAHTARI_ANAHTARI	Aktarma isteğinde kaynak anasistem adı olarak belirlenen anasistem adı
KAYNAK KULLANICI_ANAHTARI	Aktarma isteğinde kaynak kullanıcı kimliği olarak belirlenen kullanıcı adı
KAYNAK_AGENT_ANAHTARI	Aktarımın kaynağı olan aracının adı
TRANSFER_ID_KEY	Aktarma tanıtıcısı

Çizelge 2 'de verilen anahtar adları ve değer adları, TransferMetaDataConstants arabiriminde tanımlanan sabitlerdir.

Dosya meta verileri

Dosya meta verileri, dosya belirtiminin bir parçası olarak kaynak aktarma başlangıç çıkışına geçirilir. Kaynak ve hedef dosyalar için ayrı dosya meta verileri vardır.

Farklı dosya aktarımları arasında bilgi yaymak için dosya meta verilerini kullanamazsınız.

Çizelge 884. Dosya meta verileri		
Kilit	İzin verilen değerler	Açıklama
CONVERT_LINE_SEPARATORS		Metin aktarımları için kullanılan anahtar değeri, kaynak verilerdeki CRLF (satır başı satır besleme) ya da LF (satır besleme) satır ayırıcı sıralarının hedefteki satır ayırıcı sırasına dönüştürülüp dönüştürülmediğini gösterir.

Çizelge 884. Dosya meta verileri (devamı var)

Kilit	İzin verilen değerler	Açıklama
DELIMITER_KEY.		Kayıt odaklı verileri normal dosyalara aktarırken kayıt verilerini ayırmak için bir sınırlayıcı tanımlamak için kullanılan anahtar değeri. İletiler arası ve dosyalar arası aktarımlar için de kullanılır.
DELIMITER_POSITION_KEY	DELIMITER_POSITION_PREFIX_VALUE DELIMITER_POSITION_POSTFIX_VALUE	Sınırlayıcının konumunu (önek ya da art takı) tanımlamak için DELIMITER_KEY ile birlikte kullanın.
DELIMITER_TYPE_KEY	DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE DELIMITER_TYPE_SIZE_VALUE	Sınırlayıcı tipini tanımlamak için DELIMITER_KEY ile birlikte kullanın.
DESTINATION_EXIST_KEY	DESTINATION_EXIST_KEY_ERROR_VALUE DESTINATION_EXIST_KEY_OVERWRITE_VALUE	Hedef dosya varsa, dosya aktarma davranışını belirler.
DOSYA_ALIAS_ANAHTARI_ANAHTAR İ		Aktarılmakta olan dosyaya ilişkin bir diğer ad tanımlamak için kullanılan anahtar değeri.
FILE_CHECKSUM_METHOD_KEY	FILE_CHECKSUM_METHOD_NONE_VALUE FILE_CHECKSUM_METHOD_MD5_VALUE	Dosya aktarılırken kullanılacak sağlama toplamı yöntemini belirler.
FILE_CONVERSION_ANAHTAR İ	FILE_CONVERSION_TEXT_VALUE FILE_CONVERSION_BINARY_VALUE	Dosya içeriğine uygulanan dönüştürme tipini belirler.
DOSYA_KODLAMASI_ANAHTAR İ		Bir metin dosyası için kullanılan kodlamayı belirler.
FILE_END_OF_LINE_KEY	DOSYA_SON_OF_LINE_LF_VALUE dosya_son_of_line_crlf_value	Bir satırın sonunu gösteren karakter sırasını belirler: < LF > ya da < CR> < LF>.
DOSYA_BOŞLUĞI_DİĞER AD İ		Dosya alanındaki bir dosyanın diğer adını belirler. Not: Bu meta veriler yalnızca FILE_TYPE_KEY değeri FILE_TYPE_FILE_SPACE_VALUE ise kullanılabilir
DOSYA_ALANI_AD İ		Dosya alanının adını belirler. Not: Bu meta veriler yalnızca FILE_TYPE_KEY değeri FILE_TYPE_FILE_SPACE_VALUE ise kullanılabilir
DOSYA_TIPI_ANAHTAR İ	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	Hedef dosya, kuyruk ya da dosya alanı belirtimini belirler.
GROUP_ID_KEY		Kaynak kuyruktan okunacak ileti grubunu belirlemek için iletiden dosyaya aktarımlar için kullanılan anahtar değeri. Bu öznelik yalnızca USE_GROUPS_KEY değeri USE_GROUPS_TRUE_VALUE olduğunda geçerlidir.

Çizelge 884. Dosya meta verileri (devamı var)

Kilit	İzin verilen değerler	Açıklama
INCLUDE_DELIMITER_IN_MESSAGE_KEY	INCLUDE_DELIMITER_IN_MESSAGE_TRUE_VALUE INCLUDE_DELIMITER_IN_MESSAGE_FALSE_VALUE	İletilerin sonunda dosyayı birden çok iletiye bölmek için kullanılan sınırlayıcıların dahil edilip edilmeyeceğini belirlemek üzere dosya iletilerinde kullanılan anahtar değeri. Bu öznelik yalnızca, DELIMITER_TYPE_KEY değeri DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE olduğunda geçerlidir.
INSERT_RECORD_LINE_SEPARATOR_KEY		Satır ayırıcıların her kayıttan sonra verilere eklenip eklenmeyeceğini belirlemek için kayıt odaklı dosyalardan metin aktarımı için kullanılan anahtar değeri.
KEEP_TRAILING_SPACES_KEY	KEEP_TRAILING_SPACES_TRUE_VALUE KEEP_TRAILING_SPACES_FALSE_VALUE	Değişmez uzunluklu biçim veri kümelerinden okunan kayıtlardan sondaki boşlukların kaldırılıp kaldırılmayacağını belirlemek için kullanılan anahtar değeri.
NEW_RECORD_ON_LINE_SEPARATOR_KEY		Verilerdeki satır ayırıcıların kayıt verileriyle birlikte içerilip içerilmeyeceğini ya da yeni bir kayda neden olup olmayacağını belirlemek üzere kayıt odaklı dosyalara metin aktarımları için kullanılan anahtar değeri.
PERSISTENT_KEY	PERSISTENT_TRUE_VALUE PERSISTENT_FALSE_VALUE PERSISTENT_QDEF_VALUE	İletilerin kalıcı olup olmadığını belirlemek için dosyadan iletiye aktarımlar için kullanılan anahtar değeri.
SET_MQ_PROPS_KEY	SET_MQ_PROPS_TRUE_DEĞERİ SET_MQ_PROPS_FALSE_VALUE	Bir dosyadaki ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmadığını ve bir hata oluştuğunda kuyruğa yazılan iletilerin olup olmadığını belirlemek üzere dosyadan iletiye aktarımlar için kullanılan anahtar değeri.
TANINMIYOR_kod_sayfası_anahtarı	TANINMIYOR_CODE_PAGE_FAIL_VALUE UNTANSED_CODE_PAGE_BINARY_VALUE	Verilerin kod sayfası hedef kuyruk yöneticisi tarafından tanınmıyorsa, bir metin kipi aktarımının başarısız olup olmadığını ya da dönüştürmenin gerçekleştirilip gerçekleştirilmediğini belirlemek için dosyadan iletiye aktarımlar için kullanılan anahtar değeri.
USE_GROUPS_KEY	USE_GROUPS_TRUE_VALUE USE_GROUPS_FALSE_VALUE	Kaynak kuyruktan yalnızca tam bir ileti grubunun aktarılıp aktarılmayacağını belirlemek için iletiden dosyaya aktarımlar için kullanılan anahtar değeri.

Çizelge 884. Dosya meta verileri (devamı var)		
Kilit	İzin verilen değerler	Açıklama
BEKLE_SAAAT_ANAHTARI		<p>Kaynak aracısının aşağıdaki durumlardan birini beklemesi için saniye cinsinden süreyi belirlemek üzere iletiden dosyaya aktarımlar için kullanılan anahtar değeri:</p> <ul style="list-style-type: none"> Kuyruk boşsa ya da boşsa, USE_GROUPS_KEY değeri FALSE ise, kaynak kuyrukta görüntülenecek bir ileti. USE_GROUPS_KEY değeri TRUE ise, kaynak kuyrukta görüntülenecek tam bir grup.

Çizelge 3 'te verilen anahtar adları ve değer adları, FileMetaDataConstants arabiriminde tanımlanan sabitlerdir.

İlgili kavramlar

“MFT kullanıcı çıkışları için Java arabirimleri” sayfa 2100

Kullanıcı çıkışı yordamlarına ilişkin Java arabirimlerine ilişkin başvuru bilgileri için bu bölümdeki konuları kullanın.

İlgili görevler

MFT ürününü kullanıcı çıkışlarıyla özelleştirme

İlgili başvurular

“MFT kaynak izleyicisi kullanıcı çıkışları” sayfa 2094

Kaynak izleyicisi kullanıcı çıkışları, ilişkili görev başlatılmadan önce, bir izleme programının tetikleyici koşulu karşılandığında çalışacak şekilde özel kod yapılandırmanızı sağlar.

“MFT Kullanıcı çıkışları için aracı özellikleri” sayfa 2098

agent.properties dosyasındaki standart özelliklere ek olarak, özellikle kullanıcı çıkışı yordamları için birkaç gelişmiş özellik vardır. Bu özelliklerden herhangi birini kullanmak istiyorsanız, agent.properties dosyasını el ile düzenlemeniz gerekir. Aracı çalışırken agent.properties dosyasında bir değişiklik yaparsanız, değişiklikleri almak için aracıyı durdurun ve yeniden başlatın.

MFT kaynak izleyicisi kullanıcı çıkışları

Kaynak izleyicisi kullanıcı çıkışları, ilişkili görev başlatılmadan önce, bir izleme programının tetikleyici koşulu karşılandığında çalışacak şekilde özel kod yapılandırmanızı sağlar.

Doğrudan kullanıcı çıkış kodundan yeni aktarımların çağırılması önerilmez. Bazı durumlarda, kullanıcı çıkışları aracıyı yeniden başlatmaya dayanıklı olmadığı için bu, dosyaların birden çok kez aktarılmasına neden olur.

Kaynak izleyicisi kullanıcı çıkışları, kullanıcı çıkışları için var olan altyapıyı kullanır. İzleme programı kullanıcı çıkışları, bir izleme programı tetiklendikten sonra, ancak ilgili görev izleme programı görevi tarafından çalıştırılmadan önce çağırılır. Bu, kullanıcı çıkışının çalıştırılacak görevi değiştirmesini ve bir görevin devam edip etmeyeceğine karar vermesini sağlar. İzleyici görevini, özgün izleme programının yaratılmasıyla yaratılan görev belgesinde değişken yerine koyma değeri için kullanılan izleyici meta verilerini güncelleyerek değiştirebilirsiniz. Diğer bir seçenek olarak, izleyici çıkışı değiştirge olarak geçirilen görev tanımlaması XML dizgisini değiştirebilir ya da güncelleyebilir. İzleyici çıkışı, görev için 'devam et' ya da 'iptal' sonuç kodunu döndürebilir. İptal döndürülürse, görev başlatılır ve izlenen kaynak tetikleyici koşullarıyla eşleşinceye kadar izleme programı yeniden başlatılamaz. Kaynak değişmediyse, tetikleyici başlatılamaz. Diğer kullanıcı çıkışları gibi, monitör çıkışlarını zincirleme olarak izleyebilirsiniz. Çıkışlardan biri iptal sonuç kodu döndürürse, genel sonuç iptal edilir ve görev başlatılmaz.

- Ortam meta verilerinin eşlemi (diğer kullanıcı çıkışlarıyla aynı)

- Sabit sistem meta verilerini ve deęişebilir kullanıcı meta verilerini içeren bir izleyici meta verileri eşlemi. Sabit sistem meta verileri aşağıdaki gibidir:
 - FILENAME-tetikleyici koşulunu karşılayan dosyanın adı
 - FILEPATH-tetikleyici koşulunu yerine getirenin dosya yolu
 - FILESIZE (bayt cinsinden-bu meta veri var olmayabilir)-tetikleyici koşulunu karşılayan dosyanın boyutu
 - LASTMODIFIEDDATE (Local)-tetikleyici koşulunu yerine getirenin son deęiştirildięi tarih. Bu tarih, aracının çalıştığı saat diliminin yerel tarihi olarak ifade edilir ve ISO 8601 tarihi olarak biçimlendirilir.
 - LASTMODIFIEDTIME (Local)-tetikleyici koşulunu karşılayan dosyanın en son deęiştirildięi yerel biçimde saat. Bu saat, aracının çalıştığı saat diliminin yerel saati olarak ifade edilir ve ISO 8601 saati olarak biçimlendirilir.
 - LASTMODIFIEDDATEUTC-tetikleyici koşulunu karşılayan dosyanın son olarak deęiştirildięi evrensel biçimde tarih. Bu tarih, UTC saat dilimine dönüştürülen yerel tarih olarak ifade edilir ve ISO 8601 tarihi olarak biçimlendirilir.
 - LASTMODIFIEDTIMEUTC-evrensel biçimde, tetikleyici koşulunu karşılayan dosyanın en son deęiştirildięi zaman. Bu saat, UTC saat dilimine dönüştürülen yerel saat olarak ifade edilir ve ISO 8601 saati olarak biçimlendirilir.
 - AGENTNAME-izleyici aracı adı
- İzleyici tetikleyicisinin sonucu olarak çalıştırılacak görevi gösteren bir XML dizgisi.

İzleme programı çıkışları aşağıdaki verileri döndürür:

- İlerleme durumunu belirleyen bir gösterge (devam et ya da iptal et)
- Tetikleyici tarafından karşılanmış günlük iletisine eklenecek dizgi

İzleyici çıkış kodunun çalıştırılması sonucunda, başlangıçta deęiştirge olarak geçirilen izleme meta verileri ve görev tanımlaması XML dizgisi de güncellenmiş olabilir.

monitorExit aracı özelliğinin deęeri (agent . properties dosyasında), her çıkış sınıfı virgülle ayrılmış olarak hangi izleme çıkış sınıflarının yükleneceğini belirtir. Örneğin:

```
monitorExitClasses=testExits.TestExit1,testExits.testExit2
```

İzleme programı kullanıcı çıkışına ilişkin arabirim:

```
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);
}

```

İzleme programı meta verilerindeki IBM için ayrılmış değerler aşağıdaki gibidir:

```

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";

}

```

Örnek izleyici kullanıcı çıkışı

Bu örnek sınıf MonitorExit arabirimini uygular. Bu örnek, *REDIRECTEDAGENT* adlı izleyici meta verilerine özel bir yerine koyma değişkeni ekler; bu değişkene günün saati tek sayıysa LONDON değeri, çift saatler için PARIS değeri yerleştirilir. İzleme programı çıkış sonuç kodu her zaman proceeddeğerini döndürecek şekilde ayarlanır.

```

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;
    }
}

```

REDIRECTEDAGENT yerine koyma değişkenini kullanan bir izleme programına ilişkin görev aşağıdakilere benzer:

```

<?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>

```

Bu aktarma başlatılmadan önce, <destinationAgent> ögesinin agent özniteliğinin değeri LONDON ya da PARIS ile değiştirilir.

İkame değişkenini izleyici çıkış sınıfında ve görev tanımlaması XML ' inde büyük harfle belirtmeniz gerekir.

İlgili kavramlar

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

“MFT kullanıcı çıkışları için meta veriler” sayfa 2090

Managed File Transfer için kullanıcı çıkış yordamlarına sağlanabilecek üç farklı meta veri tipi vardır: ortam, aktarma ve dosya meta verileri. Bu meta veriler, Java anahtar-değer çiftlerinin eşlemeleri olarak sunulur.

“MFT kullanıcı çıkışları için Java arabirimleri” sayfa 2100

Kullanıcı çıkış yordamlarına ilişkin Java arabirimlerine ilişkin başvuru bilgileri için bu bölümdeki konuları kullanın.

[MFT kaynak ve hedef kullanıcı çıkışları](#)

İlgili başvurular

“MFT Kullanıcı çıkışları için aracı özellikleri” sayfa 2098

agent.properties dosyasındaki standart özelliklere ek olarak, özellikle kullanıcı çıkış yordamları için birkaç gelişmiş özellik vardır. Bu özelliklerden herhangi birini kullanmak istiyorsanız, agent.properties dosyasını el ile düzenlemeniz gerekir. Aracı çalışırken agent.properties dosyasında bir değişiklik yaparsanız, değişiklikleri almak için aracıyı durdurun ve yeniden başlatın.

MFT Kullanıcı çıkışları için aracı özellikleri

agent.properties dosyasındaki standart özelliklere ek olarak, özellikle kullanıcı çıkış yordamları için birkaç gelişmiş özellik vardır. Bu özelliklerden herhangi birini kullanmak istiyorsanız, agent.properties dosyasını el ile düzenlemeniz gerekir. Aracı çalışırken agent.properties dosyasında bir değişiklik yaparsanız, değişiklikleri almak için aracıyı durdurun ve yeniden başlatın.

Ortam değişkenleri, dosya ya da izin konumlarını gösteren bazı Managed File Transfer özelliklerinde kullanılabilir. Bu, ürünün parçalarını çalıştırırken kullanılan dosyaların ya da dizinlerin konumlarının, işlemi hangi kullanıcının çalıştırdığı gibi ortam değişikliklerine bağlı olarak değişmesini sağlar. Daha fazla bilgi için bkz. [MFT özelliklerindeki ortam değişkenleri](#).

Kullanıcı çıkış yordamı özellikleri

Kullanıcı çıkış yordamları, aşağıdaki çizelgede listelenen sırayla çağrılır. agent.properties kütüğüne ilişkin ek bilgi için [Advanced Agent properties: User exit routine](#) başlıklı konuya bakın.

Çizelge 885. Kullanıcı çıkışları için aracı özellikleri	
Özellik adı	Açıklama
sourceTransferEndExitSınıfları	Kaynak aktarma sonu çıkış yordamını gerçekleştiren sınıfların virgülle ayrılmış listesini belirtir.
sourceTransferStartExitSınıfları	Kaynak aktarma başlangıcı çıkış yordamını gerçekleştiren sınıfların virgülle ayrılmış listesini belirtir.
destinationTransferStartExitSınıfları	Hedef aktarma başlatma kullanıcı çıkış yordamını gerçekleştiren sınıfların virgülle ayrılmış listesini belirtir.

Çizelge 885. Kullanıcı çıkışları için aracı özellikleri (devamı var)	
Özellik adı	Açıklama
destinationTransferEndExitSınıfları	Hedef aktarma kullanıcı çıkışı yordamını gerçekleştiren sınıfların virgülle ayrılmış listesini belirtir.
exitClassYolu	<p>Kullanıcı çıkışı yordamları için sınıf yolu görevi yapan dizinlerin altyapıya özgü, karakterle ayrılmış bir listesini belirtir.</p> <p>Aracının çıkış dizini, bu sınıf yolundaki girdilerden önce aranır.</p> <p>Parantezler, virgüller (,) ve ters eğik çizgiler (\) MFT komutlarında özel karakterlerdir ve ters eğik çizgi (\) karakteri kullanılmalıdır. Windows</p> <p>Windows 'ta dosya yolları, ayırıcı olarak çift ters eğik çizgi (\\) kullanılarak ya da tek eğik çizgi (/) kullanılarak belirtilebilir. Örneğin:</p> <pre>exitClassPath=C:/mycomp/mqft/exits/encryptFileExit.jar; C:/mycomp/mqft/exits/fileFilter.jar.</pre> <p>Bu özelliğin değeri ortam değişkenlerini içerebilir.</p>
exitNativeLibraryPath	<p>Kullanıcı çıkışı yordamları için yerel kitaplık yolu olarak işlev görece dizinlerin altyapıya özgü, karakterle ayrılmış bir listesini belirtir.</p> <p>Bu özelliğin değeri ortam değişkenlerini içerebilir.</p>
monitorExitSınıflar	İzleme programı çıkış yordamını gerçekleştiren sınıfların virgülle ayrılmış listesini belirtir. Daha fazla bilgi için bkz. MFT kaynak izleyicisi kullanıcı çıkışları .
protocolBridgeCredentialExitSınıflar	Protokol köprüsü kimlik bilgileri kullanıcı çıkışı yordamını gerçekleştiren sınıfların virgülle ayrılmış listesini belirtir. Daha fazla bilgi için Çıkış sınıflarını kullanarak dosya sunucusuna ilişkin kimlik bilgilerini eşleme başlıklı konuya bakın.
protocolBridgePropertiesExitSınıflar	Protokol köprüsü sunucusu özellikleri kullanıcı çıkışı yordamını uygulayan sınıfların virgülle ayrılmış listesini belirtir. Daha fazla bilgi için bkz. ProtocolBridgePropertiesExit2: İletişim kuralı dosya sunucusu özelliklerinin araştırılması .
IOExitClasses	G/Ç kullanıcı çıkışı yordamını gerçekleştiren sınıfların virgülle ayrılmış listesini belirtir. Yalnızca IOExit arabirimini gerçekleştiren sınıfları listeleyn; örneğin, IOExitResourcePath ve IOExitChannel gibi diğer G/Ç kullanıcı çıkışı arabirimlerini uygulayan sınıfları listeleme. Daha fazla bilgi için bkz. MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması .

Çıkış çağırma sırası

Kaynak ve hedef çıkışlar aşağıdaki sırayla çağırılır:

1. SourceTransferStartExit
2. DestinationTransferStartExit
3. DestinationTransferEndExit
4. SourceTransferEndExit

Kaynak ve hedef çıkışları zincirleme olarak oluşturma

Birden çok çıkış belirtirseniz, önce listedeki ilk çıkış, ardından ikinci çıkış, vb. çağırılır. İlk çıkışta yapılan değişiklikler, daha sonra çağırılan çıkışa giriş olarak geçirilir ve bu şekilde devam eder. Örneğin, iki kaynak aktarma başlangıcı varsa, birinci çıkışla meta verilerde yapılan değişiklikler ikinci çıkışa giriş olur. Her çıkış kendi sonucunu döndürür. Verili bir tipteki tüm çıkışlar aktarma sonucu kodu olarak DEVAM (DEVAM) döndürüyorsa, genel sonuç DEVAM olur. Bir ya da daha çok çıkış CANCEL_TRANSFER değerini döndürirse, genel sonuç CANCEL_TRANSFER olur. Çıkışların döndürdüğü sonuç kodlarının ve dizgilerin tümü, aktarma günlüğüne yazılır.

Kaynak aktarma başlangıç çıkışının genel sonucu DEVAM ise, aktarma, çıkışlar tarafından yapılan tüm değişiklikleri kullanarak devam eder. Genel sonuç CANCEL_TRANSFER ise, kaynak aktarma bitiş çıkışları çağırılır ve aktarma iptal edilir. Aktarma günlüğündeki tamamlanma durumu "iptal edildi".

Hedef aktarma başlangıcı çıkışlarından elde edilen genel sonuç DEVAM ise, aktarım, çıkışlar tarafından yapılan tüm değişiklikleri kullanarak devam eder. Genel sonuç CANCEL_TRANSFER ise, hedef aktarma sonu çıkışları çağrılır ve kaynak aktarma bitiş çıkışları çağrılır. Son olarak transfer iptal edildi. Aktarma günlüğündeki tamamlanma durumu "iptal edildi".

Bir kaynak ya da hedef çıkışın, zincirdeki aşağıdaki çıkışlara ya da yürütme sırasına bilgi aktarması gerekiyorsa, aktarma meta verileri güncellenerek gerçekleştirilmelidir. Aktarma meta verilerinin kullanımı, çıkış uygulamasına özgüdür. Örneğin, bir çıkış dönüş sonucunu CANCEL_TRANSFER olarak ayarlarsa ve aktarma işleminin iptal edildiğini bildirmek için aşağıdaki çıkışlarla iletişim kurması gerekiyorsa, diğer çıkışlar tarafından anlaşılacak şekilde bir aktarma meta veri değeri ayarlanarak gerçekleştirilmelidir.

Örnek

```
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
```

İlgili kavramlar

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

“MFT kullanıcı çıkışları için meta veriler” sayfa 2090

Managed File Transfer için kullanıcı çıkış yordamlarına sağlanabilecek üç farklı meta veri tipi vardır: ortam, aktarma ve dosya meta verileri. Bu meta veriler, Java anahtar-değer çiftlerinin eşlemeleri olarak sunulur.

“MFT kullanıcı çıkışları için Java arabirimleri” sayfa 2100

Kullanıcı çıkışı yordamlarına ilişkin Java arabirimlerine ilişkin başvuru bilgileri için bu bölümdeki konuları kullanın.

İlgili başvurular

“MFT kaynak izleyicisi kullanıcı çıkışları” sayfa 2094

Kaynak izleyicisi kullanıcı çıkışları, ilişkili görev başlatılmadan önce, bir izleme programının tetikleyici koşulu karşılandığında çalışacak şekilde özel kod yapılandırmanızı sağlar.

[MFT özelliklerinde ortam değişkenleri](#)

[MFT agent.properties dosyası](#)

MFT kullanıcı çıkışları için Java arabirimleri

Kullanıcı çıkışı yordamlarına ilişkin Java arabirimlerine ilişkin başvuru bilgileri için bu bölümdeki konuları kullanın.

İlgili görevler

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

İlgili başvurular

“DestinationTransferStartExit.java arabirimi” sayfa 2104

“DestinationTransferEndExit.java arabirimi” sayfa 2103

“IOExit.java arabirimi” sayfa 2106

“IOExitChannel.java arabirimi” sayfa 2108

“IOExitLock.java arabirimi” sayfa 2110

“IOExitPath.java arabirimi” sayfa 2111

“IOExitProperties.java arabirimi” sayfa 2112

“IOExitRecordChannel.java arabirimi” sayfa 2115

“IOExitRecordResourcePath.java arabirimi” sayfa 2116

“IOExitResourcePath.java arabirimi” sayfa 2118

“IOExitWildcardPath.java arabirimi” sayfa 2122

[“MonitorExit.java arabirimi” sayfa 2122](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

[“ProtocolBridgeCredentialExit2.java arabirimi” sayfa 2124](#)

[“ProtocolBridgePropertiesExit2.java arabirimi” sayfa 2125](#)

[“SourceTransferStartExit.java arabirimi” sayfa 2129](#)

[“SourceTransferEndExit.java arabirimi” sayfa 2128](#)

CDCredentialExit.java arabirimi

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);    }

```

CredentialExitResult.java arabirimi

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;
    }
}

```

İlgili görevler

MFT ürününü kullanıcı çıktılarıyla özelleştirme

İlgili başvurular

[“SourceTransferStartExit.java arabirimi” sayfa 2129](#)

[“DestinationTransferStartExit.java arabirimi” sayfa 2104](#)

[“DestinationTransferEndExit.java arabirimi” sayfa 2103](#)

[“MonitorExit.java arabirimi” sayfa 2122](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

DestinationTransferEndExit.java arabirimi

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 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 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);
}

```

İlgili görevler

MFT ürününü kullanıcı çıkışlarıyla özelleştirme

İlgili başvurular

[“SourceTransferStartExit.java arabirimi” sayfa 2129](#)

[“SourceTransferEndExit.java arabirimi” sayfa 2128](#)

[“DestinationTransferStartExit.java arabirimi” sayfa 2104](#)

[“MonitorExit.java arabirimi” sayfa 2122](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

DestinationTransferStartExit.java arabirimi

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 <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 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);

```

İlgili görevler

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

İlgili başvurular

[“SourceTransferStartExit.java arabirimi” sayfa 2129](#)

[“SourceTransferEndExit.java arabirimi” sayfa 2128](#)

[“DestinationTransferEndExit.java arabirimi” sayfa 2103](#)

[“MonitorExit.java arabirimi” sayfa 2122](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

FileTransferResult.java arabirimi

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();
}

```

İlgili görevler

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

İlgili başvurular

[“SourceTransferStartExit.java arabirimi” sayfa 2129](#)

[“DestinationTransferStartExit.java arabirimi” sayfa 2104](#)

[“DestinationTransferEndExit.java arabirimi” sayfa 2103](#)

[“MonitorExit.java arabirimi” sayfa 2122](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

IOExit.java arabirimi

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;

```

İlgili görevler

[MFT aktarım G/Ç kullanıcı çıktıların kullanılması](#)

[MFT ürününü kullanıcı çıktılarıyla özelleştirme](#)

IOExitChannel.java arabirimi

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;
}

```

İlgili görevler

[MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

IOExitLock.java arabirimi

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();
}

```

İlgili görevler

MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması

MFT ürününü kullanıcı çıkışlarıyla özelleştirme

IOExitPath.java arabirimi

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();
}

```

İlgili görevler

[MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

IOExitProperties.java arabirimi

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;
}
}

```

İlgili görevler

[MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

IOExitRecordChannel.java arabirimi

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;
}

```

İlgili görevler

[MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

 ***IOExitRecordResourcePath.java arabirimi***

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;
}

```

İlgili görevler

[MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

IOExitResourcePath.java arabirimi

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(IOExitResourcePath 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 IOExitResourcePath} 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();
}

```

İlgili görevler

[MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

IOExitWildcardPath.java arabirimi

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 {

```

İlgili görevler

[MFT aktarım G/Ç kullanıcı çıkışlarının kullanılması](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

MonitorExit.java arabirimi

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 <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);
}

```

İlgili görevler

[MFT kaynaklarının izlenmesi](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

İlgili başvurular

[“SourceTransferStartExit.java arabirimi” sayfa 2129](#)

[“SourceTransferEndExit.java arabirimi” sayfa 2128](#)

[“DestinationTransferStartExit.java arabirimi” sayfa 2104](#)

[“DestinationTransferEndExit.java arabirimi” sayfa 2103](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

ProtocolBridgeCredentialExit.java arabirimi

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);
}

```

İlgili görevler

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

[Çıkış sınıflarını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme](#)

ProtocolBridgeCredentialExit2.java arabirimi

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);
}

```

İlgili görevler

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

[Çıkış sınıflarını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme](#)

ProtocolBridgePropertiesExit2.java arabirimi

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);
}

```

İlgili görevler

[ProtocolBridgePropertiesExit: Protokol dosya sunucusu özellikleri arıyor](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

[Çıkış sınıflarını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme](#)

SourceFileExitFileSpecification.java sınıfı

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;
    }
}

```

İlgili kavramlar

“MFT kullanıcı çıkışları için meta veriler” sayfa 2090

Managed File Transfer için kullanıcı çıkış yordamlarına sağlanabilecek üç farklı meta veri tipi vardır: ortam, aktarma ve dosya meta verileri. Bu meta veriler, Java anahtar-değer çiftlerinin eşlemeleri olarak sunulur.

SourceTransferEndExit.java arabirimi

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);
```

```
}
```

İlgili görevler

[MFT ürününü kullanıcı çıktılarıyla özelleştirme](#)

İlgili başvurular

[“SourceTransferStartExit.java arabirimi” sayfa 2129](#)

[“DestinationTransferStartExit.java arabirimi” sayfa 2104](#)

[“DestinationTransferEndExit.java arabirimi” sayfa 2103](#)

[“MonitorExit.java arabirimi” sayfa 2122](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

SourceTransferStartExit.java arabirimi

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 <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 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 <code>TransferMetaDataConstants</code> 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);
}

```

İlgili görevler

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

İlgili başvurular

[“SourceFileExitFileSpecification.java sınıfı” sayfa 2126](#)

[“SourceTransferEndExit.java arabirimi” sayfa 2128](#)

[“DestinationTransferStartExit.java arabirimi” sayfa 2104](#)

[“DestinationTransferEndExit.java arabirimi” sayfa 2103](#)

[“MonitorExit.java arabirimi” sayfa 2122](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

TransferExitResult.java arabirimi

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;
}
}

```

İlgili görevler

MFT ürününü kullanıcı çıkışlarıyla özelleştirme

İlgili başvurular

[“SourceTransferStartExit.java arabirimi” sayfa 2129](#)

[“DestinationTransferStartExit.java arabirimi” sayfa 2104](#)

[“DestinationTransferEndExit.java arabirimi” sayfa 2103](#)

[“MonitorExit.java arabirimi” sayfa 2122](#)

[“ProtocolBridgeCredentialExit.java arabirimi” sayfa 2123](#)

MFT Agent komut kuyruğuna koyabileceğiniz iletilere ilişkin ileti biçimleri

Bu XML şemaları, aracının bir eylem gerçekleştirmesini istemek için aracı komut kuyruğuna konabilecek iletilere ilişkin biçimleri tanımlar. XML iletisi, komut satırı komutları kullanılarak ya da bir uygulama tarafından aracı komut kuyruğuna yerleştirilebilir.

- [Kütük aktarma isteği iletisi biçimi](#)
- [MFT izleme programı istek iletisi biçimleri](#)
- [Ping MFT aracı istek iletisi biçimi](#)
- [MFT aracı yanıt iletisi biçimi](#)

İleti alışverişi REST API başvurusu

messaging REST API ile ilgili başvuru bilgileri.

messaging REST API kullanımı hakkında daha fazla bilgi için bkz. [Messaging using the REST API](#).

REST API Kaynak

Bu konu derlemi, messaging REST API kaynaklarının her birine ilişkin başvuru bilgileri sağlar.

messaging REST API kullanımı hakkında daha fazla bilgi için bkz. [Messaging using the REST API](#).

/messaging/qmgr/{qmgrName}/queue/{queueName}/message

İleti alışverişi REST API 'si, iletilerin bir kuyruğa konmasına ya da iletilere /messaging/qmgr/{qmgrName}/queue/{queueName}/message kaynağı kullanılarak göz atılmasına ya da kuyruktan yıkıcı bir şekilde alınmasına olanak sağlar.

POST /messaging/qmgr/{qmgrName}/queue/{queueName}/message

İletileri belirtilen kuyruk yöneticisinde belirtilen kuyruğa koymak için HTTP POST yöntemini /messaging/qmgr/{qmgrName}/queue/{queueName}/message kaynağıyla birlikte kullanabilirsiniz.

HTTP istek gövdesini içeren bir IBM MQ iletisini belirtilen kuyruk yöneticisine ve kuyruğa koyar. Kuyruk yöneticisi, mqweb sunucusuyla aynı makinede olmalıdır. Yöntem yalnızca metin tabanlı HTTP istek gövdelerini destekler. İletiler, MQSTR ya da JMS TextMessage olarak biçimlendirilmiş iletiler olarak gönderilir ve geçerli kullanıcı bağlamı kullanılarak konur.

V 9.3.0 REST API V3 , kullanıcı tanımlı ileti özelliklerini belirleme ve ileti önceliğini dahil etme yeteneğini ekler. ibm-mq-md-priority ve ibm-mq-usr istek üstbilgileri yalnızca REST API V3 ile kullanılabilir. ibm-mq-md-correlationId istek üstbilgisi, REST API V3 içinde farklı bir biçime sahiptir. Üstbilgi uygulamaya özgü bir tanıtıcı olabilir ya da kodlanmış bir dizgi için ID: öneki gerekir. POST isteğiniz kullanıcı tanımlı iletiler ya da uygulamaya özgü ilinti tanıtıcısı içeriyorsa, ileti JMS TextMessage olarak biçimlendirilir.

- “Kaynak URL” sayfa 2132
- “İstek üstbilgileri” sayfa 2133
- “İstek gövdesi biçimi” sayfa 2135
- “Güvenlik Gereksinimleri” sayfa 2135
- “Yanıt durumu kodları” sayfa 2136
- “Yanıt üstbilgileri” sayfa 2136
- “Yanıt gövdesi biçimi” sayfa 2137
- “Örnekler” sayfa 2137

Kaynak URL

`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

İleti sistemi için bağlanılacak kuyruk yöneticisinin adını belirtir. Kuyruk yöneticisi, mqweb sunucusuyla aynı makinede olmalıdır.

V 9.3.3 IBM MQ 9.3.3' den yerel bir kuyruk yöneticisine ya da uzak bir kuyruk yöneticisine bağlanabilirsiniz. **qmgrName** için belirttiğiniz ad, mqweb sunucunuzun nasıl yapılandırıldığına bağlıdır:

- mqweb sunucunuz yalnızca yerel kuyruk yöneticilerine bağlanacak şekilde yapılandırıldıysa, kuyruk yöneticisinin adını kullanın.
- mqweb sunucunuz uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırıldıysa, uzak kuyruk yöneticisi bağlantı bilgilerinde belirtildiği şekilde kuyruk yöneticisinin benzersiz adını kullanın.

dspmqweb properties komutunu kullanarak ve **mqRestMessagingConnectionMode** özelliğini görüntüleyerek, mqweb sunucunuzun yerel kuyruk yöneticilerine ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırılıp yapılandırılmadığını belirleyebilirsiniz.

Kuyruk yöneticisi adı büyük ve küçük harfe duyarlıdır.

Kuyruk yöneticisi adında eğik çizgi, nokta imi ya da yüzde imi varsa, bu karakterler URL biçiminde kodlanmış olmalıdır:

- Eğik çizgi %2Folarak kodlanmalıdır.
- Bir dönem %2Eolarak kodlanmalıdır.
- Yüzde işareti %25olarak kodlanmalıdır.

queueName

İletinin konacağı kuyruğun adını belirler.

Kuyruk, belirtilen kuyruk yöneticisinin yerel, uzak ya da diğer adı olarak tanımlanmalıdır; bu, kümelenmiş bir kuyruğa da gönderme yapabilir.

Kuyruk adı büyük ve küçük harfe duyarlıdır.

Kuyruk adı eğik çizgi ya da yüzde işareti içeriyorsa, bu karakterler URL kodlu olmalıdır:

- Eğik çizgi,/, %2Folarak kodlanmalıdır.
- Yüzde işareti,%, %25olarak kodlanmalıdır.

HTTP bağlantılarını etkinleştirirseniz, HTTPS yerine HTTP kullanabilirsiniz. HTTP' yi etkinleştirme hakkında daha fazla bilgi için bkz. [HTTP ve HTTPS kapılarını yapılandırma](#).

İstek üstbilgileri

Aşağıdaki üstbilgiler istekle birlikte gönderilmelidir:

Yetkilendirme

Temel kimlik doğrulaması kullanıyorsanız bu üstbilgi gönderilmelidir. Daha fazla bilgi için bkz. [REST API ile HTTP temel kimlik doğrulamasını kullanma](#).

İçerik Tipi

Bu üstbilgi aşağıdaki değerlerden biriyle gönderilmelidir:

- 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

Bu üstbilgi ayarlanmalıdır, ancak değer boş olmak da dahil olmak üzere herhangi bir şey olabilir.

İsteğe bağlı olarak aşağıdaki üstbilgiler istekle birlikte gönderilebilir:

Kabul Et-Dil

Bu üstbilgi, yanıt iletisi gövdesinde döndürülen kural dışı durumlar ya da hata iletileri için gereken dili belirtir.

REST API V1 > REST API V2 ibm-mq-md-correlationId

Bu üstbilgi, yaratılan iletinin ilinti tanıtıcısını ayarlar. Üstbilgi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirtilir. Değerin başına "ID: " önekini eklemeyin, REST API o dizgiyi otomatik olarak ekler.

Örneğin: ibm-mq-md-correlationId:

414d5120514d4144455620202020202067d8bf5923582e02

V 9.3.0 > REST API V3 ibm-mq-md-correlationId

Bu üstbilgi, yaratılan iletinin ilinti tanıtıcısını ayarlar. İlinti tanıtıcısı aşağıdaki formlardan birini alabilir:

- Öneki "ID: " olan, 24 baytı gösteren, 48 karakterlik onaltılı kodlanmış dizgi. Örneğin: ibm-mq-md-correlationId: ID:414d5120514d4144455620202020202067d8bf5923582e02

- Uygulamaya özgü bir değer. Değer, uygulamaya özgü bir dizedir: `ibm-mq-md-correlationId: My-Custom-CorrelId`

Bu ilinti tanıtıcısı biçimini belirtirseniz, ileti hedefi `WMQ_CLIENT_JMS_UYUMLU` olarak hedeflenir ve bu nedenle bir `MQRFH2` üstbilgisi içerir.

ibm-mq-md-süre bitimi

Bu üstbilgi, oluşturulan ileti için süre bitimini ayarlar. Bir iletinin süre bitimi, iletinin kuyruğa geldiği zamandan başlar. Sonuç olarak ağ gecikme süresi yoksayılr. Üstbilgi aşağıdaki değerlerden biri olarak belirtilmelidir:

sınırsız

İletinin süresi dolmaz.

Bu değer varsayılan değerdir.

Tamsayı değeri

İleti süre bitiminden önce milisaniye.

0-9999999999900 aralığıyla sınırlıdır.

ibm-mq-md-kalıcılık

Bu üstbilgi, oluşturulan ileti için kalıcılığı ayarlar. Üstbilgi aşağıdaki değerlerden biri olarak belirtilir:

nonPersistent

İleti, sistem hatalarından ya da kuyruk yöneticisi yeniden başlatmadan sonra devam etmez.

Bu değer varsayılan değerdir.

Kalıcı

İleti, sistem hatalarından ya da kuyruk yöneticisini yeniden başlatmaya devam eder.

V 9.3.0

REST API V3

ibm-mq-md-öncelik

REST API V3 için bu üstbilgi, oluşturulan iletinin önceliğini ayarlar. Üstbilgi aşağıdaki değerlerden biri olarak belirtilir:

asDestination

İleti, temel IBM MQ kuyruk nesnesinin `DEFPRTY` özneliğinde belirtilen önceliği kullanır.

Bu değer varsayılan değerdir.

Tamsayı değeri

Gerçek önceliği 0-9 aralığında bir tamsayı olarak belirtin.

Örneğin: `ibm-mq-md-priority: asDestination`

REST API V1 ve REST API V2 için POST için ileti önceliği her zaman 4 'tür.

ibm-mq-md-replyTo

Bu üstbilgi, yaratılan ileti için yanıt hedefini ayarlar. Üstbilginin biçimi, yanıt kuyruğunun ve isteğe bağlı bir kuyruk yöneticisinin sağlanmasına ilişkin standart gösterimi kullanır: `replyQueue[@replyQmgr]`

Örneğin: `ibm-mq-md-replyTo: myReplyQueue@myReplyQMGr`

V 9.3.0

REST API V3

ibm-mq-usr

Bu üstbilgi, istek iletisi kullanıcı tanımlı özelliklerini ayarlar.

Özelliklerin sözdizimi şöyledir:

`ibm-mq-usr: property_name; user_value; user_type`

özellik_adi

Belirtilen kullanıcı özelliğinin adı. Bu geçerli bir JMS özelliği adı olmalıdır.

kullanıcı_değeri

Özelliğin değeri.

kullanıcı_tipi

Özelliğin tipi:

- boolean (doğru/yanlış, MQBOOL)
- byte (8 bit tamsayı, MQINT8)

- short (16 bitlik tamsayı, MQINT16)
- integer (32 bitlik tamsayı, MQINT32)
- long (64 bitlik tamsayı, MQINT64)
- float (32 bit gerçek, MQFLOAT32)
- double (64 bit gerçek, MQFLOAT64)
- string (tırnak içine alınmış dizgi)

Bir iletide birden çok özellik ayarlayabilirsiniz. Tek bir ibm-mq-usr istek üstbilgisinde birden çok virgülle ayrılmış özellik belirtebilir ya da ibm-mq-usr istek üstbilgisinin iki ya da daha fazla ayrı örneğini kullanabilirsiniz.

Örneğin, tek bir üstbilgide birden çok kullanıcı tanımlı özellik ayarlayabilirsiniz: `ibm-mq-usr: userPropertyA;5;byte, userPropertyB; -10; integer`

Ya da üstbilginin ayrı eşgörünümlerinde birden çok kullanıcı tanımlı özellik ayarlayabilirsiniz: `ibm-mq-usr: userPropertyA;5;byte ibm-mq-usr: userPropertyB; -10; integer`

İstek gövdesi biçimi

İstek gövdesi metin olmalı ve UTF-8 kodlamasını kullanmalıdır. Belirli bir metin yapısı gerekli değildir. İstek gövdesi metnini içeren MQSTR biçimli bir ileti oluşturulur ve belirtilen kuyruğa yerleştirilir.

V 9.3.0 **REST API V3** REST API V3 kullanıcı tanımlı özellikleri ya da uygulamaya özgü ilinti tanıtıcısı özellikleri kullanılırsa, istek gövdesi metnini içeren JMS `TextMessage` biçimli bir ileti yaratılır ve belirtilen kuyruğa yerleştirilir.

Daha fazla bilgi için bkz. [örnekler](#).

Güvenlik Gereksinimleri

Çağıranın kimliği mqweb sunucusunda doğrulanmalıdır. MQWebAdmin ve MQWebAdminRO rolleri messaging REST API için geçerli değildir. REST API güvenliği hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenlik](#).

mqweb sunucusunda kimliği doğrulandıktan sonra, kullanıcı hem messaging REST API , hem de administrative REST API kullanabilir.

Çağıranın güvenlik birincil kullanıcıasına, iletileri belirtilen kuyruğa koyma yeteneği verilmelidir:

- URL kaynağının `{queueName}` kısmı tarafından belirtilen kuyruk PUT etkinleştirilmiş olmalıdır.
- **MQ Appliance** **ALW** URL kaynağının `{queueName}` kısmı tarafından belirtilen kuyruk için, +PUT yetkisi çağıranın güvenlik birincil kullanıcıasına verilmelidir.
- **z/OS** URL kaynağının `{queueName}` kısmı tarafından belirtilen kuyruk için, UPDATE erişimi çağıranın güvenlik birincil kullanıcıasına verilmelidir.

Bu güvenlik birincil kullanıcı, mqweb sunucusunu başlatan kullanıcı ya da mqweb sunucusunda oturum açan kullanıcı olabilir. Bağlandığınız kuyruk yöneticisi uzak bir kuyruk yöneticisiyse, güvenlik birincil kullanıcı uzak kuyruk yöneticisi bağlantı bilgilerindeki kimlik bilgileri tarafından sağlanan kullanıcı ya da kanal güvenlik kurallarının belirlediği başka bir kullanıcı olabilir. Daha fazla bilgi için, bkz. [messaging REST API tarafından kullanılan güvenlik birincil kullanıcıasının belirlenmesi](#).

ALW AIX, Linux, and Windows' da, **setmqaut** komutunu kullanarak IBM MQ kaynaklarını kullanmak için güvenlik birincil kullanıcıları için yetki verebilirsiniz. Daha fazla bilgi için bkz. [setmqaut](#) (yetki verilmesi ya da iptal etmek).

z/OS z/OS işletim üzerinde bkz. [z/OS üzerinde güvenliğin ayarlanması](#).

messaging REST API ile Gelişmiş İleti Güvenliği (Advanced Message Security; AMS) kullanıyorsanız, tüm iletilerin iletiyi postalayan kullanıcının bağlamı değil, mqweb sunucusunun bağlamı kullanılarak şifrelendiğini unutmayın.

Yanıt durumu kodları

201

İleti başarıyla oluşturuldu ve gönderildi.

400

Geçersiz veri sağlandı.

Örneğin, geçersiz bir istek üstbilgisi değeri belirtildi.

401

Kimlik doğrulanmadı.

Çağıranın kimliği mqweb sunucusunda doğrulanmalı ve MQWebAdmin, MQWebAdminROya da MQWebUser rollerinden birinin ya da daha fazlasının üyesi olmalıdır. `ibm-mq-rest-csrf-token` üstbilgisi de belirtilmelidir. Daha fazla bilgi için bkz [“Güvenlik Gereksinimleri” sayfa 2135](#).

403

Yetkili değil.

Çağıranın kimliği mqweb sunucusunda doğrulanır ve geçerli bir birincil kullanıcıyla ilişkilendirilir. Ancak, birincil kullanıcının gereken IBM MQ kaynaklarının tümüne ya da bir alt kümesine erişimi yok ya da MQWebUser rolünde yok. Gerekli erişim hakkında daha fazla bilgi için bkz. [“Güvenlik Gereksinimleri” sayfa 2135](#).

404

Kuyruk yok.

405

Kuyruk PUT engellenmiş.

415

İleti üstbilgisi ya da gövdesi desteklenmeyen bir ortam tipidir.

Örneğin, Content-Type üstbilgisi desteklenmeyen bir ortam tipine ayarlanır.

500

IBM MQ' den sunucu sorunu ya da hata kodu.

502

İleti alışverişi sağlayıcısı gereken işlevi desteklemediğinden, yürürlükteki güvenlik birincil kullanıcısı iletiyi gönderemiyor. Örneğin, mqweb sunucusu sınıf yolu geçersizse.

503

Kuyruk yöneticisi çalışmıyor.

Yanıt üstbilgileri

Yanıtla birlikte aşağıdaki üstbilgiler döndürülür:

İçerik-Dil

Herhangi bir hata ya da kural dışı durum oluşması durumunda yanıt iletilisinin dil tanıtıcısını belirtir. Herhangi bir hata ya da kural dışı durum koşulu için gerekli dili belirtmek üzere `Accept-Language` istek üstbilgisiyle birlikte kullanılır. İstenen dil desteklenmezse, mqweb sunucusu varsayılan değeri kullanılır.

İçerik Uzunluğu

İçerik olmadığında bile HTTP yanıt gövdesinin uzunluğunu belirtir. Başarıdan sonra değer sıfırdır.

İçerik Tipi

Yanıt gövdesinin tipini belirtir. Başarıya ulaştığı anda değer `text/plain; charset=utf-8` olur. Herhangi bir hata ya da kural dışı durum olması durumunda, değer `application/json; charset=utf-8` olur.

REST API V1 REST API V2 **ibm-mq-md-messageId**

IBM MQ tarafından bu iletiye ayrılan ileti tanıtıcısını belirtir. `ibm-mq-md-correlationId` istek üstbilgisi gibi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak gösterilir.

Örneğin:

```
ibm-mq-md-messageId: 414d5120514d4144455620202020202067d8ce5923582f07
```

V 9.3.0 REST API V3 **ibm-mq-md-messageId**

IBM MQ tarafından bu iletiye ayrılan ileti tanıtıcısını belirtir. `ibm-mq-md-correlationId` istek üstbilgisi gibi, önceki ID: olan 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak gösterilir.

Örneğin:

```
ibm-mq-md-messageId: ID:414d5120514d4144455620202020202067d8ce5923582f07
```

V 9.3.3 **ibm-mq-çözümlendi-qmgr**

İstek için kullanılan kuyruk yöneticisinin adını belirtir. URL kaynağında benzersiz bir ad kullanıldıysa, bu üstbilgi o benzersiz adla ilişkilendirilmiş kuyruk yöneticisinin adını belirtir. URL kaynağında kullanılan benzersiz ad bir kuyruk yöneticisi grubuna gönderme yapıyorsa, bu üstbilgi grup içindeki hangi kuyruk yöneticisinin kullanıldığını gösterir.

Yanıt gövdesi biçimi

İleti başarıyla gönderilirse yanıt gövdesi boş olur. Bir hata oluşursa, yanıt gövdesi bir hata iletisi içerir. Daha fazla bilgi için bkz. [REST API hata işleme](#).

Örnekler

Aşağıdaki örneklerde v2 kaynak URL 'si kullanılır. IBM MQ Daha önceki bir IBM MQ 9.1.5 sürümünü kullanıyorsanız, bunun yerine v1 kaynak URL adresini kullanmanız gerekir. Yani, kaynak URL 'sinde, örnek URL 'nin v2adresini kullandığı v1 yerine kullanın.

Aşağıdaki örnek, `mquser` parolasıyla `mquser` adlı bir kullanıcıda oturum açar. cURL' de oturum açma isteği aşağıdaki Windows örneğine benzer. LTPA belirteci, -c işareti kullanılarak `cookiejar.txt` dosyasında saklanır:

```
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
```

Kullanıcı oturum açtıktan sonra, LTPA simgesi ve `ibm-mq-rest-csrf-token` HTTP üstbilgisi, daha fazla isteğin kimliğini doğrulamak için kullanılır. `ibm-mq-rest-csrf-token token_value`, boşluk da içinde olmak üzere herhangi bir değer olabilir.

- Aşağıdaki Windows cURL örneği, varsayılan seçenekleri kullanarak Q1 kuyruk yöneticisinde QM1kuyruğa bir ileti gönderir. İleti "*Merhaba Dünya!*"metnini içerir:

```
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!"
```

- Aşağıdaki Windows cURL örneği, Q1 kuyruk yöneticisinde QM1kuyruğuna 2 dakika süre bitimiyle kalıcı bir ileti gönderir. İleti "*Merhaba Dünya!*"metnini içerir:

```
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!"
```

- Aşağıdaki Windows cURL örneği, süresi ve tanımlı ilinti tanıtıcısı olmayan Q1 kuyruk yöneticisinde QM1kuyruğuna kalıcı olmayan bir ileti gönderir. İleti "Merhaba Dünya!"metnini içerir:

```
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

İlişkili kuyruk yöneticisinden ve kuyruktan gelen iletilere göz atmak için /messaging/qmgr/{qmgrName}/queue/{queueName}/message kaynağıyla HTTP GET yöntemini kullanabilirsiniz.

Belirtilen kuyruk yöneticisinden ve kuyruktan gelen ilk kullanılabilir iletiye göz atılır. Kuyruk yöneticisi, mqweb sunucusuyla aynı makinede olmalıdır. HTTP yanıt gövdesinde ileti gövdesi döndürülür. İleti MQSTR ya da JMS TextMessage biçiminde olmalıdır ve yürürlükteki kullanıcı bağlamı kullanılarak alınır.

Tüm iletiler kuyrukta bırakılır ve uygun olmayan iletiler için çağırana uygun bir durum kodu döndürülür. Örneğin, MQSTR ya da JMS TextMessage biçiminde olmayan bir ileti.

V 9.3.0 REST API V3 , kullanıcı tanımlı ileti özelliklerini belirleme ve ileti önceliğini iletilere dahil etme yeteneğini ekler. ibm-mq-md-priority ve ibm-mq-usr yanıt üstbilgileri yalnızca REST API V3ile kullanılabilir. ibm-mq-md-correlationId istek üstbilgisi, REST API V3içinde farklı bir biçime sahiptir. Üstbilgi uygulamaya özgü bir tanıtıcı olabilir ya da kodlanmış bir dizgi ID: önekini korur. ibm-mq-md-messageId yanıt üstbilgisi ve sorgu parametresi, REST API V3' de farklı bir biçime sahiptir, ID: önekini korur.

- [“Kaynak URL” sayfa 2138](#)
- [“İsteğe bağlı sorgu parametreleri” sayfa 2139](#)
- [“İstek üstbilgileri” sayfa 2140](#)
- [“İstek gövdesi biçimi” sayfa 2140](#)
- [“Güvenlik Gereksinimleri” sayfa 2140](#)
- [“Yanıt durumu kodları” sayfa 2141](#)
- [“Yanıt üstbilgileri” sayfa 2141](#)
- [“Yanıt gövdesi biçimi” sayfa 2143](#)
- [“Örnekler” sayfa 2144](#)

Kaynak URL

`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

İleti sistemi için bağlanılacak kuyruk yöneticisinin adını belirtir. Kuyruk yöneticisi, mqweb sunucusuyla aynı makinede olmalıdır.

V 9.3.3 IBM MQ 9.3.3' den yerel bir kuyruk yöneticisine ya da uzak bir kuyruk yöneticisine bağlanabilirsiniz. **qmgrName** için belirttiğiniz ad, mqweb sunucunuzun nasıl yapılandırıldığına bağlıdır:

- mqweb sunucunuz yalnızca yerel kuyruk yöneticilerine bağlanacak şekilde yapılandırıldıysa, kuyruk yöneticisinin adını kullanın.

- mqweb sunucunuz uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırıldıysa, uzak kuyruk yöneticisi bağlantı bilgilerinde belirtildiği şekilde kuyruk yöneticisinin benzersiz adını kullanın.

dspmweb properties komutunu kullanarak ve **mqRestMessagingConnectionMode** özelliğini görüntüleyerek, mqweb sunucunuzun yerel kuyruk yöneticilerine ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırılıp yapılandırılmadığını belirleyebilirsiniz.

Kuyruk yöneticisi adı büyük ve küçük harfe duyarlıdır.

Kuyruk yöneticisi adında eğik çizgi, nokta imi ya da yüzde imi varsa, bu karakterler URL biçiminde kodlanmış olmalıdır:

- Eğik çizgi (/) %2Folarak kodlanmalıdır.
- Yüzde işareti (%) %25olarak kodlanmalıdır.

queueName

İletiyeye göz atılacak kuyruğun adını belirler.

Kuyruk, yerel olarak ya da yerel bir kuyruğu gösteren bir diğer ad olarak tanımlanmalıdır.

Kuyruk adı büyük ve küçük harfe duyarlıdır.

Kuyruk adı eğik çizgi ya da yüzde işareti içeriyorsa, bu karakterler URL kodlu olmalıdır:

- Eğik çizgi,/ , %2Folarak kodlanmalıdır.
- Yüzde işareti,% , %25olarak kodlanmalıdır.

HTTP bağlantılarını etkinleştirirseniz, HTTPS yerine HTTP kullanabilirsiniz. HTTP' yi etkinleştirme hakkında daha fazla bilgi için bkz. [HTTP ve HTTPS kapılarını yapılandırma](#).

İsteğe bağlı sorgu parametreleri

REST API V1 REST API V2 correlationId=hexValue

HTTP yönteminin, ilgili ilinti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirlenmelidir.

Örneğin:

```
../message?correlationId=414d5120514d414445562020202020202067d8bf5923582e02
```

V 9.3.0 REST API V3 correlationId= ID:hexValue ya da correlationId=application_specific_value

HTTP yönteminin, ilgili ilinti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren ve "ID:" dizgisinden önce gelen 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirtilmelidir.

Örneğin:

```
../message?correlationId=ID:414d5120514d414445562020202020202067d8bf5923582e02
```

uygulama_özel_değeri

Sorgu parametresi uygulamaya özgü bir dizgi olarak belirtilebilir.

Örneğin:

```
../message?correlationId=My-Custom-CorrelId
```

REST API V1 REST API V2 messageId=hexValue

HTTP yönteminin, karşılık gelen ileti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirlenmelidir.

Örneğin:

```
../message?messageId=414d5120514d4144455620202020202067d8ce5923582f07
```

V 9.3.0 REST API V3 messageId= Tanıtıcı:hexValue

HTTP yönteminin, karşılık gelen ileti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren ve "ID: " dizgisinden önce gelen 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirtilmelidir.

Örneğin:

```
../message?messageId=ID:414d5120514d4144455620202020202067d8ce5923582f07
```

İstek üstbilgileri

Aşağıdaki üstbilgiler istekle birlikte gönderilmelidir:

Yetkilendirme

Temel kimlik doğrulaması kullanıyorsanız bu üstbilgi gönderilmelidir. Daha fazla bilgi için bkz. [REST API ile HTTP temel kimlik doğrulamasını kullanma](#).

ibm-mq-rest-csrf-token

Bu üstbilgi ayarlanmalıdır, ancak değer boş olmak da dahil olmak üzere herhangi bir şey olabilir.

İsteğe bağlı olarak aşağıdaki üstbilgiler istekle birlikte gönderilebilir:

Kabul Et-Karakter Kümesi

Bu üstbilgi, yanıt için hangi karakter kümesinin kabul edilebilir olduğunu belirtmek için kullanılabilir. Belirtilirse, bu üstbilgi UTF-8 olarak ayarlanmalıdır.

Kabul Et-Dil

Bu üstbilgi, yanıt iletisi gövdesinde döndürülen kural dışı durumlar ya da hata iletileri için gereken dili belirtir.

İstek gövdesi biçimi

Yok.

Güvenlik Gereksinimleri

Çağıranın kimliği mqweb sunucusunda doğrulanmalıdır. MQWebAdmin ve MQWebAdminRO rolleri messaging REST API için geçerli değildir. REST API güvenliği hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenliği](#).

mqweb sunucusunda kimliği doğrulandıktan sonra, kullanıcı hem messaging REST API, hem de administrative REST API kullanabilir.

Çağıranın güvenlik birincil kullanıcılarına, belirtilen kuyruktaki iletilere göz atma yeteneği verilmelidir:

- Kaynak URL adresinin *{queueName}* kısmı tarafından belirtilen kuyruk BROWSE etkinleştirilmiş olmalıdır.
- **ALW** **MQ Appliance** Kaynak URL adresinin *{queueName}* kısmı tarafından belirtilen kuyruk için, çağıranın güvenlik birincil kullanıcılarına +GET, +INQ ve +BROWSE yetkisi verilmelidir.
- **z/OS** Kaynak URL adresinin *{queueName}* kısmı tarafından belirtilen kuyruk için UPDATE, çağıranın güvenlik birincil kullanıcılarına erişim verilmelidir.

Bu güvenlik birincil kullanıcısı, mqweb sunucusunu başlatan kullanıcı ya da mqweb sunucusunda oturum açan kullanıcı olabilir. Bağlandığınız kuyruk yöneticisi uzak bir kuyruk yöneticisiyse, güvenlik birincil kullanıcısı uzak kuyruk yöneticisi bağlantı bilgilerindeki kimlik bilgileri tarafından sağlanan kullanıcı ya da kanal güvenlik kurallarının belirlediği başka bir kullanıcı olabilir. Daha fazla bilgi için, bkz. [messaging REST API](#) tarafından kullanılan güvenlik birincil kullanıcısının belirlenmesi.

ALW AIX, Linux, and Windows' da, **setmqaut** komutunu kullanarak IBM MQ kaynaklarını kullanmak için güvenlik birincil kullanıcıları için yetki verebilirsiniz. Daha fazla bilgi için bkz. **setmqaut** (yetki verilmesi ya da iptal etmek).

z/OS z/OS işletim üzerinde bkz. [z/OS üzerinde güvenliğin ayarlanması](#).

Yanıt durumu kodları

200

İleti başarıyla alındı.

204

Kullanılabilir bir ileti yok.

400

Geçersiz veri sağlandı.

Örneğin, geçersiz bir sorgu parametresi değeri belirtildi.

401

Kimlik doğrulanmadı.

Çağıranın kimliği mqweb sunucusunda doğrulanmalı ve MQWebAdmin, MQWebAdminROya da MQWebUser rollerinden birinin ya da daha fazlasının üyesi olmalıdır. `ibm-mq-rest-csrf-token` üstbilgisi de belirtilmelidir. Daha fazla bilgi için bkz. [“Güvenlik Gereksinimleri” sayfa 2140](#).

403

Yetkili değil.

Çağıranın kimliği mqweb sunucusunda doğrulanır ve geçerli bir birincil kullanıcıyla ilişkilendirilir. Ancak, birincil kullanıcının gereken IBM MQ kaynaklarının tümüne ya da bir alt kümesine erişimi yok ya da MQWebUser rolünde yok. Gerekli erişim hakkında daha fazla bilgi için bkz. [“Güvenlik Gereksinimleri” sayfa 2140](#).

404

Kuyruk yok.

500

IBM MQ' den sunucu sorunu ya da hata kodu.

501

HTTP yanıtı oluşturulamadı.

Örneğin, alınan iletinin tipi yanlış ya da tipi doğru, ancak gövde işlenemedi.

502

İleti alışverişi sağlayıcısı gereken işlevi desteklemediğinden, yürürlükteki güvenlik birincil kullanıcısı iletiyi alamıyor. Örneğin, mqweb sunucusu sınıf yolu geçersizse.

503

Kuyruk yöneticisi çalışmıyor.

Yanıt üstbilgileri

Yanıtla birlikte aşağıdaki üstbilgiler döndürülür:

İçerik-Dil

Herhangi bir hata ya da kural dışı durum oluşması durumunda yanıt iletisinin dil tanıtıcısını belirtir. Herhangi bir hata ya da kural dışı durum koşulu için gerekli dili belirtmek üzere `Accept-Language` istek üstbilgisiyle birlikte kullanılır. İstenen dil desteklenmezse, mqweb sunucusu varsayılan değeri kullanılır.

İçerik Uzunluğu

İçerik olmadığında bile HTTP yanıt gövdesinin uzunluğunu belirtir. Değer, ileti verilerinin uzunluğunu (bayt) içerir.

İçerik Tipi

Alınan iletinin yanıt gövdesinde döndürülen içerik tipini belirtir. Başarıya ulaştığı anda değer `text/plain; charset=utf-8` olur. Herhangi bir hata ya da kural dışı durum olması durumunda, değer `application/json; charset=utf-8` olur.

REST API V1 > REST API V2 **ibm-mq-md-correlationId**

Alınan iletinin ilinti tanıtıcısını belirtir. Alınan ileti geçerli bir ilinti tanıtıcısı içeriyorsa üstbilgi döndürülür. 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak gösterilir.

Örneğin:

```
ibm-mq-md-correlationId: 414d5120514d414445562020202020202067d8bf5923582e02
```

V 9.3.0 > REST API V3 **ibm-mq-md-correlationId**

Alınan iletinin ilinti tanıtıcısını belirtir. Alınan ileti geçerli bir ilinti tanıtıcısı içeriyorsa üstbilgi döndürülür. İlinti tanıtıcısı aşağıdaki formlardan birini alabilir:

- Öneki "ID:" olan, 24 baytı gösteren, 48 karakterlik onaltılı kodlanmış dizgi. Örneğin:

```
ibm-mq-md-correlationId: ID:414d5120514d414445562020202020202067d8bf5923582e02
```

- Uygulamaya özgü bir değer. Değer uygulamaya özgü bir dizedir:

```
ibm-mq-md-correlationId: My-Custom-CorrelId
```

ibm-mq-md-süre bitimi

Alınan iletinin kalan süre bitimini belirtir. Üstbilgi aşağıdaki değerlerden biri olabilir:

sınırsız

İletinin süresi dolmaz.

Tamsayı değeri

İleti süre bitiminden önce kalan milisaniye sayısı.

REST API V1 > REST API V2 **ibm-mq-md-messageId**

IBM MQ tarafından bu iletiye ayrılan ileti tanıtıcısını belirtir. `ibm-mq-md-correlationId` üstbilgisi gibi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak gösterilir.

Örneğin:

```
ibm-mq-md-messageId: 414d5120514d414445562020202020202067d8ce5923582f07
```

V 9.3.0 > REST API V3 **ibm-mq-md-messageId**

IBM MQ tarafından bu iletiye ayrılan ileti tanıtıcısını belirtir. `ibm-mq-md-correlationId` üstbilgisinde olduğu gibi, öneki "ID:" olan 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak gösterilir.

Örneğin:

```
ibm-mq-md-messageId: ID:414d5120514d414445562020202020202067d8ce5923582f07
```

ibm-mq-md-kalıcılık

Alınan iletinin kalıcılığını belirtir. Üstbilgi aşağıdaki değerlerden biri olabilir:

nonPersistent

İleti, sistem hatalarından ya da kuyruk yöneticisi yeniden başlatmadan sonra devam etmez.

Kalıcı

İleti, sistem hatalarından ya da kuyruk yöneticisini yeniden başlatmaya devam eder.

V 9.3.0 REST API V3 ibm-mq-md-öncelik

İleti önceliğinin ayarını döndürür. Örneğin:

```
ibm-mq-md-priority: 3
```

ibm-mq-md-replyTo

Alınan iletiye ilişkin yanıt hedefini belirler. Üstbilginin biçimi, yanıt kuyruğu ve kuyruk yöneticisinin (`replyQueue@replyQmgr`) standart gösterimini kullanır.

Örneğin:

```
ibm-mq-md-replyTo: myReplyQueue@myReplyQMGR
```

V 9.3.3 ibm-mq-çözümlendi-qmgr

İstek için kullanılan kuyruk yöneticisinin adını belirtir. URL kaynağında benzersiz bir ad kullanıldıysa, bu üstbilgi o benzersiz adla ilişkilendirilmiş kuyruk yöneticisinin adını belirtir. URL kaynağında kullanılan benzersiz ad bir kuyruk yöneticisi grubuna gönderme yapıyorsa, bu üstbilgi grup içindeki hangi kuyruk yöneticisinin kullanıldığını gösterir.

V 9.3.0 REST API V3 ibm-mq-usr

İleti kullanıcı tanımlı özelliklerini döndürür. Bir iletide birden çok özellik ayarlanabilir; bu durumda, `ibm-mq-usr` yanıt üstbilgisinin iki ya da daha fazla ayrı eşgörünümü olur.

Örneğin:

```
ibm-mq-usr: myIPprop;5;short
ibm-mq-usr: mySProp;"hi";string
ibm-mq-usr: myBProp>true;boolean
```

Özelliklerin sözdizimi şöyledir:

```
ibm-mq-usr: property_name; user_value; user_type
```

özellik_adi

Belirtilen kullanıcı özelliğinin adı. Bu geçerli bir JMS özelliği adı olmalıdır.

kullanıcı_değeri

Özelliğin değeri.

kullanıcı_tipi

Özelliğin tipi:

- `boolean` (doğru/yanlış, MQBOOL)
- `byte` (8 bit tamsayı, MQINT8)
- `short` (16 bitlik tamsayı, MQINT16)
- `integer` (32 bitlik tamsayı, MQINT32)
- `long` (64 bitlik tamsayı, MQINT64)
- `float` (32 bit gerçək, MQFLOAT32)
- `double` (64 bit gerçək, MQFLOAT64)
- `string` (tırnak içine alınmış dizgi)

Yanıt gövdesi biçimi

Başarılı olursa, yanıt gövdesi alınan iletiden gelen ileti gövdesini içerir. Bir hata oluşursa, yanıt gövdesi JSON biçimli bir hata iletisi içerir. Her iki yanıt da UTF-8 kodludur. Daha fazla bilgi için bkz. [REST API hata işleme](#).

Kuyrukta uyumsuz iletiler bırakılır ve çağırana uygun bir durum kodu döndürülür. Örneğin, MQSTR ya da JMS TextMessage biçiminde olmayan bir ileti.

V 9.3.0 REST API V3 , kullanıcı tanımlı ileti özelliklerini belirleme ve ileti önceliğini iletilere dahil etme yeteneğini ekler. `ibm-mq-md-priority` ve `ibm-mq-md-usr` yanıt üstbilgileri yalnızca REST API V3 ile kullanılabilir. `ibm-mq-md-correlationId` istek üstbilgisi, REST API V3 içinde farklı bir biçime sahiptir. Üstbilgi uygulamaya özgü bir tanıttıcı olabilir ya da kodlanmış bir dizgi ID: önekini korur. `ibm-mq-md-messageId` yanıt üstbilgisi ve sorgu parametresi, REST API V3' de farklı bir biçime sahiptir, ID: önekini korur.

- “Kaynak URL” sayfa 2145
- “İsteğe bağlı sorgu parametreleri” sayfa 2146
- “İstek üstbilgileri” sayfa 2147
- “İstek gövdesi biçimi” sayfa 2147
- “Güvenlik Gereksinimleri” sayfa 2147
- “Yanıt durumu kodları” sayfa 2148
- “Yanıt üstbilgileri” sayfa 2148
- “Yanıt gövdesi biçimi” sayfa 2150
- “Örnekler” sayfa 2151

Kaynak URL

`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

İleti sistemi için bağlanılacak kuyruk yöneticisinin adını belirtir. Kuyruk yöneticisi, mqweb sunucusuyla aynı makinede olmalıdır.

V 9.3.3 IBM MQ 9.3.3' den yerel bir kuyruk yöneticisine ya da uzak bir kuyruk yöneticisine bağlanabilirsiniz. **qmgrName** için belirttiğiniz ad, mqweb sunucunuzun nasıl yapılandırıldığına bağlıdır:

- mqweb sunucunuz yalnızca yerel kuyruk yöneticilerine bağlanacak şekilde yapılandırıldıysa, kuyruk yöneticisinin adını kullanın.
- mqweb sunucunuz uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırıldıysa, uzak kuyruk yöneticisi bağlantı bilgilerinde belirtildiği şekilde kuyruk yöneticisinin benzersiz adını kullanın.

dspmqweb properties komutunu kullanarak ve **mqRestMessagingConnectionMode** özelliğini görüntüleyerek, mqweb sunucunuzun yerel kuyruk yöneticilerine ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırılıp yapılandırılmadığını belirleyebilirsiniz.

Kuyruk yöneticisi adı büyük ve küçük harfe duyarlıdır.

Kuyruk yöneticisi adında eğik çizgi, nokta imi ya da yüzde imi varsa, bu karakterler URL biçiminde kodlanmış olmalıdır:

- Eğik çizgi (/) %2Folarak kodlanmalıdır.
- Yüzde işareti (%) %25olarak kodlanmalıdır.

queueName

Bir sonraki iletinin gönderileceği kuyruğun adını belirler.

Kuyruk yerel olarak ya da yerel bir kuyruğu gösteren bir diğer ad olarak tanımlanmalıdır.

Kuyruk adı büyük ve küçük harfe duyarlıdır.

Kuyruk adı eğik çizgi ya da yüzde işareti içeriyorsa, bu karakterler URL kodlu olmalıdır:

- Eğik çizgi,/, %2Folarak kodlanmalıdır.
- Yüzde işareti,%, %25olarak kodlanmalıdır.

HTTP bağlantılarını etkinleştirirseniz, HTTPS yerine HTTP kullanabilirsiniz. HTTP' yi etkinleştirme hakkında daha fazla bilgi için bkz. [HTTP ve HTTPS kapılarını yapılandırma](#).

İsteğe bağlı sorgu parametreleri

REST API V1 > REST API V2 **correlationId=hexValue**

HTTP yönteminin, ilgili ilinti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirlenmelidir.

Örneğin:

```
../message?correlationId=414d5120514d4144455620202020202067d8bf5923582e02
```

V 9.3.0 > REST API V3 **correlationId= ID:hexValue ya da correlationId=application_specific_value**

HTTP yönteminin, ilgili ilinti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren ve "ID:" dizgisinden önce gelen 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirtilmelidir.

Örneğin:

```
../message?correlationId=ID:414d5120514d4144455620202020202067d8bf5923582e02
```

uygulama_özel_değeri

Sorgu parametresi uygulamaya özgü bir dizgi olarak belirtilebilir.

Örneğin:

```
../message?correlationId=My-Custom-CorrelId
```

REST API V1 > REST API V2 **messageId=hexValue**

HTTP yönteminin, karşılık gelen ileti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirlenmelidir.

Örneğin:

```
../message?messageId=414d5120514d4144455620202020202067d8ce5923582f07
```

V 9.3.0 > REST API V3 **messageId= Tanıtıcı:hexValue**

HTTP yönteminin, karşılık gelen ileti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren ve "ID:" dizgisinden önce gelen 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirtilmelidir.

Örneğin:

```
../message?messageId=ID:414d5120514d4144455620202020202067d8ce5923582f07
```

wait=integerValue

HTTP yönteminin, sonraki iletinin kullanılabilir olması için *integerValue* milisaniye bekleyeceğini belirtir.

integerValue

Sorgu parametresi, milisaniye süresini gösteren bir tamsayı değeri olarak belirtilmelidir. Değer üst sınırı 2147483647 'dir.

Örneğin:

```
../message?wait=120000
```

İstek üstbilgileri

Aşağıdaki üstbilgiler istekle birlikte gönderilmelidir:

Yetkilendirme

Temel kimlik doğrulaması kullanıyorsanız bu üstbilgi gönderilmelidir. Daha fazla bilgi için bkz. [REST API ile HTTP temel kimlik doğrulamasını kullanma](#).

ibm-mq-rest-csrf-token

Bu üstbilgi ayarlanmalıdır, ancak değer boş olmak da dahil olmak üzere herhangi bir şey olabilir.

İsteğe bağlı olarak aşağıdaki üstbilgiler istekle birlikte gönderilebilir:

Kabul Et-Karakter Kümesi

Bu üstbilgi, yanıt için hangi karakter kümesinin kabul edilebilir olduğunu belirtmek için kullanılabilir. Belirtilirse, bu üstbilgi UTF-8 olarak ayarlanmalıdır.

Kabul Et-Dil

Bu üstbilgi, yanıt iletisi gövdesinde döndürülen kural dışı durumlar ya da hata iletileri için gereken dili belirtir.

İstek gövdesi biçimi




Yok.

Güvenlik Gereksinimleri

Çağıranın kimliği mqweb sunucusunda doğrulanmalıdır. MQWebAdmin ve MQWebAdminRO rolleri messaging REST API için geçerli değildir. REST API güvenliği hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenliği](#).

mqweb sunucusunda kimliği doğrulandıktan sonra, kullanıcı hem messaging REST API , hem de administrative REST API kullanabilir.

Çağıranın güvenlik birincil kullanıcılarına, belirtilen kuyruktan ileti alma yeteneği verilmelidir:

- URL kaynağının *{queueName}* kısmı tarafından belirtilen kuyruk GET etkinleştirilmiş olmalıdır.
-   Kaynak URL, +GET, +INQ ve +BROWSE yetkilerinin *{queueName}* kısmı tarafından belirtilen kuyruk için, çağıranın güvenlik birincil kullanıcılarına yetki verilmelidir.
-  Kaynak URL' nin *{queueName}* kısmı tarafından belirtilen kuyruk için UPDATE, çağıranın güvenlik birincil kullanıcılarına erişim verilmelidir.

Bu güvenlik birincil kullanıcıları, mqweb sunucusunu başlatan kullanıcı ya da mqweb sunucusunda oturum açan kullanıcı olabilir. Bağlandığınız kuyruk yöneticisi uzak bir kuyruk yöneticisiyse, güvenlik birincil kullanıcıları uzak kuyruk yöneticisi bağlantı bilgilerindeki kimlik bilgileri tarafından sağlanan kullanıcı ya da kanal güvenlik kurallarının belirlediği başka bir kullanıcı olabilir. Daha fazla bilgi için, bkz. [messaging REST API tarafından kullanılan güvenlik birincil kullanıcılarının belirlenmesi](#).

ALW AIX, Linux, and Windows' da, **setmqaut** komutunu kullanarak IBM MQ kaynaklarını kullanmak için güvenlik birincil kullanıcıları için yetki verebilirsiniz. Daha fazla bilgi için bkz. **setmqaut** (yetki verilmesi ya da iptal etmek).

z/OS z/OS işletim üzerinde bkz. [z/OS üzerinde güvenliğin ayarlanması](#).

Yanıt durumu kodları

200

İleti başarıyla alındı.

204

Kullanılabilir bir ileti yok.

400

Geçersiz veri sağlandı.

Örneğin, geçersiz bir sorgu parametresi değeri belirtildi.

401

Kimlik doğrulanmadı.

Çağıranın kimliği mqweb sunucusunda doğrulanmalı ve MQWebAdmin, MQWebAdminROya da MQWebUser rollerinden birinin ya da daha fazlasının üyesi olmalıdır. `ibm-mq-rest-csrf-token` üstbilgisi de belirtilmelidir. Daha fazla bilgi için bkz. "[Güvenlik Gereksinimleri](#)" sayfa 2147.

403

Yetkili değil.

Çağıranın kimliği mqweb sunucusunda doğrulanır ve geçerli bir birincil kullanıcıyla ilişkilendirilir. Ancak, birincil kullanıcının gereken IBM MQ kaynaklarının tümüne ya da bir alt kümesine erişimi yok ya da MQWebUser rolünde yok. Gerekli erişim hakkında daha fazla bilgi için bkz. "[Güvenlik Gereksinimleri](#)" sayfa 2147.

404

Kuyruk yok.

405

Kuyruk GET engellidir.

500

IBM MQ' den sunucu sorunu ya da hata kodu.

501

HTTP yanıtı oluşturulamadı.

Örneğin, alınan iletinin tipi yanlış ya da tipi doğru, ancak gövde işlenemedi.

502

İleti alışverişi sağlayıcısı gereken işlevi desteklemediğinden, yürürlükteki güvenlik birincil kullanıcısı iletiyi alamıyor. Örneğin, mqweb sunucusu sınıf yolu geçersizse.

503

Kuyruk yöneticisi çalışmıyor.

Yanıt üstbilgileri

Yanıtla birlikte aşağıdaki üstbilgiler döndürülür:

İçerik-Dil

Herhangi bir hata ya da kural dışı durum oluşması durumunda yanıt iletisinin dil tanıtıcısını belirtir. Herhangi bir hata ya da kural dışı durum koşulu için gerekli dili belirtmek üzere `Accept-Language` istek üstbilgisiyle birlikte kullanılır. İstenen dil desteklenmezse, mqweb sunucusu varsayılan değeri kullanılır.

İçerik Uzunluğu

İçerik olmadığında bile HTTP yanıt gövdesinin uzunluğunu belirtir. Değer, ileti verilerinin uzunluğunu (bayt) içerir.

İçerik Tipi

Alınan iletinin yanıt gövdesinde döndürülen içerik tipini belirtir. Başarıya ulaştığı anda değer `text/plain; charset=utf-8` olur. Herhangi bir hata ya da kural dışı durum olması durumunda, değer `application/json; charset=utf-8` olur.

REST API V1 > REST API V2 **ibm-mq-md-correlationId**

Alınan iletinin ilinti tanıtıcısını belirtir. Alınan ileti geçerli bir ilinti tanıtıcısı içeriyorsa üstbilgi döndürülür. 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak gösterilir.

Örneğin:

```
ibm-mq-md-correlationId: 414d5120514d4144455620202020202067d8bf5923582e02
```

V 9.3.0 > REST API V3 **ibm-mq-md-correlationId**

Alınan iletinin ilinti tanıtıcısını belirtir. Alınan ileti geçerli bir ilinti tanıtıcısı içeriyorsa üstbilgi döndürülür. İlinti tanıtıcısı aşağıdaki formlardan birini alabilir:

- Öneki "ID:" olan, 24 baytı gösteren, 48 karakterlik onaltılı kodlanmış dizgi. Örneğin:

```
ibm-mq-md-correlationId: ID:414d5120514d4144455620202020202067d8bf5923582e02
```

- Uygulamaya özgü bir değer. Değer uygulamaya özgü bir dizedir:

```
ibm-mq-md-correlationId: My-Custom-CorrelId
```

ibm-mq-md-süre bitimi

Alınan iletinin kalan süre bitimini belirtir. Üstbilgi aşağıdaki değerlerden biri olabilir:

sınırsız

İletinin süresi dolmaz.

Tamsayı değeri

İleti süre bitiminden önce kalan milisaniye sayısı.

REST API V1 > REST API V2 **ibm-mq-md-messageId**

IBM MQ tarafından bu iletiye ayrılan ileti tanıtıcısını belirtir. `ibm-mq-md-correlationId` üstbilgisi gibi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak gösterilir.

Örneğin:

```
ibm-mq-md-messageId: 414d5120514d4144455620202020202067d8ce5923582f07
```

V 9.3.0 > REST API V3 **ibm-mq-md-messageId**

IBM MQ tarafından bu iletiye ayrılan ileti tanıtıcısını belirtir. `ibm-mq-md-correlationId` üstbilgisinde olduğu gibi, öneki "ID:" olan 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak gösterilir.

Örneğin:

```
ibm-mq-md-messageId: ID:414d5120514d4144455620202020202067d8ce5923582f07
```

ibm-mq-md-kalıcılık

Alınan iletinin kalıcılığını belirtir. Üstbilgi aşağıdaki değerlerden biri olabilir:

nonPersistent

İleti, sistem hatalarından ya da kuyruk yöneticisi yeniden başlatmadan sonra devam etmez.

Kalıcı

İleti, sistem hatalarından ya da kuyruk yöneticisini yeniden başlatmaya devam eder.

V 9.3.0 > REST API V3 **ibm-mq-md-öncelik**

İleti önceliğinin ayarını döndürür. Örneğin:

```
ibm-mq-md-priority: 3
```

ibm-mq-md-replyTo

Alınan iletiye ilişkin yanıt hedefini belirler. Üstbilginin biçimi, yanıt kuyruğu ve kuyruk yöneticisinin (`replyQueue@replyQmgr`) standart gösterimini kullanır.

Örneğin:

```
ibm-mq-md-replyTo: myReplyQueue@myReplyQMGR
```

V 9.3.3 ibm-mq-çözümlendi-qmgr

İstek için kullanılan kuyruk yöneticisinin adını belirtir. URL kaynağında benzersiz bir ad kullanıldıysa, bu üstbilgi o benzersiz adla ilişkilendirilmiş kuyruk yöneticisinin adını belirtir. URL kaynağında kullanılan benzersiz ad bir kuyruk yöneticisi grubuna gönderme yapıyorsa, bu üstbilgi grup içindeki hangi kuyruk yöneticisinin kullanıldığını gösterir.

V 9.3.0 REST API V3 ibm-mq-usr

İleti kullanıcı tanımlı özelliklerini döndürür. Bir iletide birden çok özellik ayarlanabilir; bu durumda, `ibm-mq-usr` yanıt üstbilgisinin iki ya da daha fazla ayrı eşgörünümü olur.

Örneğin:

```
ibm-mq-usr: myIPprop;5;short
ibm-mq-usr: mySProp;"hi";string
ibm-mq-usr: myBProp>true;boolean
```

Özelliklerin sözdizimi şöyledir:

```
ibm-mq-usr: property_name; user_value; user_type
```

özellik_adi

Belirtilen kullanıcı özelliğinin adı. Bu geçerli bir JMS özelliği adı olmalıdır.

kullanıcı_değeri

Özelliğin değeri.

kullanıcı_tipi

Özelliğin tipi:

- `boolean` (doğru/yanlış, MQBOOL)
- `byte` (8 bit tamsayı, MQINT8)
- `short` (16 bitlik tamsayı, MQINT16)
- `integer` (32 bitlik tamsayı, MQINT32)
- `long` (64 bitlik tamsayı, MQINT64)
- `float` (32 bit gerçek, MQFLOAT32)
- `double` (64 bit gerçek, MQFLOAT64)
- `string` (tırnak içine alınmış dizgi)

Yanıt gövdesi biçimi

Başarılı olursa, yanıt gövdesi alınan iletiden gelen ileti gövdesini içerir. Bir hata oluşursa, yanıt gövdesi JSON biçimli bir hata iletisi içerir. Her iki yanıt da UTF-8 kodludur. Daha fazla bilgi için bkz. [REST API hata işleme](#).

Bir iletiyi alırken yalnızca IBM MQ MQSTR ve JMS `TextMessage` biçimli iletilerin desteklendiğini unutmayın. Daha sonra, tüm iletiler eşitleme noktası altında alınır ve işlenmeyen iletiler kuyrukta bırakılır. IBM MQ kuyruğu, bu zehirli iletileri başka bir hedefe taşımak için yapılandırılabilir. Daha fazla bilgi için bkz. [IBM MQ classes for JMS](#).

- “İsteğe bağlı sorgu parametreleri” sayfa 2153
- “İstek üstbilgileri” sayfa 2154
- “İstek gövdesi biçimi” sayfa 2154
- “Güvenlik Gereksinimleri” sayfa 2154
- “Yanıt durumu kodları” sayfa 2154
- “Yanıt üstbilgileri” sayfa 2155
- “Yanıt gövdesi biçimi” sayfa 2155
- “Örnekler” sayfa 2156

Kaynak URL

`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

İleti sistemi için bağlanılacak kuyruk yöneticisinin adını belirtir. Kuyruk yöneticisi, mqweb sunucusuyla aynı makinede olmalıdır.

V 9.3.3 IBM MQ 9.3.3' den yerel bir kuyruk yöneticisine ya da uzak bir kuyruk yöneticisine bağlanabilirsiniz. **qmgrName** için belirttiğiniz ad, mqweb sunucunuzun nasıl yapılandırıldığına bağlıdır:

- mqweb sunucunuz yalnızca yerel kuyruk yöneticilerine bağlanacak şekilde yapılandırıldıysa, kuyruk yöneticisinin adını kullanın.
- mqweb sunucunuz uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırıldıysa, uzak kuyruk yöneticisi bağlantı bilgilerinde belirtildiği şekilde kuyruk yöneticisinin benzersiz adını kullanın.

dspmqweb properties komutunu kullanarak ve **mqRestMessagingConnectionMode** özelliğini görüntüleyerek, mqweb sunucunuzun yerel kuyruk yöneticilerine ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırılıp yapılandırılmadığını belirleyebilirsiniz.

Kuyruk yöneticisi adı büyük ve küçük harfe duyarlıdır.

Kuyruk yöneticisi adında eğik çizgi, nokta imi ya da yüzde imi varsa, bu karakterler URL biçiminde kodlanmış olmalıdır:

- Eğik çizgi (/) %2Folarak kodlanmalıdır.
- Yüzde işareti (%) %25olarak kodlanmalıdır.

queueName

İletilere göz atılacak kuyruğun adını belirler.

Kuyruk, yerel olarak ya da yerel bir kuyruğu gösteren bir diğer ad olarak tanımlanmalıdır.

Kuyruk adı büyük ve küçük harfe duyarlıdır.

Kuyruk adı eğik çizgi ya da yüzde işareti içeriyorsa, bu karakterler URL kodlu olmalıdır:

- Eğik çizgi,/, %2Folarak kodlanmalıdır.
- Yüzde işareti,%, %25olarak kodlanmalıdır.

HTTP bağlantılarını etkinleştirirseniz, HTTPS yerine HTTP kullanabilirsiniz. HTTP' yi etkinleştirme hakkında daha fazla bilgi için bkz. [HTTP ve HTTPS kapılarını yapılandırma](#).

İsteğe bağlı sorgu parametreleri

REST API V2 correlationId=hexValue

HTTP yönteminin, ilgili ilinti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirlenmelidir.

Örneğin:

```
../messageList?correlationId=414d5120514d41444556202020202067d8bf5923582e02
```

V 9.3.0 REST API V3 correlationId= ID:hexValue ya da correlationId=application_specific_value

HTTP yönteminin, ilgili ilinti tanıtıcısına sahip iletilerin bir listesini döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren ve "ID:" dizgisinden önce gelen 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirtilmelidir.

Örneğin:

```
../message?correlationId=ID:414d5120514d41444556202020202067d8bf5923582e02
```

uygulama_özel_değeri

Sorgu parametresi uygulamaya özgü bir dizgi olarak belirtilebilir.

Örneğin:

```
../message?correlationId=My-Custom-CorrelId
```

REST API V1 REST API V2 messageId=hexValue

HTTP yönteminin, karşılık gelen ileti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirlenmelidir.

Örneğin:

```
../message?messageId=414d5120514d41444556202020202067d8ce5923582f07
```

V 9.3.0 REST API V3 messageId= Tanıtıcı:hexValue

HTTP yönteminin, karşılık gelen ileti tanıtıcısına sahip sonraki iletiyi döndürdüğünü belirtir.

hexValue

Sorgu parametresi, 24 baytı gösteren ve "ID:" dizgisinden önce gelen 48 karakterlik onaltılı kodlanmış bir dizgi olarak belirtilmelidir.

Örneğin:

```
../message?messageId=ID:414d5120514d41444556202020202067d8ce5923582f07
```

limit=integerValue

HTTP yöntemi yanıt gövdesinin integerValue JSON öğeleriyle sınırlı olduğunu belirtir.

integerValue

Sorgu parametresi, JSON yanıt gövdesinde bulunan öğe sayısı üst sınırını gösteren bir tamsayı değeri olarak belirtilmelidir.

Varsayılan değer 10 ve üst sınır değeri 2147483647 'dir.

Örneğin:

```
../messageList?limit=250
```

İstek üstbilgileri

Aşağıdaki üstbilgiler istekle birlikte gönderilmelidir:

Yetkilendirme

Temel kimlik doğrulaması kullanıyorsanız bu üstbilgi gönderilmelidir. Daha fazla bilgi için bkz. [REST API ile HTTP temel kimlik doğrulamasını kullanma](#).

ibm-mq-rest-csrf-token

Bu üstbilgi ayarlanmalıdır, ancak değer boş olmak da dahil olmak üzere herhangi bir şey olabilir.

İsteğe bağlı olarak aşağıdaki üstbilgiler istekle birlikte gönderilebilir:

Kabul Et-Karakter Kümesi

Bu üstbilgi, yanıt için hangi karakter kümesinin kabul edilebilir olduğunu belirtmek için kullanılabilir. Belirtilirse, bu üstbilgi UTF-8 olarak ayarlanmalıdır.

Kabul Et-Dil

Bu üstbilgi, yanıt iletisi gövdesinde döndürülen kural dışı durumlar ya da hata iletileri için gereken dili belirtir.

İstek gövdesi biçimi




Yok.

Güvenlik Gereksinimleri


Çağrının kimliği mqweb sunucusunda doğrulanmalıdır. MQWebAdmin ve MQWebAdminRO rolleri messaging REST API için geçerli değildir. REST API güvenliği hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenliği](#).


mqweb sunucusunda kimliği doğrulandıktan sonra, kullanıcı hem messaging REST API, hem de administrative REST API kullanabilir.

Çağrının güvenlik birincil kullanıcılarına, belirtilen kuyruktaki iletilere göz atma yeteneği verilmelidir:

- Kaynak URL adresinin `{queueName}` kısmı tarafından belirtilen kuyruk BROWSE etkinleştirilmiş olmalıdır.
-   Kaynak URL adresinin `{queueName}` kısmı tarafından belirtilen kuyruk için, çağrının güvenlik birincil kullanıcılarına +GET, +INQ ve +BROWSE yetkisi verilmelidir.
-  Kaynak URL adresinin `{queueName}` kısmı tarafından belirtilen kuyruk için UPDATE, çağrının güvenlik birincil kullanıcılarına erişim verilmelidir.

Bu güvenlik birincil kullanıcı, mqweb sunucusunu başlatan kullanıcı ya da mqweb sunucusunda oturum açan kullanıcı olabilir. Bağlandığınız kuyruk yöneticisi uzak bir kuyruk yöneticisiyse, güvenlik birincil kullanıcı uzak kuyruk yöneticisi bağlantı bilgilerindeki kimlik bilgileri tarafından sağlanan kullanıcı ya da kanal güvenlik kurallarının belirlediği başka bir kullanıcı olabilir. Daha fazla bilgi için, bkz. [messaging REST API tarafından kullanılan güvenlik birincil kullanıcısının belirlenmesi](#).

 AIX, Linux, and Windows' da, **setmqaut** komutunu kullanarak IBM MQ kaynaklarını kullanmak için güvenlik birincil kullanıcıları için yetki verebilirsiniz. Daha fazla bilgi için bkz. [setmqaut](#) (yetki verilmesi ya da iptal etmek).

 z/OS işletim üzerinde bkz. [z/OS üzerinde güvenliğin ayarlanması](#).

Yanıt durumu kodları

200

İleti listesi başarıyla alındı.

400

Geçersiz veri sağlandı.

Örneğin, geçersiz bir sorgu parametresi değeri belirtildi.

401

Kimlik doğrulanmadı.

Çağıranın kimliği mqweb sunucusunda doğrulanmalı ve MQWebAdmin, MQWebAdminROya da MQWebUser rollerinden birinin ya da daha fazlasının üyesi olmalıdır. `ibm-mq-rest-csrf-token` üstbilgisi de belirtilmelidir. Daha fazla bilgi için bkz. "[Güvenlik Gereksinimleri](#)" sayfa 2154.

403

Yetkili değil.

Çağıranın kimliği mqweb sunucusunda doğrulanır ve geçerli bir birincil kullanıcıyla ilişkilendirilir. Ancak, birincil kullanıcının gereken IBM MQ kaynaklarının tümüne ya da bir alt kümesine erişimi yok ya da MQWebUser rolünde yok. Gerekli erişim hakkında daha fazla bilgi için bkz. "[Güvenlik Gereksinimleri](#)" sayfa 2154.

404

Kuyruk yok.

500

IBM MQ' den sunucu sorunu ya da hata kodu.

501

HTTP yanıtı oluşturulamadı.

Örneğin, alınan iletinin tipi yanlış ya da tipi doğru, ancak gövde işlenemedi.

502

İleti alışverişi sağlayıcısı gereken işlevi desteklemediğinden, yürürlükteki güvenlik birincil kullanıcısı iletiyi alamıyor. Örneğin, mqweb sunucusu sınıf yolu geçersizse.

503

Kuyruk yöneticisi çalışmıyor.

Yanıt üstbilgileri

İçerik-Dil

Herhangi bir hata ya da kural dışı durum oluşması durumunda yanıt iletisinin dil tanıtıcısını belirtir. Herhangi bir hata ya da kural dışı durum koşulu için gerekli dili belirtmek üzere `Accept-Language` istek üstbilgisiyle birlikte kullanılır. İstenen dil desteklenmezse, mqweb sunucusu varsayılan değeri kullanılır.

İçerik Uzunluğu

İçerik olmadığında bile HTTP yanıt gövdesinin uzunluğunu belirtir. Değer, ileti verilerinin uzunluğunu bayt cinsinden içerir.

İçerik Tipi

Yanıt gövdesinin tipini belirtir. Değer: `application/json; charset=utf-8`.

V 9.3.3

ibm-mq-çözümlendi-qmgr

İstek için kullanılan kuyruk yöneticisinin adını belirtir. URL kaynağında benzersiz bir ad kullanıldıysa, bu üstbilgi o benzersiz adla ilişkilendirilmiş kuyruk yöneticisinin adını belirtir. URL kaynağında kullanılan benzersiz ad bir kuyruk yöneticisi grubuna gönderme yapıyorsa, bu üstbilgi grup içindeki hangi kuyruk yöneticisinin kullanıldığını gösterir.

Yanıt gövdesi biçimi

Başarılı olursa, yanıt gövdesi UTF-8 ile kodlanmış bir yanıttır. Yanıt, messagesadlı tek bir JSON dizisi içeren bir dış JSON nesnesi içeriyor. Dizideki her öge, kuyruktaki bir iletiye ilişkin bilgi içeren bir JSON nesnesidir. Her öge aşağıdaki öznitelikleri içerir:

REST API V1

REST API V2

correlationId

İletinin ilinti tanıtıcısını belirtir. İleti geçerli bir ilinti tanıtıcısı içeriyorsa değer döndürülür.

dspmqweb properties komutunu kullanarak ve **mqRestMessagingConnectionMode** özelliğini görüntüleyerek, mqweb sunucunuzun yerel kuyruk yöneticilerine ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırılıp yapılandırılmadığını belirleyebilirsiniz.

Ad, büyük ve küçük harfe duyarlıdır.

Ad eğik çizgi, nokta ya da yüzde işareti içeriyorsa, bu karakterler URL biçiminde kodlanmış olmalıdır:

- Eğik çizgi %2Folarak kodlanmalıdır.
- Bir dönem %2Eolarak kodlanmalıdır.
- Yüzde işareti %25olarak kodlanmalıdır.

topicString

İletinin yayınlanacağı konu dizgisini belirtir.

Konu dizgisi büyük ve küçük harfe duyarlıdır. Konu dizgisi, eğik çizgi sınırlayıcısıyla ayrılmış birden çok konu düzeyi içerebilir.

Konu dizgisi bir yüzde işareti, nokta ya da soru işareti içeriyorsa, bu karakterler URL kodlu olmalıdır:

- Yüzde işareti %25olarak kodlanmalıdır.
- Bir dönem %2Eolarak kodlanmalıdır.
- Bir soru işareti %3Folarak kodlanmalıdır.

Konu dizesi eğik çizgiyle başlıyorsa ya da biterse, %2Folarak kodlanmalıdır.

Örneğin, konu dizgisine yayınlamak için:

- `sport/football` kuyruk yöneticisinde MY.QMGR, aşağıdaki URL'yi kullanırsınız:

```
https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/MY%2EQMGR/topic/sport/football/message
```

- `/sport/football` kuyruk yöneticisinde MY.QMGR, aşağıdaki URL'yi kullanırsınız:

```
https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/MY%2EQMGR/topic/%2Fsport/football/message
```

HTTP bağlantılarını etkinleştirirseniz, HTTPS yerine HTTP kullanabilirsiniz. HTTP'yi etkinleştirme hakkında daha fazla bilgi için bkz. [HTTP ve HTTPS kapılarını yapılandırma](#).

İstek üstbilgileri

Aşağıdaki üstbilgiler istekle birlikte gönderilmelidir:

Yetkilendirme

Temel kimlik doğrulaması kullanıyorsanız bu üstbilgi gönderilmelidir. Daha fazla bilgi için bkz. [REST API ile HTTP temel kimlik doğrulamasını kullanma](#).

İçerik Tipi

Bu üstbilgi aşağıdaki değerlerden biriyle gönderilmelidir:

- `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

Bu üstbilgi ayarlanmalıdır, ancak değer boş olmak da dahil olmak üzere herhangi bir şey olabilir.

İsteğe bağlı olarak aşağıdaki üstbilgiler istekle birlikte gönderilebilir:

Kabul Et-Dil

Bu üstbilgi, yanıt iletisi gövdesinde döndürülen kural dışı durumlar ya da hata iletileri için gereken dili belirtir.

ibm-mq-md-süre bitimi

Bu üstbilgi, oluşturulan ileti için süre bitimini ayarlar. Bir iletinin süre bitimi, iletinin kuyruk yöneticisine ulaştığı zamandan başlar. Sonuç olarak ağ gecikme süresi yoksayılr. Üstbilgi aşağıdaki değerlerden biri olarak belirtilmelidir:

sınırsız

İletinin süresi dolmaz.

Bu değer varsayılan değerdir.

Tamsayı değeri

İleti süre bitiminden önce milisaniye.

0-9999999999900 aralığıyla sınırlıdır.

ibm-mq-md-kalıcılık

Bu üstbilgi, oluşturulan ileti için kalıcılığı ayarlar. Üstbilgi aşağıdaki değerlerden biri olarak belirtilmelidir:

nonPersistent

İleti, sistem hatalarından ya da kuyruk yöneticisi yeniden başlatmadan sonra devam etmez.

Bu değer varsayılan değerdir.

Kalıcı

İleti, sistem hatalarından ya da kuyruk yöneticisini yeniden başlatmaya devam eder.

V 9.3.0

REST API V3

ibm-mq-md-öncelik

Bu üstbilgi, oluşturulan iletinin önceliğini ayarlar. Üstbilgi aşağıdaki değerlerden biri olarak belirtilmelidir:

asDestination

İleti, temel IBM MQ kuyruk nesnesinin DEFPTY özniteliğinde belirtilen önceliği kullanır.

Tamsayı değeri

Gerçek önceliği 0-9 aralığında bir tamsayı olarak belirtin.

Örneğin:

```
ibm-mq-md-priority: asDestination
```

ibm-mq-md-replyTo

Bu üstbilgi, yaratılan ileti için yanıt hedefini ayarlar. Üstbilginin biçimi, yanıt kuyruğunun ve isteğe bağlı bir kuyruk yöneticisinin sağlanmasına ilişkin standart gösterimi kullanır: `replyQueue[@replyQmgr]`

Örneğin:

```
ibm-mq-md-replyTo: myReplyQueue@myReplyQMGR
```

V 9.3.0

REST API V3

ibm-mq-usr

İstek iletisi kullanıcı tanımlı özelliklerini ayarlayın. Bir iletide birden çok özellik ayarlanabilir. Tek bir `ibm-mq-usr` istek üstbilgisinde virgülle ayrılmış birden çok özellik belirtebilir ya da `ibm-mq-usr` istek üstbilgisinin iki ya da daha fazla ayrı örneğini kullanabilirsiniz.

Örneğin:

```
ibm-mq-usr: myIPprop;5;short
ibm-mq-usr: mySPprop;"hi";string
ibm-mq-usr: myBPprop>true;boolean
ibm-mq-usr: myA;5;byte,myB;-10;integer
```

Özelliklerin sözdizimi şöyledir:

```
ibm-mq-usr: property_name; user_value; user_type
```

özellik_adi

Belirtilen kullanıcı özelliğinin adı. Bu geçerli bir JMS özelliği adı olmalıdır.

kullanıcı_değeri

Özelliğın değeri.

kullanıcı_tipi

Özelliğın tipi:

- boolean (doğru/yanlış, MQBOOL)
- byte (8 bit tamsayı, MQINT8)
- short (16 bitlik tamsayı, MQINT16)
- integer (32 bitlik tamsayı, MQINT32)
- long (64 bitlik tamsayı, MQINT64)
- float (32 bit gerçek, MQFLOAT32)
- double (64 bit gerçek, MQFLOAT64)
- string (tırnak içine alınmış dizgi)

İstek gövdesi biçimi

İstek gövdesi metin olmalı ve UTF-8 kodlamasını kullanmalıdır. Belirli bir metin yapısı gerekli değildir. İstek gövdesi metnini içeren MQSTR biçimli bir ileti oluşturulur ve belirtilen konuda yayınlanır.

V 9.3.0 **REST API V3** REST API V3 kullanıcı tanımlı özellikleri ya da uygulamaya özgü ilinti tanıtıcısı özellikleri kullanılırsa, istek gövdesi metnini içeren JMS TextMessage biçimli bir ileti yaratılır ve belirtilen kuyruğa yerleştirilir.

Daha fazla bilgi için bkz. [örnekler](#).

Güvenlik Gereksinimleri

Çağıranın kimliği mqweb sunucusunda doğrulanmalıdır. MQWebAdmin ve MQWebAdminRO rolleri messaging REST API için geçerli değildir. REST API güvenliği hakkında daha fazla bilgi için bkz. [IBM MQ Console](#) ve REST API güvenlik.

mqweb sunucusunda kimliği doğrulandıktan sonra, kullanıcı hem messaging REST API , hem de administrative REST API kullanabilir.

Çağıranın güvenlik birincil kullanıcıasına, belirtilen konuda ileti yayınlama yeteneği verilmelidir:

- URL kaynağının *{topicString}* kısmı tarafından belirtilen konu PUBLISH etkinleştirilmiş olmalıdır.
- **MQ Appliance** **ALW** Kaynak URL' nin *{topicString}* kısmı tarafından belirtilen konu için, +PUB yetkisi çağıranın güvenlik birincil kullanıcıasına verilmelidir.
- **z/OS** URL kaynağının *{topicString}* kısmı tarafından belirtilen konu için, UPDATE erişimi çağıranın güvenlik birincil kullanıcıasına verilmelidir.

Bu güvenlik birincil kullanıcı, mqweb sunucusunu başlatan kullanıcı ya da mqweb sunucusunda oturum açan kullanıcı olabilir. Bağlandığınız kuyruk yöneticisi uzak bir kuyruk yöneticisiyse, güvenlik birincil kullanıcı uzak kuyruk yöneticisi bağlantı bilgilerindeki kimlik bilgileri tarafından sağlanan kullanıcı ya da kanal güvenlik kurallarının belirlediği başka bir kullanıcı olabilir. Daha fazla bilgi için, bkz. [messaging REST API](#) tarafından kullanılan güvenlik birincil kullanıcıasının belirlenmesi.

ALW AIX, Linux, and Windows' da, **setmqaut** komutunu kullanarak IBM MQ kaynaklarını kullanmak için güvenlik birincil kullanıcıları için yetki verebilirsiniz. Daha fazla bilgi için bkz. **setmqaut** (yetki verilmesi ya da iptal etmek).

z/OS z/OS işletim üzerinde bkz. [z/OS üzerinde güvenliğın ayarlanması](#).

messaging REST API ile Gelişmiş İleti Güvenliği (Advanced Message Security; AMS) kullanıyorsanız, tüm iletilerin iletiyi postalayan kullanıcının bağlamı değil, mqweb sunucusunun bağlamı kullanılarak şifrelendiğini unutmayın.

Yanıt durumu kodları

201

İleti başarıyla oluşturuldu ve yayınlandı.

400

Geçersiz veri sağlandı.

Örneğin, geçersiz bir istek üstbilgisi değeri belirtildi.

401

Kimlik doğrulanmadı.

Çağıranın kimliği mqweb sunucusunda doğrulanmalı ve MQWebAdmin, MQWebAdminR0ya da MQWebUser rollerinden birinin ya da daha fazlasının üyesi olmalıdır. `ibm-mq-rest-csrf-token` üstbilgisi de belirtilmelidir. Daha fazla bilgi için bkz. "[Güvenlik Gereksinimleri](#)" sayfa 2160.

403

Yetkili değil.

Çağıranın kimliği mqweb sunucusunda doğrulanır ve geçerli bir birincil kullanıcıyla ilişkilendirilir. Ancak, birincil kullanıcının gereken IBM MQ kaynaklarının tümüne ya da bir alt kümesine erişimi yok ya da MQWebUser rolünde yok. Gerekli erişim hakkında daha fazla bilgi için bkz. "[Güvenlik Gereksinimleri](#)" sayfa 2160.

404

Kuyruk yöneticisi yok.

405

Konu: PUBLISH engellenmiş.

415

İleti üstbilgisi ya da gövdesi desteklenmeyen bir ortam tipidir.

Örneğin, Content-Type üstbilgisi desteklenmeyen bir ortam tipine ayarlanır.

500

IBM MQ' den sunucu sorunu ya da hata kodu.

502

İleti alışverişi sağlayıcısı gereken işlevi desteklemediğinden, yürürlükteki güvenlik birincil kullanıcısı iletiyi yayınlamıyor. Örneğin, mqweb sunucusu sınıf yolu geçersizse.

503

Kuyruk yöneticisi çalışmıyor.

Yanıt üstbilgileri

Yanıtla birlikte aşağıdaki üstbilgiler döndürülür:

İçerik-Dil

Herhangi bir hata ya da kural dışı durum oluşması durumunda yanıt iletilisinin dil tanıtıcısını belirtir. Herhangi bir hata ya da kural dışı durum koşulu için gerekli dili belirtmek üzere `Accept-Language` istek üstbilgisiyle birlikte kullanılır. İstenen dil desteklenmezse, mqweb sunucusu varsayılan değeri kullanılır.

İçerik Uzunluğu

İçerik olmadığında bile HTTP yanıt gövdesinin uzunluğunu belirtir. Başarıdan sonra değer sıfırdır.

İçerik Tipi

Yanıt gövdesinin tipini belirtir. Başarıya ulaştığı anda değer `text/plain; charset=utf-8` olur. Herhangi bir hata ya da kural dışı durum olması durumunda, değer `application/json; charset=utf-8` olur.

V 9.3.3

ibm-mq-çözümlendi-qmgr

İstek için kullanılan kuyruk yöneticisinin adını belirtir. URL kaynağında benzersiz bir ad kullanıldıysa, bu üstbilgi o benzersiz adla ilişkilendirilmiş kuyruk yöneticisinin adını belirtir. URL kaynağında kullanılan benzersiz ad bir kuyruk yöneticisi grubuna gönderme yapıyorsa, bu üstbilgi grup içindeki hangi kuyruk yöneticisinin kullanıldığını gösterir.

Yanıt gövdesi biçimi

İleti başarıyla yayınlandıysa yanıt gövdesi boş olur. Bir hata oluşursa, yanıt gövdesi bir hata iletisi içerir. Daha fazla bilgi için bkz. [REST API hata işleme](#).

Örnekler

Aşağıdaki örnek, mquserparolasıyla mquser adlı bir kullanıcıda oturum açar. cURL' de oturum açma isteği aşağıdaki Windows örneğine benzer. LTPA belirteci, -c işareti kullanılarak cookiejar.txt dosyasında saklanır:

```
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
```

Kullanıcı oturum açtıktan sonra, LTPA simgesi ve ibm-mq-rest-csrf-token HTTP üstbilgisi, daha fazla isteğin kimliğini doğrulamak için kullanılır. ibm-mq-rest-csrf-token token_value , boşluk da içinde olmak üzere herhangi bir değer olabilir.

- Aşağıdaki Windows cURL örneği, varsayılan seçenekleri kullanarak myTopic kuyruk yöneticisinde QM1konu dizgisine bir ileti yayınlr. İleti "Merhaba Dünya!"metnini içerir:

```
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!"
```

- Aşağıdaki Windows cURL örneği, myTopic/thisTopic kuyruk yöneticisinde QM1konu dizgisine 2 dakika süre bitimiyle kalıcı bir ileti yayınlr. İleti "Merhaba Dünya!"metnini içerir:

```
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!"
```

Özel notlar

Bu belge, ABD'de kullanıma sunulan ürünler ve hizmetler için hazırlanmıştır.

IBM, bu belgede sözü edilen ürün, hizmet ya da özellikleri diğer ülkelerde kullanıma sunmayabilir. Bulduğunuz yerde kullanıma sunulan ürün ve hizmetleri yerel IBM müşteri temsilcisinden ya da çözüm ortağınızdan öğrenebilirsiniz. Bir IBM ürün, program ya da hizmetine gönderme yapılması, açık ya da örtük olarak yalnızca o IBM ürünü, programı ya da hizmetinin kullanılabilirliğini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı ürün, program ya da hizmetlerle gerçekleştirilen işlemlerin değerlendirilmesi ve doğrulanması kullanıcının sorumluluğundadır.

IBM'in, bu belgedeki konularla ilgili patentleri ya da patent başvuruları olabilir. Bu belgenin size verilmiş olması, patentlerin izinsiz kullanım hakkının da verildiği anlamına gelmez. Lisansla ilgili sorularınızı aşağıdaki adrese yazabilirsiniz:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Çift byte (DBCS) bilgilerle ilgili lisans soruları için, ülkenizdeki IBM'in Fikri Haklar (Intellectual Property) bölümüyle bağlantı kurun ya da sorularınızı aşağıda adrese yazın:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japonya

İzleyen paragraf, bu tür kayıt ve koşulların, yasalarıyla bağdaşmadığı ülkeler ya da bölgeler için geçerli değildir: IBM BU YAYINI, "OLDUĞU GİBİ", HİÇBİR KONUDA AÇIK YA DA ÖRTÜK GARANTİ VERMEKSİZİN SAĞLAMAKTADIR; TİCARİ KULLANIMA UYGUNLUK AÇISINDAN HER TÜRLÜ GARANTİ VE BELİRLİ BİR AMACA UYGUNLUK İDDİASI AÇIKÇA REDDEDİLİR. Bazı ülkeler bazı işlemlerde garantinin açık ya da örtük olarak reddedilmesine izin vermez; dolayısıyla, bu bildirim sizin için geçerli olmayabilir.

Bu yayın teknik yanlışlar ya da yazım hataları içerebilir. Buradaki bilgiler üzerinde düzenli olarak değişiklik yapılmaktadır; söz konusu değişiklikler sonraki basımlara yansıtılacaktır. IBM, önceden bildirimde bulunmaksızın, bu yayında açıklanan ürünler ve/ya da programlar üzerinde iyileştirmeler ve/ya da değişiklikler yapabilir.

Bu belgede IBM dışı Web sitelerine yapılan göndermeler kullanıcıya kolaylık sağlamak içindir ve bu Web sitelerinin onaylanması anlamına gelmez. Bu Web sitelerinin içerdiği malzeme, bu IBM ürününe ilişkin malzemenin bir parçası değildir ve bu tür Web sitelerinin kullanılmasının sorumluluğu size aittir.

IBM'e bilgi ilettiğinizde, IBM bu bilgileri size karşı hiçbir yükümlülük almaksızın uygun gördüğü yöntemlerle kullanabilir ya da dağıtabilir.

(i) Bağımsız olarak yaratılan programlarla, bu program da içinde olmak üzere diğer programlar arasında bilgi değiş tokuşuna ve (ii) değiş tokuş edilen bilginin karşılıklı kullanımına olanak sağlamak amacıyla bu program hakkında bilgi sahibi olmak isteyen lisans sahipleri şu adrese yazabilirler:

IBM Corporation
Yazılım Birlikte Çalışabilirlik Koordinatörü, Bölüm 49XA
3605 Karayolu 52 N
Rochester, MN 55901
U.S.A.

Bu tür bilgiler, ilgili kayıt ve koşullar altında ve bazı durumlarda bedelli olarak edinilebilir.

Bu belgede açıklanan lisanslı program ve bu programla birlikte kullanılacak tüm lisanslı malzeme, IBM tarafından IBM Müşteri Sözleşmesi, IBM Uluslararası Program Lisans Sözleşmesi ya da taraflar arasında yapılan herhangi bir eşdeğer sözleşmenin koşulları kapsamında sağlanır.

Burada belirtilen performans verileri denetimli bir ortamda elde edilmiştir. Bu nedenle, başka işletim ortamlarında çok farklı sonuçlar alınabilir. Bazı ölçümler geliştirilme düzeyindeki sistemlerde yapılmıştır ve bu ölçümlerin genel kullanıma sunulan sistemlerde de aynı olacağı garanti edilemez. Ayrıca, bazı sonuçlar öngörü yöntemiyle elde edilmiş olabilir. Dolayısıyla, gerçek sonuçlar farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları için geçerli verileri kendileri doğrulamalıdır.

IBM dışı ürünlerle ilgili bilgiler, bu ürünleri sağlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma açık diğer kaynaklardan alınmıştır. IBM bu ürünleri sinamamıştır ve IBM dışı ürünlerle ilgili performans doğruluğu, uyumluluk gibi iddiaları doğrulayamaz. IBM dışı ürünlerin yeteneklerine ilişkin sorular, bu ürünleri sağlayan firmalara yöneltilmelidir.

IBM'in gelecekteki yönelim ve kararlarına ilişkin tüm bildirimler değişebilir ve herhangi bir duyuruda bulunulmadan bunlardan vazgeçilebilir; bu yönelim ve kararlar yalnızca amaç ve hedefleri gösterir.

Bu belge, günlük iş ortamında kullanılan veri ve raporlara ilişkin örnekler içerir. Örneklerin olabildiğince açıklayıcı olması amacıyla kişi, şirket, marka ve ürün adları belirtilmiş olabilir. Bu adların tümü gerçek dışıdır ve gerçek iş ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tümüyle rastlantıdır.

YAYIN HAKKI LİSANSI:

Bu belge, çeşitli işletim platformlarında programlama tekniklerini gösteren, kaynak dilde yazılmış örnek uygulama programları içerir. Bu örnek programları, IBM'e herhangi bir ödemede bulunmadan, örnek programların yazıldığı işletim altyapısına ilişkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliştirilmesi, kullanılması, pazarlanması ya da dağıtılması amacıyla herhangi bir biçimde kopyalayabilir, değiştirebilir ve dağıtabilirsiniz. Bu örnekler her koşul altında tüm ayrıntılarıyla sinanmamıştır. Dolayısıyla, IBM bu programların güvenilirliği, bakım yapılabilirliği ya da işlevleri konusunda açık ya da örtük güvence veremez.

Bu bilgileri elektronik kopya olarak görüntülediyseniz, fotoğraflar ve renkli resimler görünmeyebilir.

Programlama arabirimi bilgileri

Sağlandıysa, programlama arabirimi bilgileri, bu programla birlikte kullanılmak üzere uygulama yazılımı oluşturmanıza yardımcı olmak amacıyla hazırlanmıştır.

Bu kitapta, müşterinin WebSphere MQ hizmetlerini elde etmek üzere program yazmasına olanak sağlayan amaçlanan programlama arabirimlerine ilişkin bilgiler yer alır.

Ancak, bu bilgiler tanılama, değiştirme ve ayarlama bilgilerini de içerebilir. Tanılama, değiştirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

Önemli: Bu tanılama, değiştirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu bilgiler değişebilir.

Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dünya çapında birçok farklı hukuk düzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının güncel bir listesine Web üzerinde "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml (Telif hakkı ve ticari marka bilgileri) başlıklı konudan ulaşılabilir. Diğer ürün ve hizmet adları IBM'in veya diğer şirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation firmasının ABD'de ve/ya da diğer ülkelerdeki markalarıdır.

UNIX, The Open Group şirketinin ABD ve diğer ülkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diğer ülkelerdeki tescilli ticari markasıdır.

Bu ürün, Eclipse Project (<https://www.eclipse.org/>) tarafından geliştirilen yazılımları içerir.

Java ve Java tabanlı tüm markalar ve logolar, Oracle firmasının ve/ya da iřtiraklerinin markaları ya da tescilli markalarıdır.



Parça numarası:

(1P) P/N: