

9.4

Administrando o IBM MQ

IBM

Nota

Antes de usar estas informações e o produto que elas suportam, leia as informações em [“Avisos” na página 583](#).

Esta edição se aplica à versão 9, liberação 4 do IBM® MQ e a todas as liberações e modificações subsequentes, até que seja indicado de outra forma em novas edições

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

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

Índice

Administrando.....	7
Maneiras de administrar gerenciadores de filas e recursos associados do IBM MQ.....	8
Administrando o IBM MQ for Multiplatforms usando comandos de controle.....	10
Administrando o IBM MQ usando comandos MQSC.....	12
Sintaxe do comando MQSC.....	13
Sintaxe do arquivo de entrada MQSC.....	16
Executando comandos MQSC interativamente em runmqsc	18
Executando comandos MQSC a partir de arquivos de texto em runmqsc	23
Configuração automática por meio de um script MQSC na inicialização.....	24
Automatizando a administração do IBM MQ usando comandos PCF.....	25
Introdução aos formatos de comando programável do IBM MQ.....	26
Usando o MQAI para simplificar o uso de PCFs.....	38
Administração usando o REST API.....	75
Introdução ao administrative REST API.....	75
Administração remota usando a REST API.....	80
Registros de data e hora da REST API.....	85
REST API manipulação de erros.....	85
Descoberta de REST API.....	88
Suporte ao idioma nacional da REST API.....	89
Versões do REST API.....	90
Administração usando o IBM MQ Console.....	92
Introdução ao IBM MQ Console.....	92
Tour rápido do IBM MQ Console.....	94
Configurações IBM MQ Console.....	121
Administração usando o IBM MQ Explorer.....	121
O que Você Pode Fazer com IBM MQ Explorer.....	122
Configurando o IBM MQ Explorer.....	123
Usando o aplicativo IBM MQ Taskbar (Windows somente).....	129
O aplicativo monitor de alertas do IBM MQ (Windows somente).....	130
Trabalhando com objetos locais do IBM MQ.....	130
Trabalhando com Gerenciadores de Fila.....	130
Parando canais MQI.....	141
Trabalhando com Filas Locais.....	141
Trabalhando com Filas Remotas.....	151
Trabalhando com Filas de Alias.....	154
Trabalhando com filas modelo.....	156
Trabalhando com filas de mensagens não entregues.....	157
Trabalhando com tópicos administrativos.....	178
Trabalhando com assinaturas.....	181
Trabalhando com Serviços.....	186
Gerenciando os Objetos para Acionamento.....	193
Usando o utilitário dmpmqmsg entre dois sistemas.....	196
Trabalhando com objetos remotos do IBM MQ.....	199
Configurando Gerenciadores de Filas para Administração Remota.....	200
Gerenciando o servidor de comandos para administração remota.....	204
Emitindo comandos MQSC em um gerenciador de filas remotas.....	205
Conversão de dados entre conjuntos de caracteres codificados.....	207
Administrando Managed File Transfer.....	211
Iniciando um agente MFT.....	213
Listando Agentes do MFT.....	218
Parando um agente MFT.....	219
Iniciando uma Nova Transferência de Arquivos.....	220

Criando uma Transferência de Arquivo Planejada.....	224
Trabalhando com Transferências de Arquivos Pendentes.....	225
Acionando a Transferência de Arquivo.....	226
Monitorando Transferências de Arquivos que Estão em Andamento.....	227
Visualizando o Status de Transferências de Arquivos no Log de Transferência.....	229
Monitorando recursos do MFT.....	232
Trabalhando com modelos de transferência de arquivos.....	264
Transferindo dados de arquivos para mensagens.....	267
Transferindo Dados de Mensagens para Arquivos.....	283
A ponte de protocolo.....	294
A Ponte Connect:Direct.....	317
Trabalhando com o MFT no IBM Integration Bus.....	334
Recuperação e reinicialização do MFT.....	334
Configurando um tempo limite para recuperação de transferências paralisadas.....	335
Administrando MQ Telemetry.....	341
Configurando um gerenciador de filas para telemetria no Linux e AIX.....	341
Configurando um gerenciador de filas para telemetria no Windows.....	343
Configurando o enfileiramento distribuído para enviar mensagens aos clientes do MQTT.....	345
Identificação, autorização e autenticação do cliente do MQTT.....	348
Autenticação de canal de telemetria usando o TLS.....	354
Privacidade de publicação em canais de telemetria.....	358
Configuração do TLS de clientes MQTT Java e canais de telemetria.....	359
Configuração JAAS do Canal de Telemetria.....	364
Administrando um cliente de AMQP.....	366
O serviço AMQP não inicia automaticamente na inicialização do gerenciador de filas.....	366
Visualizando objetos IBM MQ em uso por clientes de AMQP.....	366
Identificação, autorização e autenticação do cliente de AMQP.....	368
privacidade em canais de publicação.....	370
Configurando clientes de AMQP com TLS.....	371
Desconectando clientes de AMQP do gerenciador de filas.....	371
Administrando o multicast.....	372
Introdução ao Multicast.....	372
Topologia do tópico do IBM MQ Multicast.....	373
Controlando o tamanho das mensagens multicast.....	374
Ativando a conversão de dados para mensagens multicast.....	376
Monitoramento de aplicativo multicast.....	377
Confiabilidade de mensagem multicast.....	377
Tarefas avançadas de multicast.....	378
Administrando IBM MQ for IBM i.....	381
Gerenciando IBM MQ for IBM i usando os comandos de CL.....	382
Maneiras alternativas de administrar o IBM MQ for IBM i.....	395
Gerenciamento de trabalho para IBM i.....	401
Disponibilidade, backup, recuperação e reinicialização no IBM i.....	408
Efetuando quiesce do IBM MQ for IBM i.....	453
Administering IBM MQ for z/OS.....	457
Issuing queue manager commands on z/OS.....	457
Using the operations and control panels on z/OS.....	471
Using the IBM MQ for z/OS utilities.....	480
Using the Command Facility on z/OS.....	482
Working with IBM MQ objects on z/OS.....	483
Implementing the system using multiple cluster transmission queues.....	485
Writing programs to administer IBM MQ for z/OS.....	488
Managing IBM MQ resources on z/OS.....	499
Recovery and restart on z/OS.....	537
IBM MQ and IMS.....	558
Operating Advanced Message Security on z/OS.....	570
Administrando IBM MQ Internet Pass-Thru.....	571
Iniciando e Parando o MQIPT.....	571

Administrando MQIPT usando a linha de comandos.....	574
Fazendo backups.....	580
Ajuste de desempenho.....	580
Avisos.....	583
Informações sobre a Interface de Programação.....	584
Marcas comerciais.....	585

Administrando IBM MQ

Para administrar seus gerenciadores de filas e recursos associados do IBM MQ, escolha seu método preferencial dentre um conjunto de tarefas que é possível usar para ativar e gerenciar esses recursos.

Sobre esta tarefa

É possível administrar objetos do IBM MQ localmente ou remotamente:

Administração Local

Administração local significa executar tarefas de administração em quaisquer gerenciadores de filas que você definiu no seu sistema local. É possível acessar outros sistemas, por exemplo, através do programa de emulação de terminal TCP/IP **telnet** e realizar a administração lá. No IBM MQ, é possível considerar isso como administração local porque nenhum canal está envolvido, ou seja, a comunicação é gerenciada pelo sistema operacional.

Para obter informações adicionais, consulte [“Trabalhando com objetos locais do IBM MQ”](#) na página 130.

Administração remota


O IBM MQ suporta a administração de um único ponto de contato por meio de administração remota. A Administração remota permite emitir comandos do sistema local que são processados em outro sistema e também se aplica ao IBM MQ Explorer. Por exemplo, é possível emitir um comando remoto para mudar uma definição de fila em um gerenciador de filas remotas. Você não precisa efetuar logon nesse sistema, embora você precise ter os canais adequados definidos. O gerenciador de filas e o servidor de comandos no sistema de destino devem estar em execução.

Alguns comandos não podem ser emitidos desta forma, em especial, criar ou iniciar gerenciadores de filas e o comando iniciar servidores. Para executar esse tipo de tarefa, deve-se efetuar logon no sistema remoto e emitir os comandos de lá ou criar um processo que possa emitir os comandos para você. Esta restrição também se aplica ao IBM MQ Explorer.




Para obter informações adicionais, consulte [“Trabalhando com objetos remotos do IBM MQ”](#) na página 199.


Existem inúmeros métodos diferentes que é possível usar para criar e administrar os seus gerenciadores de filas e seus recursos relacionados em IBM MQ. Esses métodos incluem interfaces da linha de comandos, interfaces gráficas com o usuário e uma API de administração.

Existem diferentes conjuntos de comandos que podem ser usados para administrar o IBM MQ dependendo da sua plataforma:

- [“Comandos de Controle IBM MQ”](#) na página 8
- [“Comandos do IBM MQ Script \(MQSC\)”](#) na página 8
- [“Formatos de Comandos Programáveis \(PCFs\)”](#) na página 8
- [A administrative REST API](#)
-  [“IBM i Linguagem de Controle \(CL\)”](#) na página 10

Também há outras seguintes opções para criar e gerenciar os objetos do IBM MQ:

-   [“O IBM MQ Explorer”](#) na página 9
- [“O IBM MQ Console”](#) na página 9
-  [“O Microsoft Cluster Service \(MSCS\)”](#) na página 10

 Para obter informações sobre as interfaces de administração e opções sobre o IBM MQ for z/OS, consulte [“Administering IBM MQ for z/OS”](#) na página 457

É possível automatizar algumas tarefas de administração e monitoramento para gerenciadores de filas locais e remotas usando comandos PCF. Estes comandos também podem ser simplificados usando o IBM MQ Administration Interface (MQAI) em algumas plataformas. Para obter informações adicionais sobre a automatização das tarefas de administração, consulte [“Automatizando a administração do IBM MQ usando comandos PCF”](#) na página 25.

Conceitos relacionados

[Visão Geral Técnica do IBM MQ](#)

Tarefas relacionadas

[Planejando](#)

[Configurando](#)

Referências relacionadas

[Comparação de conjuntos de comandos](#)

Maneiras de administrar gerenciadores de filas e recursos associados do IBM MQ

É possível administrar gerenciadores de filas do IBM MQ e recursos associados usando IBM MQ comandos de controle, IBM MQ Comandos de Script (MQSC), Formatos de Comandos Programáveis (PCFs), o administrative REST API, o IBM MQ Console e o IBM MQ Explorer. Para o IBM i, também é possível usar o IBM i Control Language e para o Windows também é possível usar o Microsoft Cluster Service (MSCS).

Comandos de Controle IBM MQ

Multi

Os comandos de controle fornecem uma maneira de executar várias tarefas de administração do IBM MQ. Para AIX, Linux®, and Windows, emita esses comandos na linha de comandos do sistema. Para IBM i, você emite esses comandos em um Qshell. Consulte [“Administrando o IBM MQ for Multiplatforms usando comandos de controle”](#) na página 10.

Comandos do IBM MQ Script (MQSC)

Use os comandos MQSC para gerenciar os objetos do gerenciador de filas, incluindo o próprio gerenciador de filas, filas, definições de processo, listas, canais, canais de conexão do cliente, listeners, serviços e objetos de informações de autenticação.

ALW

No AIX, Linux, and Windows, você abre um prompt de comando do **runmqsc** e, em seguida, emite comandos MQSC para um gerenciador de filas local ou remoto a partir desse prompt. É possível fazer isso interativamente ou executar uma sequência de comandos a partir de um arquivo de texto ASCII. Para obter mais informações, consulte [“Executando comandos MQSC interativamente em runmqsc”](#) na página 18 e [“Executando comandos MQSC a partir de arquivos de texto em runmqsc”](#) na página 23.

IBM i

No IBM i, você cria uma lista de comandos em um arquivo de Script e, em seguida, executa o arquivo usando o comando **STRMQMMQSC**. Para obter informações adicionais, consulte [“Administrando usando comandos MQSC no IBM i”](#) na página 396.

z/OS

No z/OS, os comandos MQSC podem ser emitidos por meio de várias origens, dependendo do comando. Para obter mais informações, consulte [“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS”](#) na página 458.

Formatos de Comandos Programáveis (PCFs)

Os Formatos de Comando Programável (PCFs) definem as mensagens de resposta e comando que podem ser trocadas entre um programa e qualquer gerenciador de filas (que suporta PCFs) em uma

rede. É possível usar os comandos PCF em um programa de aplicativo de gerenciamento de sistemas para administração de objetos IBM MQ: objetos de informações de autenticação, canais, listeners de canais, listas de nomes, definições de processo, gerenciadores de fila, filas, serviços e classes de armazenamento. O aplicativo pode operar a partir de um único ponto na rede para comunicar informações de resposta e comando com qualquer gerenciador de filas, local ou remoto, usando o gerenciador de filas locais.

Para obter informações adicionais sobre PCFs, consulte [“Introdução aos formatos de comando programável do IBM MQ”](#) na página 26.

Para definição de PCFs e estruturas para comandos e respostas, consulte [Referência de Formatos de Comando Programável](#).

O administrative REST API

A administrative REST API fornece uma interface RESTful que você pode usar para administrar o IBM MQ. Quando você usa a administrative REST API, você chama um método de HTTP em uma URL que representa um objeto do IBM MQ. Por exemplo, é possível solicitar informações sobre as instalações do IBM MQ usando o método de HTTP GET na URL a seguir:

```
https://localhost:9443/ibmmq/rest/v1/admin/installation
```

É possível usar a administrative REST API com a implementação HTTP/REST de uma linguagem de programação ou usando ferramentas como cURL, ou um complemento de navegador do cliente REST.

Para obter mais informações, consulte [A administrative REST API](#)

O IBM MQ Console

É possível usar o IBM MQ Console para administrar o IBM MQ por meio de um navegador da web.

Para obter informações adicionais, consulte [“Administração usando o IBM MQ Console”](#) na página 92.


O IBM MQ Explorer



Usando o IBM MQ Explorer, é possível executar as seguintes ações:

- Definir e controlar vários recursos, como gerenciadores de filas, filas, definições de processo, listas de nomes, canais, canais de conexão do cliente, listeners, serviços e clusters.
- Iniciar ou parar um gerenciador de filas locais e seus processos associados.
- Visualizar os gerenciadores de filas e seus objetos associados em sua estação de trabalho ou de outras estações de trabalho.
- Verificar o status de gerenciadores de filas, clusters e canais.
- Verifique se os aplicativos, usuários ou canais possuem uma fila específica aberta, a partir do status de fila.

Nos sistemas Windows e Linux for x86-64 é possível iniciar IBM MQ Explorer usando o menu do sistema ou o arquivo executável MQExplorer .

 No Linux, para iniciar o IBM MQ Explorer com êxito, deve-se conseguir gravar um arquivo para o seu diretório inicial e o diretório inicial deve existir.

Para obter informações adicionais, consulte [“Administração usando o IBM MQ Explorer”](#) na página 121.

É possível usar IBM MQ Explorer para administrar gerenciadores de filas remotas em outras plataformas incluindo z/OS.

De IBM MQ 9.3.0, IBM MQ Explorer foi removido do pacote de instalação do IBM MQ . Ele permanece disponível como um download separado e pode ser instalado a partir de download independente

do IBM MQ Explorer, disponível em Fix Central. Para obter mais informações, consulte [Instalando e desinstalando IBM MQ Explorer como um aplicativo independente no Linux e no Windows](#)

IBM i Linguagem de Controle (CL)

IBM i

Essa é a maneira preferencial de emitir comandos de administração para IBM MQ for IBM i. Os comandos podem ser emitidos na linha comando ou gravando um programa CL. Esses comandos executam funções semelhantes aos comandos PCF, mas o formato é diferente. Os comandos CL são projetados exclusivamente para servidores e as respostas CL são legíveis, enquanto os comandos PCF são independentes de plataforma e os formatos de resposta e de comando são destinados para uso do programa.

Para obter detalhes completos do IBM i Control Language (CL), consulte [“Gerenciando IBM MQ for IBM i usando os comandos de CL”](#) na página 382 e [IBM MQ for IBM i Comandos CL](#).

O Microsoft Cluster Service (MSCS)

Windows

Microsoft Cluster Service (MSCS) permite que você conecte os servidores a um *cluster*, oferecendo maior disponibilidade de dados e aplicativos e tornando-o mais fácil de gerenciar o sistema. O MSCS pode automaticamente detectar e recuperar-se das falhas no servidor ou aplicativo.

É importante não confundir os clusters na detecção MSCS com clusters IBM MQ. A distinção é a seguinte:

Clusters do IBM MQ

Estes são grupos de dois ou mais gerenciadores de filas em um ou mais computadores, fornecendo interconexão automática e permitindo que as filas sejam compartilhadas entre eles para balanceamento de carga e redundância.

Clusters do MSCS

Estes são grupos de computadores conectados e configurados de maneira que, se um falhar, o MSCS executará um *failover*, transferindo os dados de estado dos aplicativos do computador com falha para outro computador no cluster e reiniciando sua operação nele.

[Suportando o Microsoft Cluster Service \(MSCS\)](#) fornece informações detalhadas sobre como configurar seu sistema para utilizar o MSCS do IBM MQ for Windows.

Tarefas relacionadas

[“Administrando o IBM MQ usando comandos MQSC”](#) na página 12

É possível usar comandos MQSC para gerenciar objetos do gerenciador de filas, incluindo o próprio gerenciador de filas, filas, definições de processo, canais, canais de conexão do cliente, listeners, serviços, listas de nomes, clusters e objetos de informações de autenticação. Os comandos MQSC estão disponíveis em todas as plataformas.

Referências relacionadas

[Referência de administração](#)

Multi

Administrando o IBM MQ for Multiplatforms usando comandos de controle

Os comandos de controle fornecem uma maneira de executar várias tarefas de administração do IBM MQ. Para AIX, Linux e Windows, você emite esses comandos na linha de comandos do sistema. Para IBM i, você emite esses comandos em um Qshell.

Antes de começar

Ao usar comandos de controle que operam em um gerenciador de filas, você deve usar o comando da instalação associada ao gerenciador de filas com o qual está trabalhando.

Ao usar comandos de controle que operam em um gerenciador de filas configurado para usar a autenticação de conexão com CHCKLOCL (REQUIRED) e uma falha ao se conectar for observada, escolha um ou outro:

- Forneça um ID do usuário e senha se o comando de controle permitir isso.
- Use equivalentes do MQSC dos comandos de controle no qual eles existem.
- Inicie o gerenciador de filas usando a opção -ns, enquanto comandos de controle que não podem se conectar precisam ser executados.

Nota: Diferentes plataformas podem aceitar argumentos de comando inseridos em uma ordem diferente.. Especificamente, isso significa que os comandos que funcionam no Linux podem não funcionar em outras plataformas. Por esse motivo, você deve sempre inserir argumentos conforme especificado nos diagramas de sintaxe.

Para obter uma lista completa dos comandos de controle, consulte a [IBM MQ referência de comandos de controle](#)

Procedimento

-  Linux AIX

Use comandos de controle em sistemas AIX and Linux ..

Em sistemas IBM MQ for AIX or Linux, você insere comandos de controle em uma janela shell.

Para emitir comandos de controle, seu ID de usuário deve ser um membro do grupo mqm para a maioria dos comandos de controle. Para obter mais informações sobre isso, consulte [Autoridade para administrar o IBM MQ no AIX, Linux, and Windows](#). Além disso, observe as informações específicas do ambiente. para a plataforma, ou plataformas, que a sua empresa usa.

Em ambientes UNIX and Linux, os controles de comando, incluindo o próprio nome do comando, as sinalizações e quaisquer argumentos, fazem distinção entre maiúsculas e minúsculas. Por exemplo, no comando:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- O nome do comando deve ser `crtmqm`, não `CRTMQM`.
- A sinalização deve ser `-u`, não `-U`.
- A fila de mensagens não entregues é chamada `SYSTEM.DEAD.LETTER.QUEUE`.
- O argumento é especificado como `jupiter.queue.manager`, que é diferente de `JUPITER.queue.manager`.

Tome cuidado para digitar os comandos exatamente como você os vê nos exemplos.

-  Windows

Use comandos de controle em sistemas Windows ..

No IBM MQ for Windows, você insere comandos de controle em um prompt de comandos.

Para emitir comandos de controle, seu ID de usuário deve ser um membro do grupo mqm para a maioria dos comandos de controle. Para obter mais informações sobre isso, consulte [Autoridade para administrar o IBM MQ no AIX, Linux, and Windows](#). Além disso, observe as informações específicas do ambiente. para a plataforma, ou plataformas, que a sua empresa usa.

Os comandos de controle e suas sinalizações não fazem distinção entre maiúsculas e minúsculas, mas os argumentos para esses comandos, como nomes de filas e nomes de gerenciadores de filas, fazem distinção entre maiúsculas e minúsculas.

Por exemplo, no comando:

```
crtmqm /u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- O nome do comando pode ser inserido em maiúsculas ou minúsculas ou uma combinação dos dois. Todos estes são válidos: `crtmqm`, `CRTMQM` e `CRTmqm`.
- A sinalização pode ser inserida como `-u`, `-U`, `/u` ou `/U`.
- `SYSTEM.DEAD.LETTER.QUEUE` e `jupiter.queue.manager` devem ser inseridos exatamente conforme mostrado.

IBM i

Use comandos de controle em sistemas IBM i ..

No IBM MQ for IBM i, você executa os comandos de controle de um ambiente Qshell. Para usar o Qshell, digite `STRQSH` na linha de comandos IBM i . É possível sair e retornar à linha de comandos a qualquer momento pressionando F3.

Um pequeno número de comandos de controle não é suportado no IBM i Por exemplo, comandos de múltiplas instalações não são suportados porque não é possível ter mais de uma cópia de IBM MQ em um sistema IBM i . Comandos que não são suportados no IBM i são sinalizados **ALW** na [IBM MQ referência de comandos de controle](#).

Referências relacionadas

[Referência de comandos de controle do IBM MQ](#)

Administrando o IBM MQ usando comandos MQSC

É possível usar comandos MQSC para gerenciar objetos do gerenciador de filas, incluindo o próprio gerenciador de filas, filas, definições de processo, canais, canais de conexão do cliente, listeners, serviços, listas de nomes, clusters e objetos de informações de autenticação. Os comandos MQSC estão disponíveis em todas as plataformas.

Sobre esta tarefa

Comandos MQSC disponíveis são detalhados na [referência de comandos MQSC](#).

A maneira como você emite comandos MQSC depende de sua plataforma:

- ALW** No AIX, Linux, and Windows, você emite comandos MQSC para um gerenciador de fila a partir do prompt de comandos `runmqsc` . É possível usar esse prompt de comandos de diversas maneiras:
 - Interativamente, emitindo comandos MQSC a partir de um teclado. Consulte [“Executando comandos MQSC interativamente em runmqsc”](#) na página 18.
 - Emitir comandos MQSC a partir de um arquivo de texto ASCII. Consulte [“Executando comandos MQSC a partir de arquivos de texto em runmqsc”](#) na página 23.
 - Emitindo comandos do MQSC em um gerenciador de filas remotas Consulte [“Emitindo comandos MQSC em um gerenciador de filas remotas”](#) na página 205.
- IBM i** No IBM i, você cria uma lista de comandos em um arquivo de Script e, em seguida, executa o arquivo usando o comando **STRMQMQSC** Para obter informações adicionais, consulte [“Administrando usando comandos MQSC no IBM i”](#) na página 396.
- z/OS** No z/OS, os comandos MQSC podem ser emitidos por meio de várias origens, dependendo do comando. Para obter mais informações, consulte [“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS”](#) na página 458.

Procedimento

- [“Sintaxe do comando MQSC”](#) na página 13
- [“MQSC: caracteres especiais e valores genéricos”](#) na página 15
- [“Executando comandos MQSC interativamente em runmqsc”](#) na página 18

- [“Executando comandos MQSC a partir de arquivos de texto em runmqsc” na página 23](#)
- [“Configuração automática por meio de um script MQSC na inicialização” na página 24](#)

Tarefas relacionadas

[Resolvendo problemas com os comandos do MQSC](#)

Referências relacionadas

[runmqsc \(executar comandos MQSC\)](#)

Sintaxe do comando MQSC

É possível usar comandos MQSC para gerenciar objetos do gerenciador de filas. Os comandos MQSC estão disponíveis em todas as plataformas. Alguns elementos da sintaxe de comando são específicos da plataforma.

Sequência de parâmetros

Cada comando inicia com um parâmetro primário (um verbo) e isso é seguido por um parâmetro secundário (um nome). Em seguida, isso é seguido pelo nome ou nome genérico do objeto (entre parênteses), se houver um, que existe na maioria dos comandos. Seguindo isso, os parâmetros geralmente podem ocorrer em qualquer ordem; se um parâmetro tiver um valor correspondente, o valor deve ocorrer diretamente após o parâmetro ao qual se relaciona.

Nota:  No z/OS, o parâmetro secundário não precisa ser o segundo.

Espaços em branco e vírgulas.

As palavras-chave, os parênteses e os valores podem ser separados por qualquer número de espaços em branco e vírgulas. Uma vírgula mostrada nos diagramas de sintaxe sempre pode ser substituída por um ou mais espaços em branco. Deve haver pelo menos um espaço em branco imediatamente antes de cada parâmetro (após o parâmetro primário) exceto no z/OS.

Pode ocorrer qualquer número de espaços em branco no início ou término do comando e entre parâmetros, pontuação e valores. Por exemplo, o seguinte comando é válido:

```
ALTER QLOCAL ('Account' ) TRIGDPTH ( 1)
```

Os espaços em branco em um par de marcas de aspas são significativos.

As vírgulas adicionais podem aparecer em qualquer lugar em que os espaços em branco forem permitidos e tratados como se fossem espaços em branco (a menos que estejam, obviamente, dentro de sequências entre aspas).

Parâmetros repetidos

Não são permitidos parâmetros repetidos. Repetir um parâmetro com sua versão "NO", como em REPLACE NOREPLACE, também não é permitido.

Sequências e aspas simples

As sequências que contêm espaços em branco, caracteres minúsculos ou caracteres especiais devem ser colocadas entre aspas simples, a menos que um dos seguintes seja verdadeiro:

- Os caracteres especiais são um ou mais dos caracteres a seguir:
 - Ponto (.)
 - Barra (/)
 - Sublinhado (_)
 - Sinal de percentual (%)

- **z/OS** O comando é emitido por meio de operações e painéis de controle do IBM MQ for z/OS.
- A sequência é um valor genérico que termina com um asterisco. (No IBM i, isso deve ser colocado entre aspas simples)
- A sequência é um asterisco único, por exemplo, TRACE(*) (no IBM i, isso deve ser colocado entre aspas simples)
- A sequência é uma especificação de intervalo contendo dois-pontos, por exemplo, CLASS(01:03)

Se a própria sequência contiver aspas simples, ela será representada por duas aspas simples.

Multi No Multiplataformas, uma sequência que não contém caracteres (ou seja, duas aspas simples sem espaço entre elas) é interpretada como um espaço em branco entre aspas simples, ou seja, interpretado da mesma forma que (' '). A exceção a isso é se o atributo usado for um dos seguintes atributos, quando duas aspas simples sem espaço são interpretadas como uma sequência de comprimento zero:

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

z/OS No z/OS, se desejar um espaço em branco entre aspas simples, deve-se inseri-lo como (' '). Uma sequência sem caracteres (") é o mesmo que inserir ().

Quaisquer espaços em branco à direita nos atributos de sequência que são baseados em tipos MQCHARV, como SELECTOR, dados do subusuário, são tratados como significativos, o que significa que 'abc ' não é igual a 'abc '.

Parênteses vazios

Um parêntese de abertura seguido por um parêntese de fechamento, sem informações significativas no meio, não é válido, exceto onde indicado especificamente. Por exemplo, a sequência a seguir não é válida:

```
NAME ( )
```

Minúsculas e maiúsculas

As palavras-chave não fazem distinção entre maiúsculas e minúsculas: AltERAR, alterar e ALTERAR são todas aceitáveis.

Tudo o que não estiver contido nas aspas é convertido em letras maiúsculas.

Sinônimos

Os sinônimos são definidos para alguns parâmetros. Por exemplo, o DEF é sempre um sinônimo para DEFINE; portanto, DEF QLOCAL é válido. No entanto, os sinônimos não são apenas sequências mínimas; DEFI não é um sinônimo válido para DEFINE.

Nota: Não há sinônimo para o parâmetro DELETE. Isso evita a exclusão acidental de objetos ao usar DEF, o sinônimo para DEFINE.

Caracteres Especiais

Os comandos MQSC usam certos caracteres especiais para ter certos significados. Para obter informações adicionais sobre estes caracteres especiais e como usá-los, consulte [“MQSC: caracteres especiais e valores genéricos” na página 15](#).

Tarefas relacionadas

Resolvendo problemas com os comandos do MQSC

Referências relacionadas

[runmqsc \(executar comandos MQSC\)](#)

MQSC: caracteres especiais e valores genéricos


Alguns caracteres, por exemplo, barra invertida (\) e aspas duplas (") caracteres têm significados especiais quando usados com comandos MQSC. Alguns caracteres especiais que podem ser utilizados com parâmetros podem ter valores genéricos, mas devem ser especificados corretamente.

Preceder barra invertida (\) e aspas duplas (") com um \, ou seja, insira \\ ou \" se desejar \ ou " em seu texto.

Sempre que um parâmetro puder ter um valor genérico, ele é inserido encerrando com um asterisco (*), por exemplo, ABC*. Um valor genérico significa todos os valores que começam com; portanto, ABC* significa todos os valores que começam com ABC. Se caracteres que requerem aspas são usados no valor, o asterisco deve ser colocado dentro das aspas, portanto, 'abc*'. O asterisco deve ser o último ou o único caractere no valor.

O ponto de interrogação (?) e o colon (:) não são permitidos em valores genéricos.

Quando você precisar usar algum desses caracteres especiais em um campo (por exemplo, como parte de uma descrição), deve colocar a sequência inteira entre aspas simples.

Caractere	Descrição
	Os espaços em branco são usados como separadores. Vários espaços em branco são equivalentes a um único espaço em branco, exceto em sequências que estão entre apóstrofos ('). Quaisquer espaços em branco à direita nesses atributos de sequência que são baseados em tipos MQCHARV são tratados como significativos.
,	As vírgulas são usadas como separadoras. Múltiplas vírgulas são equivalentes a um único vírgula, exceto em cadeias que são delimitadas entre apóstrofos (').
'	Um apóstrofo indica o início ou fim de uma sequência. O IBM MQ deixa todos os caracteres que são colocados entre aspas exatamente como eles são inseridos. Os apóstrofos pertencentes não são incluídos ao calcular o comprimento da sequência.
"	Aspas únicas dentro de uma sequência são tratadas pelo IBM MQ como um caractere ao calcular o comprimento da sequência e a sequência não é finalizada.
=	 No z/OS, um sinal de igual indica o início de um valor de parâmetro que é finalizado por uma vírgula ou em branco.
(Um parêntese aberto indica o início de um valor de parâmetro ou lista de valores.
)	Um parêntese de fechamento indica o fim de um valor de parâmetro ou lista de valores.
:	Um dois pontos indica um intervalo inclusivo. Por exemplo, (1:5) significa (1,2,3,4,5). Essa notação pode ser usada apenas nos comandos TRACE .
*	Um asterisco significa tudo. Por exemplo, DISPLAY TRACE (*) significa exibir todos os rastreios, e DISPLAY QUEUE (PAY*) significa exibir todas as filas com nomes que começam com PAY.

Sintaxe do arquivo de entrada MQSC

Se você tiver comandos longos ou estiver usando uma sequência específica de comandos repetidamente, poderá usar um arquivo de entrada para emitir comandos MQSC. O conteúdo do arquivo de entrada deve seguir a sintaxe descrita neste tópico

Visão Geral

Os comandos MQSC são inseridos por meio do *dispositivo de entrada padrão*, também referido como `stdin`. Geralmente, esse é o teclado, mas é possível especificar que a entrada deve vir de um arquivo de entrada

É possível usar esse arquivo de entrada com qualquer uma das seguintes ferramentas específicas da plataforma:

- ▶ **ALW** O comando **runmqsc** no AIX, Linux, and Windows. Consulte o [“Executando comandos MQSC a partir de arquivos de texto em runmqsc”](#) na página 23
- ▶ **IBM i** O comando **STRMQM** no IBM i. Consulte o [“Administrando usando comandos MQSC no IBM i”](#) na página 396
- ▶ **z/OS** Os conjuntos de dados de inicialização CSQINP1, CSQINP2 e CSQINPX ou o utilitário em lote CSQUTIL no z/OS. Consulte o [“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS”](#) na página 458

Sintaxe

Sintaxe do arquivo de entrada MQSC:

- Para portabilidade entre ambientes IBM MQ, limite o comprimento de linha em arquivos de comando MQSC a 72 caracteres.
- Cada comando deve iniciar em uma nova linha.
- Uma linha iniciando com um asterisco (*) na primeira posição é ignorada. Isso pode ser usado para inserir os comentários no arquivo.
- Linhas em branco são ignoradas.
- Um sinal de mais (+) indica que o comando continua a partir do primeiro caractere não em branco na próxima linha. Se você usar + para continuar um comando, lembre-se de deixar pelo menos um espaço em branco antes do próximo parâmetro (exceto no z/OS em que isso não é necessário). Quaisquer comentários ou linhas em branco são descartados quando o comando é remontado em uma única sequência.
- Um sinal de menos (-), isso indica que o comando deve ser continuado do início da próxima linha. Quaisquer comentários ou linhas em branco são descartados quando o comando é remontado em uma única sequência.
- Os comandos MQSC que estão contidos dentro de um comando Escape PCF (Formato de comando programável) não podem ser continuados com o sinal de mais ou o sinal de menos. O comando inteiro deve estar contido dentro de um único comando de Escape. Para obter informações sobre os comandos PCF, consulte [“Introdução aos formatos de comando programável do IBM MQ”](#) na página 26.
- No Multiplataformas e no z/OS, para comandos emitidos por meio do programa utilitário em lote CSQUTIL, é possível usar um caractere de ponto e vírgula (;) para finalizar um comando, mesmo que você tenha inserido um sinal de mais (+) no final da linha anterior.
- Uma linha não deve encerrar em um caractere de controle de teclado (por exemplo, uma guia).
- Se você executar o comando **runmqsc** no modo cliente redirecionando `stdin` de um arquivo de texto e fornecer a sinalização **-u** para fornecer credenciais, o comando **runmqsc** não solicitará uma senha e, em vez disso, a senha será lida de `stdin`. Deve-se assegurar que a primeira linha de dados fornecida por meio de `stdin` seja a senha. Isso pode ser feito usando ferramentas de linha de comandos como "echo" ou "cat" e passando a senha seguida pelo script MQSC para o comando **runmqsc** `stdin`.

- Windows No Windows, se alguns caracteres especiais, como o sinal de libra (£) e o NOT lógico (¬), são usados em um script de comando (por exemplo, como parte de uma descrição de objeto), eles são exibidos de forma diferente na saída de um comando como **DISPLAY QLOCAL**.

Consulte também [“Sintaxe do comando MQSC”](#) na página 13.

Exemplos

O exemplo a seguir é uma extração de um arquivo de comando MQSC que mostra o comando **DEFINE QLOCAL**.

```
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +
DESCR(' ') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(1024) +
DEFSOPT(SHARED) +
NOHARDENBO +
USAGE(NORMAL) +
NOTRIGGER;
```

Figura 1. Extraída a partir de um arquivo de comandos MQSC

Quando o comando **runmqsc** é concluído, um relatório é retornado. O exemplo a seguir é uma extração de um relatório:

```
Starting MQSC for queue manager jupiter.queue.manager.
.
.
12:  DEFINE QLOCAL('ORANGE.LOCAL.QUEUE') REPLACE +
:    DESCR(' ') +
:    PUT(ENABLED) +
:    DEFPRTY(0) +
:    DEFPSIST(NO) +
:    GET(ENABLED) +
:    MAXDEPTH(5000) +
:    MAXMSGL(1024) +
:    DEFSOPT(SHARED) +
:    NOHARDENBO +
:    USAGE(NORMAL) +
:    NOTRIGGER;
AMQ8006: IBM MQ queue created.
.
.
```

Figura 2. Extração de um arquivo de relatório de comandos MQSC

Também é possível usar os arquivos de comandos MQSC de exemplo para ajudar a criar seu arquivo de texto:

amqscos0.tst

As definições de objetos usados por programas de amostra.

amqscic0.tst

Definições de filas para transações do CICS.

Linux
AIX
 No AIX and Linux, esses arquivos estão localizados no diretório `MQ_INSTALLATION_PATH/samp`. O `MQ_INSTALLATION_PATH` representa o diretório de alto nível no qual o IBM MQ está instalado.

Windows

No Windows, esses arquivos estão localizados no diretório `MQ_INSTALLATION_PATH\tools\mqsc\samples`. O `MQ_INSTALLATION_PATH` representa o diretório de alto nível no qual o IBM MQ está instalado.

ALW

Executando comandos MQSC interativamente em `runmqsc`

No AIX, Linux, and Windows, é possível usar o prompt de comandos `runmqsc` para emitir comandos MQSC para um gerenciador de filas interativamente. A execução interativa é particularmente adequada para testes rápidos

Antes de começar

Você deve usar o comando `runmqsc` a partir da instalação associada ao gerenciador de filas com o qual você está trabalhando. É possível descobrir a qual instalação um gerenciador de filas está associado usando o comando `dspmqr -o installation`.

É possível facilitar para ver que você está em um ambiente MQSC e ver alguns detalhes do ambiente atual configurando um prompt de sua escolha usando a variável de ambiente `MQPROMPT`. Para obter mais informações, consulte [“Configurando o prompt de comandos MQSC”](#) na página 20.

Linux

AIX

Quando você executa comandos MQSC interativamente em plataformas AIX and Linux, o prompt de comando `runmqsc` também suporta funções adicionais do editor de linhas de comando. Consulte [“Ativando a chamada e a conclusão do comando e as chaves de comando Emacs, para `runmqsc`”](#) na página 22.

Sobre esta tarefa

O comando `runmqsc` é usado para abrir um prompt de comandos a partir do qual é possível emitir comandos MQSC.. Esses comandos e sua sintaxe são descritos na [Referência de comandos do MQSC](#)

Ao iniciar o prompt de comandos `runmqsc` conforme descrito nesta tarefa, você configura o prompt para executar em um dos três modos, dependendo dos sinalizadores configurados no comando:

- *Modo de verificação*, em que os comandos MQSC são verificados em um gerenciador de filas local, mas não são executados.
- *Modo direto*, em que os comandos MQSC são executados em um gerenciador de filas local.
- *Modo indireto*, em que os comandos MQSC são executados em um gerenciador de filas remotas.

O procedimento abaixo configura o prompt para ser executado no modo direto. Outras opções são ilustradas nos exemplos que seguem as etapas principais

Procedimento

1. Abra uma janela de comando ou shell e insira o seguinte comando:

```
runmqsc QMgrName
```

Em que `QMgrName` especifica o nome do gerenciador de filas no qual você deseja processar os comandos MQSC. É possível deixar `QMgrName` em branco para processar comandos MQSC no gerenciador de filas padrão.

2. Digite quaisquer comandos MQSC, conforme necessário. Por exemplo, para criar uma fila local chamada `ORANGE.LOCAL.QUEUE`, insira o comando a seguir:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE)
```

Para comandos que possuem parâmetros em excesso para caber em uma única linha, utilize caracteres de continuação para indicar que um comando é continuado na linha a seguir:

- Um sinal de menos (-) indica que o comando deve ser continuado a partir do início da linha seguinte.
- Um sinal de mais (+) indica que o comando deve ser continuado a partir do primeiro caractere não em branco na linha a seguir.

A entrada de comando termina com o caractere final de uma linha não em branco que não é um caractere de continuação. É possível também finalizar de entrada de comando explicitamente digitando um ponto e vírgula (;).

3. Pare de trabalhar com comandos MQSC inserindo o comando a seguir:

```
end
```

Como alternativa, é possível usar o comando **exit**, o comando **quit** ou o caractere EOF para seu sistema operacional.

Resultados

Ao emitir os comandos do MQSC, o gerenciador de filas retorna mensagens do operador que confirmam suas ações ou informam sobre os erros que você cometeu. Por exemplo, a mensagem a seguir confirma que uma fila foi criada:

```
AMQ8006: IBM MQ queue created.
```

A mensagem a seguir indica que você fez um erro de sintaxe:

```
AMQ8405: Syntax error detected at or near end of command segment below:-
AMQ8426: Valid MQSC commands are:
```

```
ALTER
CLEAR
DEFINE
DELETE
DISPLAY
END
PING
REFRESH
RESET
RESOLVE
RESUME
START
STOP
SUSPEND
4 : end
```

Essas mensagens são enviadas ao dispositivo de saída padrão, que por padrão é a exibição. Se você não tiver inserido o comando corretamente, consulte as informações de referência para o comando para localizar a sintaxe correta. Consulte [Referência de comandos do MQSC](#)

Exemplo

A seguir estão as variantes do comando `runmqsc QMgrName` que é usado nas etapas anteriores: Essas variantes criam diferentes configurações do prompt de comandos do **runmqsc**

- O comando a seguir usa a filtragem de comando para transmitir um único comando MQSC para o interpretador do MQSC

No Windows:

```
echo display chstatus(*) | runmqsc QMname
```

No Linux:

```
echo "display chstatus(*)" | runmqsc QMname
```

- O comando a seguir não especifica um nome do gerenciador de filas, portanto, os comandos MQSC são processados no gerenciador de filas padrão

```
runmqsc
```

- Esse comando envia comandos para o gerenciador de filas QMREMOTE, usando QMLOCAL para enviar os comandos.

```
runmqsc -w 30 -m QMLOCAL QMREMOTE
```

- Este comando verifica se a sintaxe de comandos está correta em um gerenciador de fila local sem executar os comandos. Observe que os comandos a serem verificados são lidos de um arquivo de entrada `myprog.in`.

```
runmqsc -f myprog.in -v QmgrName
```

Para obter mais informações sobre como trabalhar com arquivos de entrada e de saída, consulte [“Executando comandos MQSC a partir de arquivos de texto em runmqsc” na página 23](#)

Como proceder a seguir

Para obter detalhes completos da sintaxe de comando `runmqsc`, parâmetros opcionais e códigos de retorno, consulte [runmqsc \(executar comandos MQSC\)](#).

Tarefas relacionadas

[“Executando comandos MQSC a partir de arquivos de texto em runmqsc” na página 23](#)

Se você tiver comandos longos ou estiver usando uma sequência específica de comandos repetidamente, poderá usar um arquivo de texto para emitir comandos MQSC. É possível redirecionar `stdin` de um arquivo de texto. Também é possível redirecionar a saída para um arquivo.

Referências relacionadas

[Referência de comandos MQSC](#)

ALW Configurando o prompt de comandos MQSC




No AIX, Linux, and Windows, use a variável de ambiente `MQPROMPT` para configurar o prompt que é exibido ao executar o comando `runmqsc`. Isso facilita ver que você está em um ambiente MQSC e ver alguns detalhes do ambiente atual.

Sobre esta tarefa

É possível configurar o prompt que é exibido quando você executa o comando `runmqsc`. O prompt é inserido quando o comando `runmqsc` é executado interativamente e quando a entrada é redirecionada para o `runmqsc` a partir de um arquivo ou do dispositivo de entrada padrão (`stdin`).

É possível incluir texto simples no prompt de comando e também inserir variáveis de ambiente usando a notação `+VARNAME+` da mesma maneira que as definições de objeto de serviço IBM MQ. Para obter informações adicionais, consulte [“Usando inserções substituíveis em definições de serviço” na página 190](#).

Há várias outras inserções substituíveis adicionais fornecidas pelo IBM MQ, descritas na tabela a seguir.

Inserção substituível	Descrição
<code>MQ_HOST_NAME</code>	Nome do host do sistema
<code>MQ_FILE_SEP</code>	Separador de arquivos específico da plataforma: <ul style="list-style-type: none">•   Em sistemas AIX and Linux, o <code>MQ_FILE_SEP</code> é /•  Em sistemas Windows, o local do <code>MQ_FILE_SEP</code> é \
<code>MQ_PATH_SEP</code>	Separador de caminho específico da plataforma:

Inserção substituível	Descrição
	<ul style="list-style-type: none"> Linux AIX Em sistemas AIX and Linux , o MQ_PATH_SEP é : Windows Em sistemas Windows , o local do MQ_PATH_SEP é ;
MQ_DATE_TIME	Data e hora do sistema local em formato fixo YYYY-MM-DD hh:mm:ss.SSS, por exemplo: 2020-12-25 17:41:37.408

Notas:

- Os valores de inserções substituíveis do MQ se relacionam à instalação e ao sistema host do IBM MQ aos quais o comando **runmqsc** está associado.
- MQPROMPT** é limitado a um máximo de 256 caracteres quando as inserções são expandidas. As expansões **MQPROMPT** sobre esse valor resultam na sequência **MQPROMPT** inteira sendo truncada sem as expansões.

Exemplo

O exemplo a seguir configura o aviso para MQSC::

- Linux AIX

```
export MQPROMPT="MQSC"
```

- Windows

```
set "MQPROMPT=MQSC"
```

AIX O exemplo a seguir configura a variável **MQPROMPT** em um sistema AIX . O prompt é configurado para exibir um nome do usuário (obtido a partir da variável de ambiente do sistema associada), o nome do gerenciador de filas e o nome do host IBM MQ (obtido a partir das inserções substituíveis IBM MQ):

```
sh> export MQPROMPT="+USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
sh> runmqsc MY.QMGR
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager MY.QMGR.

myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT="+USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
C:\ > runmqsc MY.QMGR
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager MY.QMGR.

myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

O exemplo a seguir inclui um registro de data e hora nos exemplos **MQPROMPT** acima, obtidos das inserções substituíveis do MQ :

```
sh> export MQPROMPT="+MQ_DATE_TIME+ +USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
sh> runmqsc MY.QMGR
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager MY.QMGR.

2020-11-24 18:10:00.404 myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT="+MQ_DATE_TIME+ +USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
C:\ > runmqsc MY.QMGR
```

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.
```

```
2020-11-24 18:10:01.007 myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

Linux

AIX

Ativando a chamada e a conclusão do comando e as chaves de comando Emacs, para runmqsc

Use o prompt de comandos do **runmqsc** no AIX and Linux para ativar a chamada de comando, a conclusão de comando e as teclas de comandos do Emacs

Sobre esta tarefa

Em sistemas AIX and Linux , é possível disponibilizar as seguintes funções adicionais do editor de linha de comandos a partir do prompt de comando **runmqsc** :

- Lembre-se dos comandos inseridos anteriormente usando a tecla de seta para cima e a tecla de seta para baixo
- A conclusão automática para a próxima palavra-chave de um comando usando a tecla tab e a barra de espaço
- Chaves de comando [Emacs](#) ou funções de tecla de comando semelhantes

Para usar essas funções, a biblioteca curses deve estar instalada. Se a biblioteca curses não estiver instalada em seu sistema, o prompt **runmqsc** não terá as funções do editor de linha de comandos e uma mensagem será mostrada quando o prompt de comandos do **runmqsc** for iniciado O nome da biblioteca curses a ser instalada dependerá da plataforma UNIX:

- **AIX** No AIX, instale curses.
- **Linux** No Linux, instale ncurses.

Procedimento

- Instale ncurses ou curses.

Nota: O exemplo a seguir usa instruções para Linux

Execute o comando a seguir para localizar os pacotes ncurses existentes:

```
rpm -qa | grep -i ncurses
```

Os pacotes ncurses necessários são os seguintes:

```
ncurses-term-6.1-7.20180224.el8.noarch  
ncurses-6.1-7.20180224.el8.x86_64  
ncurses-base-6.1-7.20180224.el8.noarch  
ncurses-c++-libs-6.1-7.20180224.el8.x86_64  
ncurses-libs-6.1-7.20180224.el8.x86_64  
ncurses-compat-libs-6.1-7.20180224.el8.x86_64  
ncurses-devel-6.1-7.20180224.el8.x86_64
```

É possível instalar todos os pacotes ncurses necessários listados no texto anterior executando o comando a seguir:

```
yum install ncurses*
```

- Customize as ligações de teclas Emacs

É possível customizar as teclas ligadas aos comandos. Por exemplo, é possível ligar as teclas a ligações vi, em vez das ligações de teclas Emacs padrão.

As chaves são customizadas editando o arquivo `.editrc` que é armazenado no diretório inicial. Para obter mais informações, consulte [editrc](#) nas páginas principais do FreeBSD.

- Desative a chamada do comando, a conclusão do comando e as chaves de comando do Emacs

Para isso, configure a variável de ambiente **MQ_OVERRIDE_LIBEDIT_LOAD** como TRUE.

Essa variável de ambiente pode ser usada como uma solução alternativa quando o prompt de comando **runmqsc** exibe a seguinte mensagem informativa:

```
AMQ8521I: Command completion and history unavailable
```

ALW Executando comandos MQSC a partir de arquivos de texto em **runmqsc**

Se você tiver comandos longos ou estiver usando uma sequência específica de comandos repetidamente, poderá usar um arquivo de texto para emitir comandos MQSC. É possível redirecionar `stdin` de um arquivo de texto. Também é possível redirecionar a saída para um arquivo.

Antes de começar

Esta tarefa supõe que você criou um arquivo de texto que contém os comandos MQSC que deseja executar. Para obter sintaxe detalhada e exemplos desses arquivos, consulte [“Sintaxe do arquivo de entrada MQSC”](#) na página 16.

É possível configurar o prompt de comando MQSC para um prompt de sua escolha usando a variável de ambiente **MQPROMPT**. Para obter mais informações, consulte [“Configurando o prompt de comandos MQSC”](#) na página 20.

Sobre esta tarefa

A entrada para o comando **runmqsc** é tomada do *dispositivo de entrada padrão*, também referido como `stdin`. Geralmente este é o teclado, mas você pode especificar que a entrada deve vir de uma porta serial ou um arquivo.

A saída para o comando **runmqsc** é enviada para o *dispositivo de saída padrão*, também referido como `stdout`. Geralmente esse é um vídeo, mas é possível redirecionar a saída para uma porta serial ou um arquivo.

Procedimento

1. Em um gerenciador de fila local, verifique se a sintaxe de comando no arquivo está correta sem executar os comandos

Use a sinalização **-v** no comando **runmqsc**, juntamente com uma das opções a seguir:

- Use a opção **-f** para identificar o nome do arquivo de texto de saída. Por exemplo:

```
runmqsc -f myprog.in -v localQmgrName
```

Não é possível especificar um gerenciador de filas remotas ao verificar comandos. Ou seja, não é possível especificar a sinalização **-w**.

O relatório retornado é semelhante àquele mostrado em [Figura 2 na página 17](#).

2. Quando a sintaxe de comando estiver correta, remova a sinalização **-v** e execute novamente o comando **runmqsc**.

Observe que agora é possível especificar um gerenciador de filas remotas

- Execute (por exemplo) o comando a seguir:

```
runmqsc -f myprog.in QmgrName
```

[Figura 1 na página 17](#) mostra uma extração de um arquivo de comando como `myprog.in` e [Figura 2 na página 17](#) mostra a extração correspondente da saída de um arquivo de relatório como `results.out`.

Como proceder a seguir

Para obter detalhes completos da sintaxe de comando **runmqsc**, parâmetros opcionais e códigos de retorno, consulte [runmqsc \(executar comandos MQSC\)](#).

Tarefas relacionadas

“Configurando o prompt de comandos MQSC” na página 20

No AIX, Linux, and Windows, use a variável de ambiente **MQPROMPT** para configurar o prompt que é exibido ao executar o comando **runmqsc**. Isso facilita ver que você está em um ambiente MQSC e ver alguns detalhes do ambiente atual.

“Executando comandos MQSC interativamente em runmqsc” na página 18

No AIX, Linux, and Windows, é possível usar o prompt de comandos **runmqsc** para emitir comandos MQSC para um gerenciador de filas interativamente. A execução interativa é particularmente adequada para testes rápidos

Referências relacionadas

[Referência de comandos MQSC](#)

Multi Configuração automática por meio de um script MQSC na inicialização

É possível configurar seu gerenciador de filas para aplicar automaticamente o conteúdo de um script MQSC ou conjunto de scripts MQSC em cada início do gerenciador de filas.

É possível usar essa funcionalidade para ter uma configuração que pode ser modificada e reproduzida automaticamente na próxima reinicialização do gerenciador de filas. Como exemplo, se o script ou scripts estão localizados em uma unidade montada, é possível ter uma configuração centralizada em que a versão mais recente é aplicada a cada gerenciador de filas conforme são iniciados.

Um determinado cenário no qual isso pode ser útil é garantir que um cluster uniforme contenha as mesmas definições em todos os gerenciadores de filas no cluster, tendo um único conjunto de configuração que eles todos aplicam. Para obter um exemplo disso, consulte [Criando um novo cluster uniforme](#).

Antes de Começar

É possível utilizar:

1. Um único script e criar um arquivo de texto usando comandos MQSC.
2. Um conjunto de scripts MQSC:
 - Para identificar um diretório em que as configurações existirão e
 - Nesse diretório, crie arquivos, cada um com a extensão `.mqsc`, por exemplo, `queues.mqsc`.

Uma vez que esse script é reaplicado em cada início do gerenciador de filas, é importante que os comandos possam ser reproduzidos. Por exemplo, um comando **DEFINE** deve incluir a sequência **REPLACE**, caso contrário, o comando aparecerá como uma falha no início do segundo gerenciador de filas, já que o objeto já existe.

Observe que em um script MQSC, qualquer linha prefixada com `*` é tratada como um comentário.

Ativando a configuração automática de scripts MQSC

Importante: Não deve-se emitir comandos para canais do tipo MQTT, pois eles não são suportados para configuração automática durante a inicialização

É possível configurar um novo gerenciador de filas usando a sinalização **-ic** para o comando **crtmqm** e apontando para um arquivo ou um diretório específico. O valor fornecido é armazenado no arquivo `qm.ini`, na sub-rotina `AutoConfig`, como o atributo **MQSCConfig**.

É possível configurar um gerenciador de filas existente para ativar a configuração automática do MQSC, incluindo o atributo de sub-rotina **MQSCConfig** de AutoConfig, apontando para um arquivo ou um diretório válido. Por exemplo:

```
AutoConfig:
MQSCConfig=C:\mq_configuration\uniclus.mqsc
```

Como a configuração automática funciona?

Durante a inicialização do gerenciador de filas, a configuração identificada pelo **AutoConfig** atributo de sub-rotina **MQSCConfig** é passada por meio da validação **runmqsc** para assegurar uma sintaxe válida e, em seguida, armazenada na árvore de dados do gerenciador de filas no diretório `autocfg` como um único arquivo `cached.mqsc`.

Quando múltiplos arquivos de um diretório são processados, eles são processados em ordem alfabética, e se ele contém um comando de término ou encerramento de MQSC, o resto do conteúdo desse arquivo é ignorado.

Durante o primeiro início do gerenciador de filas, uma incapacidade de ler o arquivo ou diretório ou um arquivo com sintaxe MQSC que não é válido impede o gerenciador de filas de ser iniciado com uma mensagem de erro apropriada tanto para o console quanto para o log de erro do gerenciador de filas.

Em reinicializações subsequentes, se o arquivo ou diretório apontado estiver ilegível ou contiver sintaxe MQSC inválida, o arquivo armazenado em cache anteriormente será usado e uma mensagem gravada no log de erro do gerenciador de filas destacará isso.

No ponto em que o conteúdo do `cached.mqsc` é aplicado no gerenciador de fila, quando todos os comandos MQSC foram aplicados, o gerenciador de filas é ativado para que os aplicativos se conectem. O log **runmqsc** da configuração que está sendo aplicada é armazenado no diretório de erros do gerenciador de filas, como um arquivo chamado `autocfgmqsc.LOG`.

Além disso, qualquer comando MQSC que não seja concluído com sucesso é registrado no log de erros do gerenciador de filas, identificando por que o comando falha.

Automatizando a administração do IBM MQ usando comandos PCF

É possível concluir que será benéfico para sua instalação automatizar algumas tarefas de administração e monitoramento. É possível automatizar tarefas de administração para gerenciadores de filas locais e remotas usando comandos `programmable command format` (PCF). Esta seção supõe que você tenha experiência para administrar objetos do IBM MQ.

comandos PCF

Comandos `programmable command format` (PCF) do IBM MQ podem ser utilizados para as tarefas de administração do programa para um programa de administração. Desta maneira, a partir de um programa é possível manipular os objetos do gerenciador de filas (filas, definições do processo, listas, canais, canais de conexão do cliente, listeners, serviços e objetos das informações de autenticação) e até manipular os próprios gerenciadores de fila.

Os comandos PCF abrangem a mesma faixa de funções fornecidas pelos comandos MQSC. É possível gravar um programa para emitir comandos PCF para qualquer gerenciador de filas na rede a partir de um único nó. Desta maneira, é possível centralizar e automatizar as tarefas de administração.

Cada comando PCF é uma estrutura de dados que está integrado na parte de dados do aplicativo em uma mensagem do IBM MQ. Cada comando é enviado ao gerenciador de filas de destino usando a função `MQI MQPUT` da mesma maneira que qualquer outra mensagem. Desde que o servidor de comandos esteja em execução no gerenciador de filas que recebe a mensagem, o servidor de comandos interpreta como uma mensagem de comando e executa o comando. Para obter as respostas, o aplicativo emite uma chamada `MQGET` e os dados de resposta são retornados em outra estrutura de dados. O aplicativo pode então processar a resposta e agir de acordo.

Nota: Diferente de comandos MQSC, os comandos PCF e suas respostas não estão em um formato de texto que você possa ler.

Resumidamente, estas são algumas das coisas necessárias para criar uma mensagem de comando PCF:

Descritor de Mensagens

Esse é um descritor de mensagem padrão do IBM MQ, no qual:

- Tipo de mensagem (*MsgType*) é MQMT_REQUEST.
- Formato da mensagem (*Format*) é MQFMT_ADMIN.

Dados do aplicativo

Contém a mensagem PCF que inclui o cabeçalho PCF, em que:

- O tipo de mensagem PCF (*Type*) especifica MQCFT_COMMAND.
- O identificador de comando especifica o comando, por exemplo, *Change Queue* (MQCMD_CHANGE_Q)

Para obter uma descrição completa das estruturas de dados PCF e como implementá-las, consulte [“Introdução aos formatos de comando programável do IBM MQ” na página 26.](#)

Atributos de Objeto PCF

Os atributos do objeto no PCF não são limitados a oito caracteres como se fossem para comandos MQSC. Eles são mostrados neste guia em itálico. Por exemplo, o PCF equivalente de RQMNAME for *RemoteQMGrName*.

PCFs de Escape

PCFs de escape são comandos PCF que contêm comandos MQSC no texto de mensagem. É possível usar PCFs para enviar comandos para um gerenciador de filas remotas. Para obter mais informações sobre PCFs de escape, consulte [Escape](#).

Introdução aos formatos de comando programável do IBM MQ

Os Formatos de Comando Programável (PCFs) definem as mensagens de resposta e comando que podem ser trocadas entre um programa e qualquer gerenciador de filas (que suporta PCFs) em uma rede. Os PCFs simplificam a administração do gerenciador de filas e outra administração de rede. Eles podem ser usados para resolver o problema da administração complexa de redes distribuídas especialmente conforme as redes crescem de tamanho e complexidade.

Os formatos de comando programáveis são suportados em todas as plataformas IBM MQ .

Problemas que os comandos PCF resolvem

A administração de redes distribuídas pode ser complexa. Os problemas de administração continuam crescendo conforme as redes aumentam de tamanho e complexidade.

Exemplos de administração específica para o sistema de mensagens e enfileiramento incluem:

- Gerenciamento de recurso.

Por exemplo, a criação e exclusão da fila.

- Monitoramento de desempenho.

Por exemplo, profundidade da fila máxima ou taxa de mensagens.

- Controle.

Por exemplo, ajustando parâmetros de fila como a profundidade máxima da fila, comprimento máximo de mensagem e ativação e desativação de filas.

- Roteamento de mensagem.

Definição de rotas alternativas por meio de uma rede.

Os comandos podem ser utilizados para PCF do IBM MQ simplificam a administração do gerenciador de filas e outra administração de rede. Os comandos PCF permitem usar um único aplicativo para executar a administração de rede de um único gerenciador de filas na rede.

O que são PCFs?

Os PCFs definem as mensagens de resposta e comando que podem ser trocadas entre um programa e qualquer gerenciador de filas (que suporta PCFs) em uma rede. É possível usar os comandos PCF em um programa de aplicativo de gerenciamento de sistemas para administração de objetos IBM MQ: objetos de informações de autenticação, canais, listeners de canais, listas de nomes, definições de processo, gerenciadores de fila, filas, serviços e classes de armazenamento. O aplicativo pode operar a partir de um único ponto na rede para comunicar informações de resposta e comando com qualquer gerenciador de filas, local ou remoto, usando o gerenciador de filas locais.


Cada gerenciador de filas tem uma fila de administração com um nome de fila padrão e seu aplicativo pode enviar mensagens de comando PCF para essa fila. Cada gerenciador de filas também tem um servidor de comandos para atender as mensagens de comando da fila de administração. Portanto, as mensagens de comando PCF podem ser processadas por qualquer gerenciador de filas na rede e os dados de resposta podem ser retornados ao seu aplicativo, usando a sua fila de resposta especificada. As mensagens de resposta e comandos PCF são enviados e recebidos usando a Message Queue Interface (MQI) normal.

Para obter uma lista dos comandos PCF disponíveis, incluindo seus parâmetros, consulte [Definições dos formatos de comando programáveis](#).

Usando os formatos de comando programável do IBM MQ

É possível usar PCFs em um programa de gerenciamento de sistemas para administração remota do IBM MQ.

Esta seção inclui:


- [“Mensagens de comando PCF” na página 27](#)
- [“Respostas PCF no IBM MQ” na página 30](#)
-  [“Extended responses” na página 32](#)
- [Regras para nomenclatura de objetos IBM MQ](#)
- [“Verificação de autoridade para comandos PCF no IBM MQ” na página 33](#)


Mensagens de comando PCF

Mensagens de comando PCF consistem em um cabeçalho PCF, parâmetros identificados nesse cabeçalho e também em dados da mensagem definidos pelo usuário. As mensagens são emitidas utilizando chamadas de interface de Fila de Mensagens.

Cada comando e seus parâmetros são enviados como uma mensagem de comando separada contendo um cabeçalho PCF seguido por um número de estruturas de parâmetros; para obter detalhes sobre o cabeçalho PCF, consulte [MQCFH – cabeçalho PCF](#) e para um exemplo de uma estrutura do parâmetro, consulte [MQCFST – sequência do parâmetro PCF](#). O cabeçalho PCF identifica o comando e o número de estruturas de parâmetro que seguem na mesma mensagem. Cada estrutura do parâmetro fornece um parâmetro para o comando.

As respostas para os comandos geradas pelo servidor de comandos têm uma estrutura semelhante. Há um cabeçalho PCF, seguido por um número de estruturas de parâmetros. Respostas podem consistir em mais de uma mensagem, mas os comandos sempre consistem em uma única mensagem.

 No [Multiplataformas](#), a fila para a qual os comandos PCF são enviados é sempre chamada de `SYSTEM.ADMIN.COMMAND.QUEUE`.

 No [z/OS](#), comandos são enviados para `SYSTEM.COMMAND.INPUT`, embora `SYSTEM.ADMIN.COMMAND.QUEUE` pode ser um alias para ele. O servidor de comandos que atende a

essa fila envia as respostas para a fila definida pelos campos *ReplyToQ* e *ReplyToQMgr* no descritor de mensagem da mensagem de comando.

Como emitir mensagens de comando PCF

Utilize as chamadas normais do Message Queue Interface (MQI), MQPUT, MQGET e assim por diante, para colocar e recuperar o comando PCF e as mensagens de resposta para e a partir de suas filas.

Nota:

Certifique-se de que o servidor de comandos esteja em execução no gerenciador de filas de destino para que o comando PCF processe nesse gerenciador de filas.

Para obter uma lista de arquivos de cabeçalho fornecidos, consulte [IBM MQ COPY, cabeçalho, inclua e arquivos de módulo](#).

Descritor de mensagens para um comando PCF

O descritor de mensagens do IBM MQ está totalmente documentado no [MQMD – Descritor de Mensagens](#).

Uma mensagem de comando PCF contém os seguintes campos no descritor de mensagens:

Relatório

Qualquer valor válido, conforme necessário.

MsgType

Este campo deve ser MQMT_REQUEST para indicar uma mensagem que requer uma resposta.

Expiração

Qualquer valor válido, conforme necessário.

Feedback

Configure para MQFB_NONE

Codificação

Se você estiver enviando para um sistema IBM MQ for Multiplatforms, configure este campo para a codificação usada para os dados da mensagem. A conversão é realizada se necessário.

CodedCharSetId

Se você estiver enviando para um sistema IBM MQ for Multiplatforms, configure este campo para o identificador do conjunto de caracteres codificados utilizado para os dados da mensagem. A conversão é realizada se necessário.

Formato

Configure para MQFMT_ADMIN.

Priority

Qualquer valor válido, conforme necessário.

Persistence

Qualquer valor válido, conforme necessário.

MsgId

O aplicativo de envio pode especificar qualquer valor ou MQMI_NONE pode ser especificado para solicitar ao gerenciador de filas que gere um identificador de mensagem exclusivo.

CorrelId

O aplicativo de envio pode especificar qualquer valor ou MQCI_NONE pode ser especificado para indicar nenhum identificador de correlação.

ReplyToQ

O nome da fila para receber a resposta.

ReplyToQMgr

O nome do gerenciador de filas para a resposta (ou em branco).

Campos de contexto da mensagem

Esses campos podem ser configurados para quaisquer valores válidos, conforme necessário. Normalmente a opção MQPMO_DEFAULT_CONTEXT de inserção de mensagem é utilizada para definir os campos de contexto da mensagem para os valores padrão.

Se você estiver utilizando uma estrutura MQMD da versão 2, deve-se configurar os campos adicionais a seguir:

GroupId

Configure para MQGI_NONE

MsgSeqNumber

Configurado para 1

Offset

Configurado para 0

MsgFlags

Configure para MQMF_NONE

OriginalLength

Configure como MQOL_UNDEFINED

Enviando dados do usuário

As estruturas PCF também podem ser utilizadas para enviar dados da mensagem definidos pelo usuário. Neste caso, o campo descritor de mensagens *Format* deve ser configurado como MQFMT_PCF.

Enviando e recebendo mensagens PCF em uma fila especificada

Enviando mensagens PCF para uma fila especificada

Para enviar uma mensagem para uma fila especificada, a chamada mqPutBag converte o conteúdo do pacote especificado em uma mensagem PCF e envia a mensagem para a fila especificada. O conteúdo do pacote permanece inalterado após a chamada.

Como entrada para essa chamada, deve-se fornecer:

- Um MQI identificador de conexão.
- Um identificador de objeto para a fila na qual a mensagem deve ser colocada.
- Um descritor de mensagens. Para obter mais informações sobre o descritor de mensagens, consulte [MQMD – Descritor de Mensagens](#).
- Opções de Colocação de Mensagens utilizando a estrutura MQPMO. Para obter mais informações sobre a estrutura MQPMO, consulte [MQPMO Put-opções de mensagens](#).
- A alça do pacote a ser convertidos em uma mensagem.

Nota: Se o pacote contém uma mensagem de administração e a chamada mqAddInquiry foi utilizada para inserir valores no pacote, o valor do item de dados MQIASY_COMMAND deve ser um comando INQUIRE reconhecido pelo MQAI.

Para obter uma descrição completa da chamada mqPutBag, consulte [mqPutBag](#).

Recebendo mensagens PCF de uma fila especificada

Para receber uma mensagem de uma fila especificada, a chamada mqGetBag obtém uma mensagem PCF a partir de uma fila especificada e converte os dados da mensagem em um pacote de dados.

Como entrada para essa chamada, deve-se fornecer:

- Um MQI identificador de conexão.

- Uma manipulação de objetos da fila a partir da qual a mensagem deve ser lida.
- Um descritor de mensagens. Dentro da estrutura MQMD, o parâmetro **Format** deve ser MQFMT_ADMIN, MQFMT_EVENT ou MQFMT_PCF.

Nota: Se a mensagem for recebida em uma unidade de trabalho (ou seja, com a opção MQGMO_SYNCPOINT) e a mensagem possuir um formato não suportado, a unidade de trabalho poderá ser desfeita. A mensagem é então restabelecida na fila e pode ser recuperada através da chamada MQGET em vez de a chamada mqGetBag. Para obter informações adicionais sobre o descritor de mensagens, consulte [MQGMO Get-opções de mensagem](#).

- Opções de obtenção de mensagens utilizando a estrutura MQGMO. Para obter mais informações sobre a estrutura MQGMO, consulte [MQMD – Descritor de Mensagens](#).
- A alça do pacote para conter a mensagem convertida.

Para obter uma descrição completa da chamada mqGetBag, consulte [mqGetBag](#).

Respostas PCF no IBM MQ

Em resposta a cada comando, o servidor de comandos gerará uma ou mais mensagens de resposta. Uma mensagem de resposta possui um formato semelhante a uma mensagem de comando.

O cabeçalho PCF possui o mesmo valor identificador de comando que o comando para o qual ele é uma resposta (consulte MQCFH - Cabeçalho PCF para obter detalhes). O identificador de mensagem e o identificador de correlação são configurados de acordo com as opções de relatório da solicitação.

Se o tipo de cabeçalho PCF da mensagem de comando for MQCFT_COMMAND, somente as respostas padrão serão geradas. Tais comandos são suportados apenas em Multiplataformas (Multiplatforms) Aplicativos mais antigos não suportam PCF no z/OS ; o IBM MQ Windows Explorer é um desses aplicativos (no entanto, o IBM WebSphere MQ 6.0 ou posterior IBM MQ Explorer suporta PCF no z/OS).

Se o tipo de cabeçalho PCF da mensagem de comando é MQCFT_COMMAND_XR, respostas estendidas ou padrão são geradas. Tais comandos são suportados no z/OS e em algumas Multiplataformas Comandos emitidos em z/OS geram apenas respostas estendidas.

Se um comando único especifica um nome de objeto genérico, uma resposta separada é retornada em sua própria mensagem para cada objeto correspondente. Para geração de resposta, um comando único com um nome genérico é tratado como vários comandos individuais (exceto para o campo de controle MQCFC_LAST ou MQCFC_NOT_LAST). Caso contrário, uma mensagem de comando gera uma mensagem de resposta.

Algumas respostas PCF podem retornar uma estrutura mesmo quando não solicitada. Essa estrutura é mostrada na definição da resposta ([Definições dos formatos de comando programáveis](#)) como *sempre retornado*. O motivo é que, para essas respostas, é necessário nomear os objetos na resposta para identificar a qual objeto os dados se aplicam.

Descritor de mensagens para uma resposta

Uma mensagem de resposta possui os seguintes campos no descritor de mensagens:

MsgType

Este campo é MQMT_REPLY.

MsgId

Este campo é gerado pelo gerenciador de filas.

CorrelId

Este campo é gerado de acordo com as opções de relatório da mensagem de comando.

Formato

Este campo é MQFMT_ADMIN.

Codificação

Configure para MQENC_NATIVE.

CodedCharSetId

Configure para MQCCSI_Q_MGR.

Persistence

O mesmo que na mensagem de comando.

Priority

O mesmo que na mensagem de comando.

A resposta é gerada com MQPMO_PASS_IDENTITY_CONTEXT.

Multi Respostas padrão

Mensagens de comando com um tipo de cabeçalho de MQCFT_COMMAND, as respostas padrão são geradas. Tais comandos são suportados apenas em Multiplataformas

Há três tipos de resposta padrão:

- Resposta OK
- Resposta de Erro
- Resposta de dados

Resposta OK

Esta resposta consiste em uma mensagem que começa com um cabeçalho em formato de comando, com um campo *CompCode* de MQCC_OK ou MQCC_WARNING.

Para MQCC_OK, o *Reason* é MQRC_NONE.

Para MQCC_WARNING, o *Reason* identifica a natureza do aviso. Neste caso, o cabeçalho de formato de comando pode ser seguido por uma ou mais estruturas de parâmetro de aviso apropriadas para esse código de razão.

Em qualquer caso, para um comando de consulta, mais estruturas de parâmetro podem seguir, conforme descrito nas seções a seguir.

Resposta de Erro

Se o comando tiver um erro, uma ou mais mensagens de resposta de erro são enviadas (mais de uma pode ser enviada mesmo para um comando que normalmente teria somente uma única mensagem de resposta). Essas mensagens de resposta de erro têm MQCFC_LAST ou MQCFC_NOT_LAST configurado conforme apropriado.

Cada mensagem começa com um cabeçalho em formato de resposta, com um valor *CompCode* de MQCC_FAILED e um campo *Reason* que identifica o erro específico. Em geral, cada mensagem descreve um erro diferente. Além disso, cada mensagem tem zero ou uma (nunca mais de uma) estrutura de parâmetro de erro após o cabeçalho. Esta estrutura de parâmetro, se houver uma, é uma estrutura MQCFIN, com um campo *Parameter* que contém um dos seguintes:

- MQIACF_PARAMETER_ID

O campo *Value* na estrutura é o identificador do parâmetro que estava em erro (por exemplo, MQCA_Q_NAME).

- MQIACF_ERROR_ID

Esse valor é utilizado com um valor *Reason* (no cabeçalho em formato de comando) de MQRC_UNEXPECTED_ERROR. O campo *Value* na estrutura MQCFIN é o código de razão inesperado recebido pelo servidor de comandos.

- MQIACF_SELECTOR

Esse valor ocorre se uma estrutura de lista (MQCFIL) enviada com o comando contém um seletor duplicado ou um que não é válido. O campo *Reason* no cabeçalho em formato de comando identifica o erro e o campo *Value* na estrutura MQCFIN é o valor do parâmetro na estrutura MQCFIL do comando que estava em erro.

- MQIACF_ERROR_OFFSET

Esse valor ocorre quando há um erro de comparação de dados no comando Ping Channel. O campo *Value* na estrutura é o deslocamento do erro de comparação Ping Channel.

- MQIA_CODED_CHAR_SET_ID

Esse valor ocorre quando o identificador de conjunto de caracteres codificado no descritor de mensagens da mensagem de comando PCF recebida não corresponde àquela do gerenciador de filas de destino. O campo *Value* na estrutura é o identificador do conjunto de caracteres codificado do gerenciador de filas.

A última (ou única) mensagem de resposta de erro é uma resposta de resumo, com um campo *CompCode* de MQCC_FAILED e um campo *Reason* de MQRCCF_COMMAND_FAILED. Essa mensagem não possui estrutura de parâmetros após o cabeçalho.

Resposta de dados

Essa resposta consiste em uma resposta OK (conforme descrito anteriormente) para um comando de consulta. A resposta OK é seguida por estruturas adicionais contendo os dados solicitados conforme descrito em [Definições dos formatos de comando programáveis](#).

Aplicativos não devem depender dessas estruturas de parâmetro adicionais que estão sendo retornadas em qualquer ordem particular.

Extended responses

Commands issued on z/OS generate extended responses.

There are three types of extended response:

- Message response, with type MQCFT_XR_MSG
- Item response, with type MQCFT_XR_ITEM
- Summary response, with type MQCFT_XR_SUMMARY

Each command can generate one, or more, sets of responses. Each set of responses comprises one or more messages, numbered sequentially from 1 in the *MsgSeqNumber* field of the PCF header. The *Control* field of the last (or only) response in each set has the value MQCFC_LAST. For all other responses in the set, this value is MQCFC_NOT_LAST.

Any response can include one, or more, optional MQCFBS structures in which the *Parameter* field is set to MQBACF_RESPONSE_SET, the value being a response set identifier. Identifiers are unique and identify the set of responses which contain the response. For every set of responses, there is an MQCFBS structure that identifies it.

Extended responses have at least two parameter structures:

- An MQCFBS structure with the *Parameter* field set to MQBACF_RESPONSE_ID. The value in this field is the identifier of the set of responses to which the response belongs. The identifier in the first set is arbitrary. In subsequent sets, the identifier is one previously notified in an MQBACF_RESPONSE_SET structure.
- An MQCFST structure with the *Parameter* field set to MQCACF_RESPONSE_Q_MGR_NAME, the value being the name of the queue manager from which the set of responses come.

Many responses have additional parameter structures, and these structures are described in the following sections.

You cannot determine in advance how many responses there are in a set other than by getting responses until one with MQCFC_LAST is found. Neither can you determine in advance how many sets of responses there are as any set might include MQBACF_RESPONSE_SET structures to indicate that additional sets are generated.

Extended responses to Inquire commands

Inquire commands normally generate an item response (type MQCFT_XR_ITEM) for each item found that matches the specified search criteria. The item response has a *CompCode* field in the header

with a value of MQCC_OK, and a *Reason* field with a value of MQRC_NONE. It also includes other parameter structures describing the item and its requested attributes, as described in [Definitions of the Programmable Command Formats](#).

If an item is in error, the *CompCode* field in the header has a value of MQCC_FAILED and the *Reason* field identifies the particular error. Additional parameter structures are included to identify the item.

Certain Inquire commands might return general (not name-specific) message responses in addition to the item responses. These responses are informational, or error, responses of the type MQCFT_XR_MSG.

If the Inquire command succeeds, there might, optionally, be a summary response (type MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_OK, and a *Reason* field value of MQRC_NONE.

If the Inquire command fails, item responses might be returned, and there might optionally be a summary response (type MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_FAILED, and a *Reason* field value of MQRCCF_COMMAND_FAILED.

Extended responses to commands other than Inquire

Successful commands generate message responses in which the *CompCode* field in the header has a value of MQCC_OK, and the *Reason* field has a value of MQRC_NONE. There is always at least one message; it might be informational (MQCFT_XR_MSG) or a summary (MQCFT_XR_SUMMARY). There might optionally be additional informational (type MQCFT_XR_MSG) messages. Each informational message might include a number of additional parameter structures with information about the command; see the individual command descriptions for the structures that can occur.

Commands that fail generate error message responses (type MQCFT_XR_MSG), in which the *CompCode* field in the header has a value of MQCC_FAILED and the *Reason* field identifies the particular error. Each message might include a number of additional parameter structures with information about the error: see the individual error descriptions for the structures that can occur. Informational message responses might be generated. There might, optionally, be a summary response (MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_FAILED, and a *Reason* field value of MQRCCF_COMMAND_FAILED.

Extended responses to commands using CommandScope

If a command uses the **CommandScope** parameter, or causes a command using the **CommandScope** parameter to be generated, there is an initial response set from the queue manager where the command was received. Then a separate set, or sets, of responses is generated for each queue manager to which the command is directed (as if multiple individual commands were issued). Finally, there is a response set from the receiving queue manager which includes an overall summary response (type MQCFT_XR_SUMMARY). The MQCACF_RESPONSE_Q_MGR_NAME parameter structure identifies the queue manager that generates each set.


The initial response set has the following additional parameter structures:

- MQIACF_COMMAND_INFO (MQCFIN). Possible values in this structure are MQCMDI_CMDScope_ACCEPTED or MQCMDI_CMDScope_GENERATED.
- MQIACF_CMDScope_Q_MGR_COUNT (MQCFIN). This structure indicates the number of queue managers to which the command is sent.

Verificação de autoridade para comandos PCF no IBM MQ

Quando um comando PCF for processado, o *UserIdentifier* do descritor de mensagens na mensagem de comando é utilizado para as verificações de autoridade de objeto do IBM MQ. A verificação de autoridade é implementada de forma diferente em cada plataforma, conforme descrito neste tópico.

As verificações são executadas no sistema no qual o comando está sendo processado; portanto, esse ID de usuário deve existir no sistema de destino e possuir as autoridades necessárias para processar o comando. Se a mensagem chegou de um sistema remoto, uma forma de conseguir o ID existente no sistema de destino deve ter um ID do usuário correspondente em ambos os sistemas local e remoto.

Nota:  Para obter informações sobre a verificação de autoridade no z/OS, consulte [Tarefa 1: identificar os parâmetros do sistema z/OS](#).

IBM MQ for IBM i



Para processar qualquer comando PCF, o ID do usuário deve ter autoridade *dsp* para o IBM MQ de objeto no sistema de destino.

Além disso, as verificações da autoridade de objeto do IBM MQ são executadas para determinados comandos PCF, conforme mostrado em [Tabela 2 na página 35](#).

Na maioria dos casos, estas verificações são as mesmas verificações que as verificações executadas pelos comandos CL equivalentes do IBM MQ emitidos em um sistema local. Consulte [Configurando segurança no IBM i](#), para obter informações adicionais sobre o mapeamento a partir das autoridades do IBM MQ para autoridades do sistema do IBM i e os requisitos de autoridade para os comandos CL do IBM MQ. Os detalhes sobre saídas referentes à segurança são fornecidos na documentação [Segurança em nível de link utilizando uma saída de segurança](#).

Para processar qualquer um dos seguintes comandos o ID do usuário deverá ser um membro do perfil do grupo QMQMADM:

- Executar ping no Canal
- Alterar Canal
- Copiar Canal
- Criar Canal
- Excluir Canal
- Redefinir Canal
- Resolver Canal
- Iniciar o Canal
- Parar Canal
- Iniciar Inicializador de Canal
- Iniciar Ouvinte de Canal

IBM MQ for UNIX, Linux, and Windows



Para processar qualquer comando do PCF, o ID do usuário deve ter autoridade *dsp* para o objeto de gerenciador de filas no sistema de destino. Além disso, as verificações da autoridade de objeto do IBM MQ são executadas para determinados comandos PCF, conforme mostrado em [Tabela 2 na página 35](#).

Para processar qualquer um dos seguintes comandos o ID do usuário deve pertencer ao grupo *mqm*.

Nota: Para o Windows **somente**, o ID do usuário pode pertencer ao grupo *Administradores* ou grupo *mqm*.

- Alterar Canal
- Copiar Canal
- Criar Canal
- Excluir Canal
- Executar ping no Canal
- Redefinir Canal
- Iniciar o Canal
- Parar Canal
- Iniciar Inicializador de Canal

- Iniciar Ouvinte de Canal
- Resolver Canal
- Reconfigurar Cluster
- Refresh Cluster
- Suspende Gerenciador de Filas
- Retomar Gerenciador de Filas

Autoridades de objeto do IBM MQ para multiplataformas



<i>Tabela 2. Autoridades do objeto</i>		
Comando:	Autoridade de objeto IBM MQ	Autoridade de classe (para tipo de objeto)
Alterar Informações sobre Autenticação	dsp e chg	n/a
Alterar Canal	dsp e chg	n/a
Mudar ouvinte de canal	dsp e chg	n/a
Mudar canal de conexão do cliente	dsp e chg	n/a
Alterar Lista de Nomes	dsp e chg	n/a
Processo de Mudança	dsp e chg	n/a
Alterar a Fila	dsp e chg	n/a
Alterar Gerenciador de Filas	chg <i>consulte Nota 3 e Nota 5</i>	n/a
mudar Serviço	dsp e chg	n/a
Limpar Fila	clr	n/a
Copiar Informações sobre Autenticação	dsp	crt
Copiar Informações de Autenticação (Substituir) <i>consulte a Nota 1</i>	<i>de: dsp para: chg</i>	crt
Copiar Canal	dsp	crt
Copiar Canal (Substituir) <i>consulte a Nota 1</i>	<i>de: dsp para: chg</i>	crt
Copiar ouvinte de canal	dsp	crt
Copiar ouvinte de canal (Substituir) <i>consulte a Nota 1</i>	<i>de: dsp para: chg</i>	crt
Copiar canal de conexão do cliente	dsp	crt
Copiar canal de conexão do cliente (Substituir) <i>consulte a Nota 1</i>	<i>de: dsp para: chg</i>	crt
Copiar Lista de Nomes	dsp	crt

<i>Tabela 2. Autoridades do objeto (continuação)</i>		
Comando:	Autoridade de objeto IBM MQ	Autoridade de classe (para tipo de objeto)
Copiar lista de nomes (Substituir) <i>consulte a Nota 1</i>	<i>a partir de:</i> dsp <i>para:</i> dsp e chg	crt
Copiar processo	dsp	crt
Copiar processo (substituir) <i>consulte a Nota 1</i>	<i>de:</i> dsp <i>para:</i> chg	crt
Copiar Fila	dsp	crt
Copiar fila (substituir) <i>consulte a Nota 1</i>	<i>a partir de:</i> dsp <i>para:</i> dsp e chg	crt
Criar Informações sobre Autenticação	<i>(informações de autenticação padrão do sistema)</i> dsp	crt
Criar informações de autenticação (substituir) <i>consulte a Nota 1</i>	<i>(informações de autenticação padrão do sistema)</i> dsp <i>para:</i> chg	crt
Criar Canal	<i>(canal padrão do sistema)</i> dsp	crt
Crie canal (substituir) <i>consulte a Nota 1</i>	<i>(canal padrão do sistema)</i> dsp <i>para:</i> chg	crt
Criar ouvinte de canal	<i>(listener padrão do sistema)</i> dsp	crt
Criar ouvinte de canal (substituir) <i>consulte a Nota 1</i>	<i>(listener padrão do sistema)</i> dsp <i>para:</i> chg	crt
Criar canal de conexão do cliente	<i>(canal padrão do sistema)</i> dsp	crt
Criar canal de conexão do cliente (substituir) <i>consulte a Nota 1</i>	<i>(canal padrão do sistema)</i> dsp <i>para:</i> chg	crt
Criar Lista de Nomes	<i>(lista de nomes padrão do sistema)</i> dsp	crt
Criar lista de nomes (substituir) <i>consulte a Nota 1</i>	<i>(lista de nomes padrão do sistema)</i> dsp <i>para:</i> dsp e chg	crt
Criar processo	<i>(processo padrão do sistema)</i> dsp	crt
Criar processo (substituir) <i>consulte a Nota 1</i>	<i>(processo padrão do sistema)</i> dsp <i>para:</i> chg	crt
Criar fila	<i>(fila padrão do sistema)</i> dsp	crt
Criar fila (substituir) <i>consulte a Nota 1</i>	<i>(fila padrão do sistema)</i> dsp <i>para:</i> dsp e chg	crt
Criar Serviço	<i>(fila padrão do sistema)</i> dsp	crt
Criar serviço (substituir) <i>consulte a Nota 1</i>	<i>(fila padrão do sistema)</i> dsp <i>para:</i> chg	crt
Excluir Informações sobre Autenticação	dsp e dlt	n/a
Excluir Registro de Autoridade	<i>(objeto do gerenciador de filas)</i> chg <i>consulte a Nota 4</i>	<i>consulte a Nota 4</i>
Excluir Canal	dsp e dlt	n/a

Tabela 2. Autoridades do objeto (continuação)

Comando:	Autoridade de objeto IBM MQ	Autoridade de classe (para tipo de objeto)
Excluir ouvinte de canal	dsp e dlt	n/a
Excluir canal de conexão do cliente	dsp e dlt	n/a
Delete Namelist	dsp e dlt	n/a
Excluir Processo	dsp e dlt	n/a
Excluir fila	dsp e dlt	n/a
Excluir Serviço	dsp e dlt	n/a
Consultar Informações sobre Autenticação	dsp	n/a
Consultar Registros de Autoridade	<i>consulte a Nota 4</i>	<i>consulte a Nota 4</i>
Consultar Canal	dsp	n/a
Consultar ouvinte de canal	dsp	n/a
Consultar status do canal (para ChannelType MQCHT_CLSSDR)	inq	n/a
Consultar canal de conexão do cliente	dsp	n/a
Consultar Lista de Nomes	dsp	n/a
Consultar Processo	dsp	n/a
Consultar Fila	dsp	n/a
Consultar Gerenciador de Filas	<i>consulte a nota 3</i>	n/a
Consultar Status da Fila	dsp	n/a
Consultar Serviço	dsp	n/a
Executar ping no Canal	ctrl	n/a
Executar Ping do Gerenciador de Filas	<i>consulte a nota 3</i>	n/a
Atualizar Gerenciador de Filas	(objeto do gerenciador de filas) chg	n/a
Atualizar segurança (para SecurityType MQSECTYPE_SSL)	(objeto do gerenciador de filas) chg	n/a
Redefinir Canal	ctrlx	n/a
Reconfigurar Gerenciador de Filas	(objeto do gerenciador de filas) chg	n/a
Reconfigurar as Estatísticas de Fila	dsp e chg	n/a
Resolver Canal	ctrlx	n/a
Configurar Registro de Autoridade	<i>(objeto do gerenciador de filas) chg</i> <i>consulte a Nota 4</i>	<i>consulte a Nota 4</i>

Tabela 2. Autoridades do objeto (continuação)

Comando:	Autoridade de objeto IBM MQ	Autoridade de classe (para tipo de objeto)
Iniciar o Canal	ctrl	n/a
Parar Canal	ctrl	n/a
Para Conexão	(objeto do gerenciador de filas) chg	n/a
Iniciar Atendente	ctrl	n/a
Parar Atendente	ctrl	n/a
Iniciar Serviço	ctrl	n/a
Parar Serviços	ctrl	n/a
Escape	consulte a Nota 2	consulte a Nota 2

Notas:

1. Este comando aplica-se se o objeto a ser substituído existir, caso contrário, a verificação de autoridade é como para Criar ou Copiar sem Substituir.
2. A autoridade necessária é determinada pelo comando MQSC definido pelo texto de escape e é equivalente a um dos comandos anteriores.
3. Para processar qualquer comando PCF, o ID do usuário deve ter autoridade de dsp para o objeto de gerenciador de filas no sistema de destino.
4. Este comando PCF for autorizado a menos que o servidor de comandos foi iniciado com o parâmetro -a. Por padrão, o servidor de comandos é iniciado quando o gerenciador de filas é iniciado e sem o parâmetro -a. Para obter mais informações, veja [Referência de formatos de comando programável](#).
5. Conceder a um ID do usuário a autoridade chg para um gerenciador de filas fornece a capacidade de configurar os registros de autoridade para todos os grupos e usuários. Não conceda essa autoridade para usuários ou aplicativos comuns.

IBM MQ também fornece alguns pontos de saída de segurança do canal para que você possa fornecer seus próprios programas de saída de usuário para verificação de segurança. Para obter mais informações, veja [Exibindo um canal](#).

Multi

Usando o MQAI para simplificar o uso de PCFs

O IBM MQ Administration Interface (MQAI) é uma interface de programação para o IBM MQ que está disponível no AIX, no IBM i, no Linux, no e no Windows. Ele desempenha tarefas de administração em um gerenciador de filas usando pacotes de dados do IBM MQ manipular propriedades (ou parâmetros) de objetos de forma que é mais fácil do que usar Programmable Command Formats (PCFs).

O MQAI executa tarefas de administração em um gerenciador de filas por meio do uso de *pacotes de dados*. Pacotes de dados permitem manipular propriedades (ou parâmetros) de objetos de uma forma que é mais fácil do que usar PCFs.

As vantagens de uso da MQAI são as seguintes:

Simplificar o uso de mensagens do PCF

A MQAI é uma maneira mais fácil de administrar o IBM MQ. Se você usar a MQAI, não será necessário gravar suas próprias mensagens PCF. Isso evita os problemas associados às estruturas de dados complexos.

Para transmitir parâmetros em programas gravados usando chamadas MQI, a mensagem PCF deve conter o comando e os detalhes da sequência ou dos dados de número inteiro. Para criar esta configuração manualmente, é necessário incluir várias instruções em seu programa para cada estrutura e alocar espaço de memória. Esta tarefa pode ser longa e trabalhosa.

Os programas gravados usando a MQAI transmitem parâmetros no pacote de dados apropriado e é necessário somente uma instrução para cada estrutura. O uso dos pacotes de dados MQAI remove a necessidade de manipulação das matrizes e alocação de armazenamento e fornece um grau de isolamento dos detalhes do PCF.

Manipular condições de erro mais facilmente

É difícil obter códigos de retorno de volta a partir de comandos PCF. A MQAI facilita a manipulação das condições de erro pelo programa.

Trocar dados entre aplicativos

Os dados do aplicativo são enviados em formato PCF e compactados e descompactados pelo MQAI. Se os seus dados da mensagem consistem em seqüências de caracteres e números inteiros, é possível usar a MQAI para tirar vantagem da conversão de dados integrados do IBM MQ para dados PCF. Isso evita a necessidade de gravar saídas de conversão de dados.

Após criar e preencher o pacote de dados, será possível enviar uma mensagem de comando de administração para o servidor de comandos de um gerenciador de filas, usando a chamada mqExecute. Esta chamada aguarda quaisquer mensagens de resposta. A chamada mqExecute manipula a troca com o servidor de comandos e retorna respostas em um *pacote de respostas*.

Exemplos do Uso do MQAI

Os programas de amostra a seguir demonstram o uso do MQAI para executar as várias tarefas:

- [amqsaicq.c](#): criar uma fila local.
- [amqsaiem.c](#): exibir eventos na tela usando um monitor de eventos simples.
- [amqsailq.c](#): imprimir uma lista de todas as filas locais e suas profundidades atuais.
- [amqsaicl.c](#): imprimir uma lista de todos os canais e seus tipos.

Construindo seu Aplicativo do MQAI

Para construir seu aplicativo utilizando o MQAI, você faz link para as mesmas bibliotecas que você faz para IBM MQ. Para obter informações sobre como construir seus aplicativos IBM MQ, consulte [Construindo um aplicativo processual](#).

Dicas e sugestões para configurar IBM MQ usando MQAI

O MQAI usa mensagens do PCF para enviar comandos de administração para o servidor de comandos em vez de lidar diretamente com o servidor de comandos. Dicas para configurar o IBM MQ usando o MQAI podem ser localizadas em [“Sugestões e dicas para usar o MQAI para configurar o IBM MQ” na página 39](#).

Referências relacionadas

[Referência do IBM MQ Administration Interface](#)

Multi

Sugestões e dicas para usar o MQAI para configurar o IBM MQ

A interface de administração do IBM MQ (MQAI) usa mensagens do PCF para enviar comandos de administração para o servidor de comandos em vez de lidar diretamente com o servidor de comandos em si. Aqui estão algumas dicas para configurar o IBM MQ usando o MQAI.

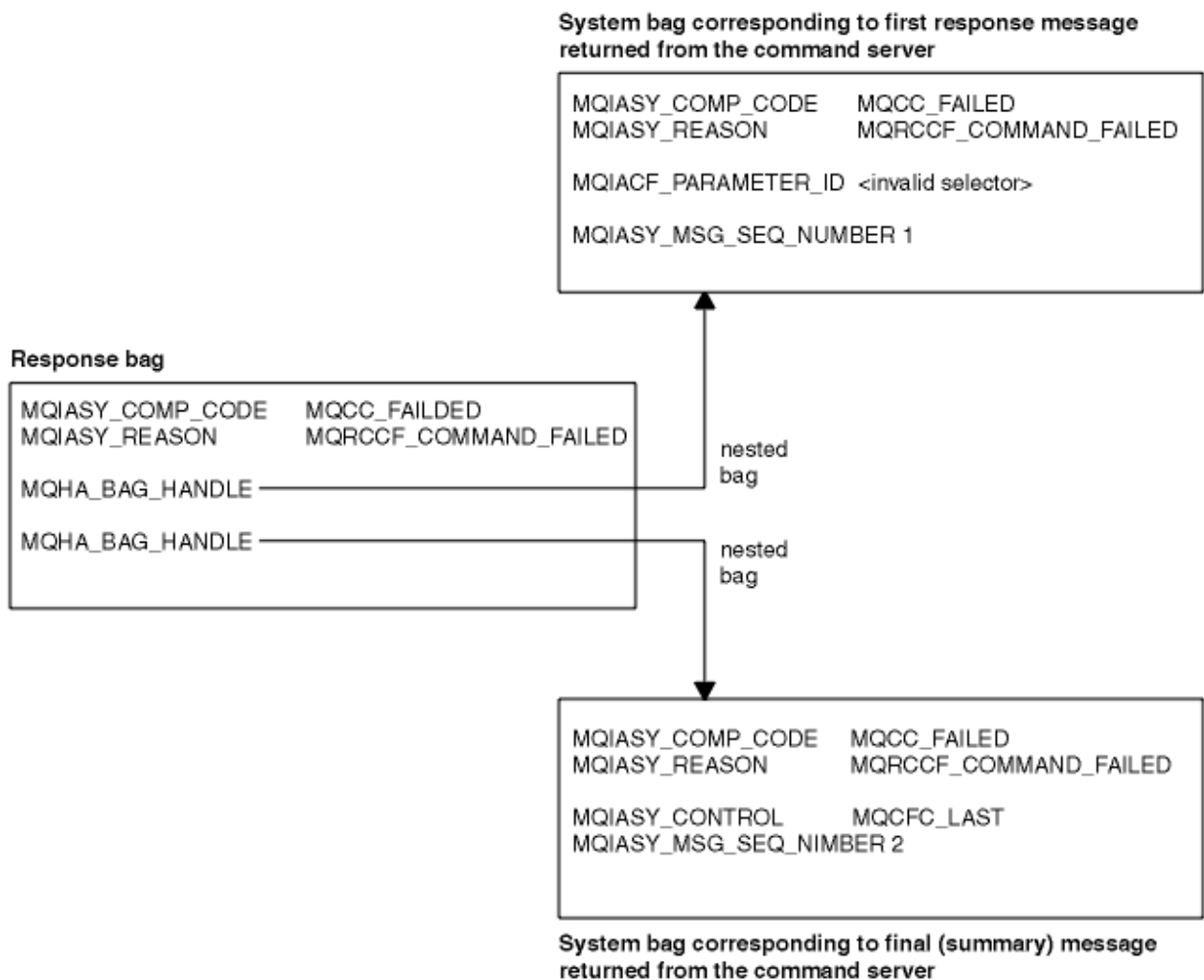
- As seqüências de caracteres no IBM MQ são preenchidas com branco para um comprimento fixo. Usando C, seqüências terminadas em nulo pode ser normalmente fornecidos como parâmetros de entrada para interfaces de programação do IBM MQ.
- Para limpar o valor de um atributo de seqüência, configure-o para um único caractere em branco, em vez de uma seqüência vazia.
- Considere previamente os atributos que você deseja mudar e consultar somente naqueles atributos.
- Certos atributos não podem ser mudados, por exemplo, um nome de fila ou um tipo de canal. Verifique se você tenta mudar somente aqueles atributos que possam ser modificados. Consulte a lista de

parâmetros obrigatórios e opcionais para o objeto de mudança PCF específico. Consulte [Definições dos formatos de comando programáveis](#).

- Se uma chamada MQAI falhar, alguns detalhes da falha são retornados ao pacote de respostas. Detalhes adicionais podem ser localizados em um pacote aninhado que pode ser acessado pelo seletor MQHA_BAG_HANDLE. Por exemplo, se uma chamada mqExecute falhar com um código de razão de MQRCCF_COMMAND_FAILED, essas informações serão retornadas no pacote de respostas. Uma razão possível para esse código de razão é que um seletor especificado não era válido para o tipo de mensagem de comando e esse detalhe de informações é encontrado em um pacote aninhado que pode ser acessado por um identificador de pacote.

Para obter mais informações sobre MQExecute, consulte [“Enviando comandos de administração para o servidor de comandos qm usando a chamada mqExecute”](#) na página 74

O diagrama a seguir mostra este cenário:



Multi Tópicos avançados do MQAI

Informações sobre indexação, conversão de dados e a utilização de descritor de mensagens

Indexando

Índices são usados ao substituir ou remover itens de dados existentes de um pacote para preservar a ordem de inserção.

Conversão de Dados

As sequências contidas em um pacote de dados do MQAI podem estar em uma variedade de conjuntos de caracteres codificados e estes podem ser convertidos usando a chamada [mqSetInteger](#).

Uso do descritor de mensagens

O MQAI gera um descritor de mensagens que é configurado para um valor inicial quando o pacote de dados é criado.

Multi **Indexação no MQAI**

Os índices são usados ao substituir ou remover itens de dados existentes de um pacote. Há três tipos de indexação, permitindo que os itens de dados sejam facilmente recuperados.

Cada seletor e valor dentro de um item de dados em um pacote têm três números de índice associados:

- O índice relativo a outros itens que têm o mesmo seletor.
- O índice relativo à categoria de seletor (usuário ou sistema) ao qual o item pertence.
- O índice relativo a todos os itens de dados no pacote (usuário e sistema).

Isso permite a indexação por seletores de usuários, seletores de sistema ou ambos, conforme mostrado na [Figura 3 na página 41](#).

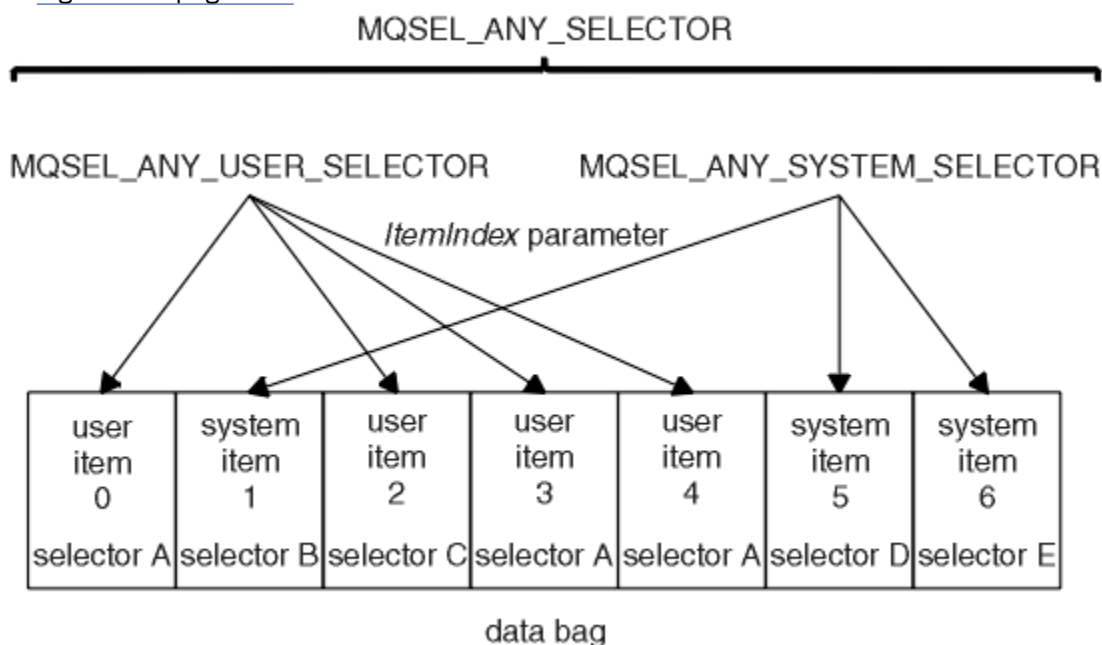


Figura 3. Indexando

No [Figura 3 na página 41](#), o item do usuário 3 (seletor A) pode ser referido pelos pares de índice a seguir:

- seletor A (ItemIndex 1)
- MQSEL_ANY_USER_SELECTOR (ItemIndex 2)
- MQSEL_ANY_SELECTOR (itemIndex 3)

O índice baseia-se em zero como uma matriz em C; se houver 'n' ocorrências, o índice variará de zero a 'n-1', sem diferenças.

Os índices são usados ao substituir ou remover itens de dados existentes de um pacote. Quando usados dessa forma, a ordem de inserção é preservada, mas os índices de outros itens de dados podem ser afetados. Para obter exemplos disso, consulte [“Mudando informações dentro de um pacote” na página 70](#) e [“Excluindo os itens de dados” na página 73](#).

Os três tipos de indexação permitem fácil recuperação de itens de dados. Por exemplo, se houver três instâncias de um determinado seletor em um pacote, a chamada `mqCountItems` poderá contar o número de instâncias desse seletor e as chamadas `mqInquire*` poderão especificar tanto o seletor quanto o índice para consultar apenas esses valores. Isso é útil para atributos que podem ter uma lista de valores, como algumas das saídas em canais.

As sequências contidas em um pacote de dados MQAI podem estar em uma variedade de conjuntos de caracteres codificados. Essas sequências podem ser convertidas usando a chamada `mqSetInteger`.

Como mensagens PCF, as sequências contidas em um pacote de dados MQAI podem estar em uma variedade de conjuntos de caracteres codificados. Geralmente, todas as sequências em uma mensagem PCF estão no mesmo conjunto de caracteres codificados, ou seja, o mesmo conjunto que o gerenciador de filas.

Cada item de sequência em um pacote de dados contém dois valores; a própria sequência e o CCSID. A sequência incluída no pacote é obtida do parâmetro **Buffer** da chamada `mqAddString` ou `mqSetString`. O CCSID é obtido do item de sistema que contém um seletor de `MQIASY_CODED_CHAR_SET_ID`. Isso é conhecido como o *pacote CCSID* e pode ser mudado usando a chamada `mqSetInteger`.

Quando você consulta o valor de uma sequência contida em um pacote de dados, o CCSID é um parâmetro de saída da chamada.

Tabela 3 na página 42 mostra as regras aplicadas ao converter pacotes de dados em mensagens e vice-versa:

<i>Tabela 3. Processamento do CCSID</i>			
Chamada MQAI	CCSID	Entrada a ser chamada	Saída a ser chamada
<code>mqBagToBuffer</code>	CCSID do pacote (1)	Ignorado	Sem mudança
<code>mqBagToBuffer</code>	CCSIDs de sequência no pacote	Utilizada	Sem mudança
<code>mqBagToBuffer</code>	CCSIDs de sequência no buffer	Não-aplicável	Copiada de CCSIDs de sequência no pacote
<code>mqBufferToBag</code>	CCSID do pacote (1)	Ignorado	Sem mudança
<code>mqBufferToBag</code>	CCSIDs de sequência no buffer	Utilizada	Sem mudança
<code>mqBufferToBagmqBufferToBag</code>	CCSIDs de sequência no pacote	Não-aplicável	Copiada de CCSIDs de sequência no buffer
<code>mqPutBag</code>	CCSID do MQMD	Utilizada	Sem mudança (2)
<code>mqPutBag</code>	CCSID do pacote (1)	Ignorado	Sem mudança
<code>mqPutBag</code>	CCSIDs de sequência no pacote	Utilizada	Sem mudança
<code>mqPutBag</code>	CCSIDs de sequência na mensagem enviada	Não-aplicável	Copiada de CCSIDs de sequência no pacote
<code>mqGetBolsa</code>	CCSID do MQMD	Usada para conversão de dados de mensagem	Configurada como CCSID de dados retornados (3)
<code>mqGetBolsa</code>	CCSID do pacote (1)	Ignorado	Sem mudança
<code>mqGetBolsa</code>	CCSIDs de sequência na mensagem	Utilizada	Sem mudança
<code>mqGetBolsa</code>	CCSIDs de sequência no pacote	Não-aplicável	Copiada de CCSIDs de sequência na mensagem
<code>mqExecute</code>	CCSID do pacote de solicitação	Usada para MQMD de mensagem de solicitação (4)	Sem mudança

Chamada MQAI	CCSID	Entrada a ser chamada	Saída a ser chamada
mqExecute	CCSID do pacote de resposta	Usada para conversão de dados de mensagem de resposta (4)	Configurada como CCSID de dados retornados (3)
mqExecute	CCSIDs de sequência no pacote de solicitação	Usada para mensagem de solicitação	Sem mudança
mqExecute	CCSIDs de sequência no pacote de resposta	Não-aplicável	Copiada de CCSIDs de sequência na mensagem de resposta

Notas:

1. CCSID de pacote é o item de sistema com o seletor MQIASY_CODED_CHAR_SET_ID.
2. MQCCSI_Q_MGR mudou para o CCSID do gerenciador de filas real.
3. Se a conversão de dados for solicitada, o CCSID de dados retornados será o mesmo que o valor de saída. Se a conversão de dados não for solicitada, o CCSID de dados retornados será o mesmo que o valor da mensagem. Observe que nenhuma mensagem será retornada se a conversão de dados for solicitada, mas falhará.
4. Se o CCSID for MQCCSI_DEFAULT, o CCSID do gerenciador de filas será usado.

Conceitos relacionados

“Conversão de dados entre conjuntos de caracteres codificados” na página 207

Os dados da mensagem em formatos definidos pelo IBM MQ (também conhecidos como formatos integrados) podem ser convertidos pelo gerenciador de filas de um conjunto de caracteres codificados para outro, desde que ambos os conjuntos de caracteres se relacionem a um único idioma ou grupo de idiomas semelhantes.

“O arquivo ccsid_part2.tbl” na página 209

O arquivo ccsid_part2.tbl é usado para fornecer informações adicionais de CCSID. O arquivo ccsid_part2.tbl substitui o arquivo ccsid.tbl que era usado antes da IBM MQ 9.0.

Multi **Uso do descritor de mensagens no MQAI**

O descritor de mensagens gerado pelo MQAI é configurado como um valor inicial quando o pacote de dados é criado.

O tipo de comando PCF é obtido do item do sistema com o seletor MQIASY_TYPE. Quando você cria seu pacote de dados, o valor inicial desse item é configurado dependendo do tipo de pacote criado:

Tipo de pacote	Valor inicial do item MQIASY_TYPE
MQCBO_ADMIN_BAG	MQCFT_COMMAND
MQCBO_COMMAND_BAG	MQCFT_COMMAND
MQCBO_*	MQCFT_USER

Quando o MQAI gera um descritor de mensagens, os valores usados nos parâmetros **Format** e **MsgType** dependem do valor do item de sistema com o seletor MQIASY_TYPE, conforme mostrado na Tabela 4 na página 43.

Tipo de comando PCF	Formato	MsgType
MQCFT_COMMAND	MQFMT_ADMIN	MQMT_REQUEST

Tipo de comando PCF	Formato	MsgType
MQCFT_REPORT	MQFMT_ADMIN	MQMT_REPORT
MQCFT_RESPONSE	MQFMT_ADMIN	MQMT_REPLY
MQCFT_TRACE_ROUTE	MQFMT_ADMIN	MQMT_DATAGRAM
MQCFT_EVENT	MQFMT_EVENT	MQMT_DATAGRAM
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

A Tabela 5 na página 43 mostra que se você criar um pacote de administração ou um pacote de comandos, o *Format* do descritor de mensagens será MQFMT_ADMIN e o *MsgType* será MQMT_REQUEST. Isso é adequado para uma mensagem de solicitação do PCF enviada para o servidor de comandos quando se espera uma resposta de volta.

Outros parâmetros no descritor de mensagens usam os valores mostrados na Tabela 6 na página 44.

Parâmetro	Value
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_1
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	Consulte a Tabela 5 na página 43
<i>Expiry</i>	30 segundos (nota “1” na página 44)
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	depende do CCSID do pacote (nota “2” na página 44)
<i>Format</i>	Consulte a Tabela 5 na página 43
<i>Priority</i>	MQPRI_PRIORITY_AS_Q_DEF
<i>Persistence</i>	MQPER_NOT_PERSISTENT
<i>MsgId</i>	MQMI_NONE
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	consulte a nota “3” na página 44
<i>ReplyToQMgr</i>	Em branco

Notas:

1. Esse valor pode ser substituído na chamada mqExecute usando o parâmetro **OptionsBag**. Para obter informações sobre isso, consulte [mqExecute](#).
2. Consulte “Processamento de conversão de dados no MQAI” na página 42.
3. Nome da fila de resposta especificada pelo usuário ou da fila dinâmica temporária gerada por MQAI para mensagens do tipo MQMT_REQUEST. Caso contrário, em branco.

O programa C de amostra amqsaicq.c cria uma fila local usando o MQAI.

```

/*****
/*
/* Program name: AMQSAICQ.C
/*
/* Description: Sample C program to create a local queue using the
/* IBM MQ Administration Interface (MQAI).
/*
/* Statement: Licensed Materials - Property of IBM
/*
/* 84H2000, 5765-B73
/* 84H2001, 5639-B42
/* 84H2002, 5765-B74
/* 84H2003, 5765-B75
/* 84H2004, 5639-B43
/*
/* (C) Copyright IBM Corp. 1999, 2024
/*
/*****
/*
/* Function:
/* AMQSAICQ is a sample C program that creates a local queue and is an
/* example of the use of the mqExecute call.
/*
/* - The name of the queue to be created is a parameter to the program.
/*
/* - A PCF command is built by placing items into an MQAI bag.
/* These are:-
/* - The name of the queue
/* - The type of queue required, which, in this case, is local.
/*
/* - The mqExecute call is executed with the command MQCMD_CREATE_Q.
/* The call generates the correct PCF structure.
/* The call receives the reply from the command server and formats into
/* the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server then the code returned by the
/* command server is retrieved from the system bag that is
/* embedded in the response bag to the mqExecute call.
/*
/* Note: The command server must be running.
/*
/*****
/*
/* AMQSAICQ has 2 parameters - the name of the local queue to be created
/* - the queue manager name (optional)
/*
/*****
/*****
/* Includes
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h> /* MQI */
#include <cmqcfh.h> /* PCF */
#include <cmqbc.h> /* MQAI */

void CheckCallResult(MQCHAR *, MQLONG , MQLONG );
void CreateLocalQueue(MQHCONN, MQCHAR *);

int main(int argc, char *argv[])
{
    MQHCONN hConn; /* handle to IBM MQ connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */
    MQLONG reason; /* reason code */

    /*****
    /* First check the required parameters
    */

```

```

/*****
printf("Sample Program to Create a Local Queue\n");
if (argc < 2)
{
    printf("Required parameter missing - local queue name\n");
    exit(99);
}

/*****
/* Connect to the queue manager */
/*****
if (argc > 2)
    strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(QMName, &hConn, &compCode, &connReason);

/*****
/* Report reason and stop if connection failed */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("MQCONN", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Call the routine to create a local queue, passing the handle to the */
/* queue manager and also passing the name of the queue to be created. */
/*****
CreateLocalQueue(hConn, argv[1]);

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
}
return 0;
}

/*****
/*
/* Function:      CreateLocalQueue
/* Description:   Create a local queue by sending a PCF command to the command
/*               server.
/*
/*
/*****
/*
/* Input Parameters:  Handle to the queue manager
/*                   Name of the queue to be created
/*
/*
/* Output Parameters: None
/*
/*
/* Logic: The mqExecute call is executed with the command MQCMD_CREATE_Q.
/*        The call generates the correct PCF structure.
/*        The default options to the call are used so that the command is sent*
/*        to the SYSTEM.ADMIN.COMMAND.QUEUE.
/*        The reply from the command server is placed on a temporary dynamic
/*        queue.
/*        The reply is read from the temporary queue and formatted into the
/*        response bag.
/*
/*
/*        The completion code from the mqExecute call is checked and if there
/*        is a failure from the command server then the code returned by the
/*        command server is retrieved from the system bag that is
/*        embedded in the response bag to the mqExecute call.
/*
/*****
void CreateLocalQueue(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG reason; /* reason code */
    MQLONG compCode; /* completion code */
    MQHBBAG commandBag = MQHB_UNUSABLE_HBBAG; /* command bag for mqExecute */
    MQHBBAG responseBag = MQHB_UNUSABLE_HBBAG; /* response bag for mqExecute */
    MQHBBAG resultBag; /* result bag from mqExecute */
    MQLONG mqExecuteCC; /* mqExecute completion code */
    MQLONG mqExecuteRC; /* mqExecute reason code */

    printf("\nCreating Local Queue %s\n", qName);

```

```

/*****
/* Create a command Bag for the mqExecute call. Exit the function if the */
/* create fails. */
/*****
mqCreateBag(MQCB0_ADMIN_BAG, &commandBag, &compCode, &reason);
CheckCallResult("Create the command bag", compCode, reason);
if (compCode !=MQCC_OK)
    return;

/*****
/* Create a response Bag for the mqExecute call, exit the function if the */
/* create fails. */
/*****
mqCreateBag(MQCB0_ADMIN_BAG, &responseBag, &compCode, &reason);
CheckCallResult("Create the response bag", compCode, reason);
if (compCode !=MQCC_OK)
    return;

/*****
/* Put the name of the queue to be created into the command bag. This will */
/* be used by the mqExecute call. */
/*****
mqAddString(commandBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, qName, &compCode,
            &reason);
CheckCallResult("Add q name to command bag", compCode, reason);

/*****
/* Put queue type of local into the command bag. This will be used by the */
/* mqExecute call. */
/*****
mqAddInteger(commandBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type to command bag", compCode, reason);

/*****
/* Send the command to create the required local queue. */
/* The mqExecute call will create the PCF structure required, send it to */
/* the command server and receive the reply from the command server into */
/* the response bag. */
/*****
mqExecute(hConn, /* IBM MQ connection handle */
          MQCMD_CREATE_Q, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          commandBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqExecute */
          &reason); /* Reason code from mqExecute call */

if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <stimqcsv QMgrName>\n");
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call and find the error if it failed. */
/*****
if ( compCode == MQCC_OK )
    printf("Local queue %s successfully created\n", qName);
else
{
    printf("Creation of local queue %s failed: Completion Code = %d\n",
          qName, compCode, reason);
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        /*****
        /* Get the system bag handle out of the mqExecute response bag. */
        /* This bag contains the reason from the command server why the */
        /* command failed. */
        /*****
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &resultBag, &compCode,
                    &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */

```

```

/*****
mqInquireInteger(resultBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                 &compCode, &reason);
CheckCallResult("Get the completion code from the result bag",
                compCode, reason);
mqInquireInteger(resultBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                 &compCode, &reason);
CheckCallResult("Get the reason code from the result bag", compCode,
                reason);
printf("Error returned by the command server: Completion code = %d :
        Reason = %d\n", mqExecuteCC, mqExecuteRC);
    }
}
/*****
/* Delete the command bag if successfully created. */
/*****
if (commandBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&commandBag, &compCode, &reason);
    CheckCallResult("Delete the command bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}
} /* end of CreateLocalQueue */

/*****
/*
/* Function: CheckCallResult
/*
/*****
/*
/* Input Parameters:  Description of call
/*                   Completion code
/*                   Reason code
/*
/* Output Parameters: None
/*
/* Logic: Display the description of the call, the completion code and the
/*        reason code if the completion code is not successful
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d :
                Reason = %d\n", callText, cc, rc);
}

```

Multi Programa C de amostra para exhibir eventos usando um monitor de eventos (amqsaiem.c)

O programa C de amostra amqsaiem.c demonstra um monitor de eventos básico usando o MQAI.

```

*****
/*
/* Program name: AMQSAIEM.C
/*
/* Description: Sample C program to demonstrate a basic event monitor
/*              using the IBM MQ Admin Interface (MQAI).
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1999, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/*****

```



```

/*                                                                    */
/* Function:                                                            */
/* AMQSAIEM is a sample C program that demonstrates how to write a simple */
/* event monitor using the mqGetBag call and other MQAI calls.          */
/*                                                                    */
/* The name of the event queue to be monitored is passed as a parameter */
/* to the program. This would usually be one of the system event queues:- */
/* SYSTEM.ADMIN.QMGR.EVENT      Queue Manager events                  */
/* SYSTEM.ADMIN.PERFM.EVENT     Performance events                    */
/* SYSTEM.ADMIN.CHANNEL.EVENT   Channel events                       */
/* SYSTEM.ADMIN.LOGGER.EVENT    Logger events                        */
/*                                                                    */
/* To monitor the queue manager event queue or the performance event queue, */
/* the attributes of the queue manager need to be changed to enable */
/* these events. For more information about this, see Part 1 of the */
/* Programmable System Management book. The queue manager attributes can */
/* be changed using either MQSC commands or the MQAI interface.      */
/* Channel events are enabled by default.                              */
/*                                                                    */
/* Program logic                                                        */
/* Connect to the Queue Manager.                                       */
/* Open the requested event queue with a wait interval of 30 seconds.  */
/* Wait for a message, and when it arrives get the message from the queue */
/* and format it into an MQAI bag using the mqGetBag call.            */
/* There are many types of event messages and it is beyond the scope of */
/* this sample to program for all event messages. Instead the program */
/* prints out the contents of the formatted bag.                      */
/* Loop around to wait for another message until either there is an error */
/* or the wait interval of 30 seconds is reached.                    */
/*                                                                    */
/*****                                                                    */
/* AMQSAIEM has 2 parameters - the name of the event queue to be monitored */
/* - the queue manager name (optional)                                  */
/*                                                                    */
/*****                                                                    */

/*****                                                                    */
/* Includes                                                            */
/*****                                                                    */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>                /* MQI                */
#include <cmqcfh.h>             /* PCF                */
#include <cmqbc.h>              /* MQAI                */

/*****                                                                    */
/* Macros                                                              */
/*****                                                                    */
#if MQAT_DEFAULT == MQAT_WINDOWS_NT
#define Int64 "I64"
#elif defined(MQ_64_BIT)
#define Int64 "l"
#else
#define Int64 "ll"
#endif

/*****                                                                    */
/* Function prototypes                                                */
/*****                                                                    */
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);
void GetQEvents(MQHCONN, MQCHAR *);
int PrintBag(MQHBAG);
int PrintBagContents(MQHBAG, int);

/*****                                                                    */
/* Function: main                                                      */
/*****                                                                    */
int main(int argc, char *argv[])
{
    MQHCONN hConn;                /* handle to connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QM name */
    MQLONG reason;                /* reason code */
    MQLONG connReason;            /* MQCONN reason code */
    MQLONG compCode;              /* completion code */

    /*****                                                                    */
    /* First check the required parameters                                */
    /*****                                                                    */

```

```

printf("Sample Event Monitor (times out after 30 secs)\n");
if (argc < 2)
{
    printf("Required parameter missing - event queue to be monitored\n");
    exit(99);
}

/*****
/* Connect to the queue manager */
*****/
if (argc > 2)
    strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(QMName, &hConn, &compCode, &connReason);
/*****
/* Report the reason and stop if the connection failed */
*****/
if (compCode == MQCC_FAILED)
{
    CheckCallResult("MQCONN", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Call the routine to open the event queue and format any event messages */
/* read from the queue. */
*****/
GetQEvents(hConn, argv[1]);

/*****
/* Disconnect from the queue manager if not already connected */
*****/
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
}

return 0;
}

/*****
/*
/* Function: CheckCallResult */
/*
*****/
/*
/* Input Parameters: Description of call */
/* Completion code */
/* Reason code */
/*
/* Output Parameters: None */
/*
/* Logic: Display the description of the call, the completion code and the */
/* reason code if the completion code is not successful */
*****/
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}

/*****
/*
/* Function: GetQEvents */
/*
*****/
/*
/* Input Parameters: Handle to the queue manager */
/* Name of the event queue to be monitored */
/*
/* Output Parameters: None */
/*
/* Logic: Open the event queue. */
/* Get a message off the event queue and format the message into */
/* a bag. */
/* A real event monitor would need to be programmed to deal with */
/* each type of event that it receives from the queue. This is */
/* outside the scope of this sample, so instead, the contents of */
*****/

```

```

/*          the bag are printed.                                */
/*          The program waits for 30 seconds for an event message and then */
/*          terminates if no more messages are available.        */
/*          */
/*****
void GetQEvents(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG openReason;                /* MQOPEN reason code      */
    MQLONG reason;                    /* reason code             */
    MQLONG compCode;                  /* completion code         */
    MQHOBJ eventQueue;                /* handle to event queue   */

    MQHBAG eventBag = MQHB_UNUSABLE_HBAG; /* event bag to receive event msg */
    MQOD  od = {MQOD_DEFAULT};          /* Object Descriptor      */
    MQMD  md = {MQMD_DEFAULT};          /* Message Descriptor     */
    MQGMO gmo = {MQGMO_DEFAULT};        /* get message options    */
    MQLONG bQueueOK = 1;                /* keep reading msgs while true */

    /*****
    /* Create an Event Bag in which to receive the event.          */
    /* Exit the function if the create fails.                      */
    /*****
    mqCreateBag(MQCBO_USER_BAG, &eventBag, &compCode, &reason);
    CheckCallResult("Create event bag", compCode, reason);
    if (compCode != MQCC_OK)
        return;

    /*****
    /* Open the event queue chosen by the user                      */
    /*****
    strncpy(od.ObjectName, qName, (size_t)MQ_Q_NAME_LENGTH);
    MQOPEN(hConn, &od, MQOO_INPUT_AS_Q_DEF+MQOO_FAIL_IF_QUIESCING, &eventQueue,
            &compCode, &openReason);
    CheckCallResult("Open event queue", compCode, openReason);

    /*****
    /* Set the GMO options to control the action of the get message from the */
    /* queue.                                                                */
    /*****
    gmo.WaitInterval = 30000;          /* 30 second wait for message */
    gmo.Options = MQGMO_WAIT + MQGMO_FAIL_IF_QUIESCING + MQGMO_CONVERT;
    gmo.Version = MQGMO_VERSION_2;    /* Avoid need to reset Message ID */
    gmo.MatchOptions = MQMO_NONE;     /* and Correlation ID after every */
                                     /* mqGetBag                       */

    /*****
    /* If open fails, we cannot access the queue and must stop the monitor. */
    /*****
    if (compCode != MQCC_OK)
        bQueueOK = 0;

    /*****
    /* Main loop to get an event message when it arrives                */
    /*****
    while (bQueueOK)
    {
        printf("\nWaiting for an event\n");

        /*****
        /* Get the message from the event queue and convert it into the event */
        /* bag.                                                                */
        /*****
        mqGetBag(hConn, eventQueue, &md, &gmo, eventBag, &compCode, &reason);

        /*****
        /* If get fails, we cannot access the queue and must stop the monitor. */
        /*****
        if (compCode != MQCC_OK)
        {
            bQueueOK = 0;

            /*****
            /* If get fails because no message available then we have timed out, */
            /* so report this, otherwise report an error.                          */
            /*****
            if (reason == MQRC_NO_MSG_AVAILABLE)
            {
                printf("No more messages\n");
            }
            else
            {
                CheckCallResult("Get bag", compCode, reason);
            }
        }
    }
}

```

```

    }

    /*****
    /* Event message read - Print the contents of the event bag */
    /*****
    else
    {
        if ( PrintBag(eventBag) )
            printf("\nError found while printing bag contents\n");

        } /* end of msg found */
    } /* end of main loop */
    /*****
    /* Close the event queue if successfully opened */
    /*****
    if (openReason == MQRC_NONE)
    {
        MQCLOSE(hConn, &eventQueue, MQCO_NONE, &compCode, &reason);
        CheckCallResult("Close event queue", compCode, reason);
    }

    /*****
    /* Delete the event bag if successfully created. */
    /*****
    if (eventBag != MQHB_UNUSABLE_HBAG)
    {
        mqDeleteBag(&eventBag, &compCode, &reason);
        CheckCallResult("Delete the event bag", compCode, reason);
    }

} /* end of GetQEvents */

/*****
/*
/* Function: PrintBag */
/*
/*
/*****
/*
/* Input Parameters: Bag Handle */
/*
/*
/* Output Parameters: None */
/*
/*
/* Returns: Number of errors found */
/*
/*
/* Logic: Calls PrintBagContents to display the contents of the bag. */
/*
/*****

int PrintBag(MQHBAG dataBag)
{
    int errors;

    printf("\n");
    errors = PrintBagContents(dataBag, 0);
    printf("\n");

    return errors;
}

/*****
/*
/* Function: PrintBagContents */
/*
/*
/*****
/*
/* Input Parameters: Bag Handle */
/*
/* Indentation level of bag */
/*
/*
/* Output Parameters: None */
/*
/*
/* Returns: Number of errors found */
/*
/*
/* Logic: Count the number of items in the bag */
/* Obtain selector and item type for each item in the bag. */
/* Obtain the value of the item depending on item type and display the */
/* index of the item, the selector and the value. */
/* If the item is an embedded bag handle then call this function again */
/* to print the contents of the embedded bag increasing the */
/* indentation level. */
/*
/*****
int PrintBagContents(MQHBAG dataBag, int indent)

```

```

{
/*****
/* Definitions
/*****
#define LENGTH 500                /* Max length of string to be read*/
#define INDENT 4                  /* Number of spaces to indent */
                                /* embedded bag display */

/*****
/* Variables
/*****
MQLONG itemCount;                /* Number of items in the bag */
MQLONG itemType;                /* Type of the item */
int i;                          /* Index of item in the bag */
MQCHAR stringVal[LENGTH+1];     /* Value if item is a string */
MQBYTE byteStringVal[LENGTH];  /* Value if item is a byte string */
MQLONG stringLength;           /* Length of string value */
MQLONG ccsid;                   /* CCSID of string value */
MQINT32 iValue;                 /* Value if item is an integer */
MQINT64 i64Value;              /* Value if item is a 64-bit
                                /* integer */
MQLONG selector;               /* Selector of item */
MQHBAG bagHandle;              /* Value if item is a bag handle */
MQLONG reason;                 /* reason code */
MQLONG compCode;               /* completion code */
MQLONG trimLength;             /* Length of string to be trimmed */
int errors = 0;                 /* Count of errors found */
char blanks[] = " ";           /* Blank string used to
                                /* indent display */

/*****
/* Count the number of items in the bag
/*****
mqCountItems(dataBag, MQSEL_ALL_SELECTORS, &itemCount, &compCode, &reason);

if (compCode != MQCC_OK)
    errors++;
else
{
    printf("
    printf("
    printf("
}

/*****
/* If no errors found, display each item in the bag
/*****
if (!errors)
{
    for (i = 0; i < itemCount; i++)
    {

/*****
/* First inquire the type of the item for each item in the bag
/*****
mqInquireItemInfo(dataBag, /* Bag handle
                    MQSEL_ANY_SELECTOR, /* Item can have any selector*/
                    i, /* Index position in the bag */
                    &selector, /* Actual value of selector */
                    /* returned by call */
                    &itemType, /* Actual type of item */
                    /* returned by call */
                    &compCode, /* Completion code */
                    &reason); /* Reason Code */

    if (compCode != MQCC_OK)
        errors++;

    switch(itemType)
    {
    case MQITEM_INTEGER:
/*****
/* Item is an integer. Find its value and display its index,
/* selector and value.
/*****
mqInquireInteger(dataBag, /* Bag handle
                  MQSEL_ANY_SELECTOR, /* Allow any selector */
                  i, /* Index position in the bag */
                  &iValue, /* Returned integer value */
                  &compCode, /* Completion code */
                  &reason); /* Reason Code */

```

```

        if (compCode != MQCC_OK)
            errors++;
        else
            printf("%.s %-2d %-4d (%d)\n",
                indent, blanks, i, selector, iValue);
        break

case MQITEM_INTEGER64:
    /******
    /* Item is a 64-bit integer. Find its value and display its
    /* index, selector and value.
    /******
    mqInquireInteger64(dataBag, /* Bag handle
                        MQSEL_ANY_SELECTOR, /* Allow any selector
                        i, /* Index position in the bag
                        i64Value, /* Returned integer value
                        &compCode, /* Completion code
                        &reason); /* Reason Code

    if (compCode != MQCC_OK)
        errors++;
    else
        printf("%.s %-2d %-4d (%"Int64"d)\n",
            indent, blanks, i, selector, i64Value);
    break;

case MQITEM_STRING:
    /******
    /* Item is a string. Obtain the string in a buffer, prepare
    /* the string for displaying and display the index, selector,
    /* string and Character Set ID.
    /******
    mqInquireString(dataBag, /* Bag handle
                    MQSEL_ANY_SELECTOR, /* Allow any selector
                    i, /* Index position in the bag
                    LENGTH, /* Maximum length of buffer
                    stringVal, /* Buffer to receive string
                    &stringLength, /* Actual length of string
                    &ccsid, /* Coded character set ID
                    &compCode, /* Completion code
                    &reason); /* Reason Code

    /******
    /* The call can return a warning if the string is too long for
    /* the output buffer and has been truncated, so only check
    /* explicitly for call failure.
    /******
    if (compCode == MQCC_FAILED)
        errors++;
    else
    {
        /******
        /* Remove trailing blanks from the string and terminate with
        /* a null. First check that the string should not have been
        /* longer than the maximum buffer size allowed.
        /******
        if (stringLength > LENGTH)
            trimLength = LENGTH;
        else
            trimLength = stringLength;
        mqTrim(trimLength, stringVal, &compCode, &reason);
        printf("%.s %-2d %-4d '%s' %d\n",
            indent, blanks, i, selector, stringVal, ccsid);
    }
    break;

case MQITEM_BYTE_STRING:
    /******
    /* Item is a byte string. Obtain the byte string in a buffer,
    /* prepare the byte string for displaying and display the
    /* index, selector and string.
    /******
    mqInquireByteString(dataBag, /* Bag handle
                        MQSEL_ANY_SELECTOR, /* Allow any selector
                        i, /* Index position in the bag
                        LENGTH, /* Maximum length of buffer
                        byteStringVal, /* Buffer to receive string
                        &stringLength, /* Actual length of string
                        &compCode, /* Completion code
                        &reason); /* Reason Code

```

```

/*****
/* The call can return a warning if the string is too long for */
/* the output buffer and has been truncated, so only check */
/* explicitly for call failure. */
/*****
if (compCode == MQCC_FAILED)
    errors++;
else
{
    printf("%.s %-2d %-4d X'",
           indent, blanks, i, selector);

    for (i = 0 ; i < stringLength ; i++)
        printf("

    printf("\n");
}
break;

case MQITEM_BAG:
/*****
/* Item is an embedded bag handle, so call the PrintBagContents*/
/* function again to display the contents. */
/*****
mqInquireBag(dataBag, /* Bag handle */
             MQSEL_ANY_SELECTOR, /* Allow any selector */
             i, /* Index position in the bag */
             &bagHandle, /* Returned embedded bag handle */
             &compCode, /* Completion code */
             &reason); /* Reason Code */

if (compCode != MQCC_OK)
    errors++;
else
{
    printf("%.s %-2d %-4d (%d)\n", indent, blanks, i,
           selector, bagHandle);
    if (selector == MQHA_BAG_HANDLE)
        printf("
    else
        printf("
        PrintBagContents(bagHandle, indent+INDENT);
}
break;

default:
    printf("
}
}
}
return errors;
}

```

Multi Programa C de amostra para consulta sobre objetos do canal (amqsaicl.c)

O exemplo de programa C amqsaicl.c consulta objetos de canal usando o MQAI.

```

/*****
/*
/* Program name: AMQSAICL.C */
/*
/* Description: Sample C program to inquire channel objects */
/*              using the IBM MQ Administration Interface (MQAI) */
/*
/* <N_OCO_COPYRIGHT> */
/* Licensed Materials - Property of IBM */
/*
/* 63H9336 */
/* (c) 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. */
/* <NOC_COPYRIGHT> */
/*****

```

```

/*                                                                    */
/* Function:                                                            */
/* AMQSAICL is a sample C program that demonstrates how to inquire    */
/* attributes of the local queue manager using the MQAI interface. In  */
/* particular, it inquires all channels and their types.              */
/*                                                                    */
/* - A PCF command is built from items placed into an MQAI administration */
/* bag.                                                                */
/* These are:-                                                         */
/*   - The generic channel name "*"                                   */
/*   - The attributes to be inquired. In this sample we just want    */
/*     name and type attributes                                     */
/*                                                                    */
/* - The mqExecute MQCMD_INQUIRE_CHANNEL call is executed.          */
/* The call generates the correct PCF structure.                      */
/* The default options to the call are used so that the command is sent */
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.                                */
/* The reply from the command server is placed on a temporary dynamic  */
/* queue.                                                            */
/* The reply from the MQCMD_INQUIRE_CHANNEL is read from the        */
/* temporary queue and formatted into the response bag.              */
/*                                                                    */
/* - The completion code from the mqExecute call is checked and if there */
/* is a failure from the command server, then the code returned by the */
/* command server is retrieved from the system bag that has been     */
/* embedded in the response bag to the mqExecute call.              */
/*                                                                    */
/* Note: The command server must be running.                          */
/*                                                                    */
/*****                                                                    */
/* AMQSAICL has 2 parameter - the queue manager name (optional)      */
/*   - output file (optional) default varies                          */
/*****                                                                    */

/*****                                                                    */
/* Includes                                                            */
/*****                                                                    */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#if (MQAT_DEFAULT == MQAT_OS400)
#include <recio.h>
#endif

#include <cmqc.h>                /* MQI                */
#include <cmqcfh.h>              /* PCF                */
#include <cmqbc.h>               /* MQAI               */
#include <cmqxc.h>              /* MQCD               */

/*****                                                                    */
/* Function prototypes                                                */
/*****                                                                    */
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****                                                                    */
/* DataTypes                                                         */
/*****                                                                    */
#if (MQAT_DEFAULT == MQAT_OS400)
typedef _RFILE OUTFILEHDL;
#else
typedef FILE OUTFILEHDL;
#endif

/*****                                                                    */
/* Constants                                                         */
/*****                                                                    */
#if (MQAT_DEFAULT == MQAT_OS400)
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "*SDR      ", /* MQCHT_SENDER      */
    "*SVR      ", /* MQCHT_SERVER      */
    "*RCVR     ", /* MQCHT_RECEIVER    */
    "*RQSTR    ", /* MQCHT_REQUESTER   */
    "*ALL      ", /* MQCHT_ALL          */
    "*CLTCN    ", /* MQCHT_CLNTCONN    */
    "*SVRCONN  ", /* MQCHT_SVRCONN     */
    "*CLUSRCVR", /* MQCHT_CLUSRCVR    */

```



```

    "CLUSSDR "      /* MQCHT_CLUSSDR */
};
#else
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "sdr      ", /* MQCHT_SENDER */
    "svr      ", /* MQCHT_SERVER */
    "rcvr     ", /* MQCHT_RECEIVER */
    "rqstr    ", /* MQCHT_REQUESTER */
    "all      ", /* MQCHT_ALL */
    "cltconn  ", /* MQCHT_CLNTCONN */
    "svrcn    ", /* MQCHT_SVRCONN */
    "clusrcvr ", /* MQCHT_CLUSRCVR */
    "cluscdr  ", /* MQCHT_CLUSSDR */
};
#endif

/*****/
/* Macros */
/*****/
#if (MQAT_DEFAULT == MQAT_OS400)
#define OUTFILE "QTEMP/AMQSAICL(AMQSAICL)"
#define OPENOUTFILE(hdl, fname) \
    (hdl = _Ropen((fname), "wr", rtncode=Y));
#define CLOSEOUTFILE(hdl) \
    _Rclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    _Rwrite((hdl), (buf), (buflen));
#elif (MQAT_DEFAULT == MQAT_UNIX)
#define OUTFILE "/tmp/amqsaicl.txt"
#define OPENOUTFILE(hdl, fname) \
    (hdl = fopen((fname), "w"));
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));
#else
#define OUTFILE "amqsaicl.txt"
#define OPENOUTFILE(fname) \
    fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));
#endif

#define ChlType2String(t) ChlTypeMap[(t)-1].name

/*****/
/* Function: main */
/*****/
int main(int argc, char *argv[])
{
    /*****/
    /* MQAI variables */
    /*****/
    MQHCONN hConn; /* handle to MQ connection */
    MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */
    MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG cAttrsBag; /* bag containing chl attributes */
    MQHBAG errorBag; /* bag containing cmd server error */
    MQLONG mqExecuteCC; /* mqExecute completion code */
    MQLONG mqExecuteRC; /* mqExecute reason code */
    MQLONG chlNameLength; /* Actual length of chl name */
    MQLONG chlType; /* Channel type */
    MQLONG i; /* loop counter */
    MQLONG numberOfBags; /* number of bags in response bag */
    MQCHAR chlName[MQ_OBJECT_NAME_LENGTH+1]; /* name of chl extracted from bag */
    MQCHAR OutputBuffer[100]; /* output data buffer */
    OUTFILEHDL *outfp = NULL; /* output file handle */

    /*****/

```

```

/* Connect to the queue manager */
/*****
if (argc > 1)
    strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(qmName, &hConn;, &compCode;, &connReason;);

/*****
/* Report the reason and stop if the connection failed. */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("Queue Manager connection", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Open the output file */
/*****
if (argc > 2)
{
    OPENOUTFILE(outfp, argv[2]);
}
else
{
    OPENOUTFILE(outfp, OUTFILE);
}

if(outfp == NULL)
{
    printf("Could not open output file.\n");
    goto MOD_EXIT;
}
/*****
/* Create an admin bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &adminBag;, &compCode;, &reason;);
CheckCallResult("Create admin bag", compCode, reason);

/*****
/* Create a response bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &responseBag;, &compCode;, &reason;);
CheckCallResult("Create response bag", compCode, reason);

/*****
/* Put the generic channel name into the admin bag */
/*****
mqAddString(adminBag, MQCACH_CHANNEL_NAME, MQBL_NULL_TERMINATED, "*",
            &compCode;, &reason;);
CheckCallResult("Add channel name", compCode, reason);

/*****
/* Put the channel type into the admin bag */
/*****
mqAddInteger(adminBag, MQIACH_CHANNEL_TYPE, MQCHT_ALL, &compCode;, &reason;);
CheckCallResult("Add channel type", compCode, reason);

/*****
/* Add an inquiry for various attributes */
/*****
mqAddInquiry(adminBag, MQIACH_CHANNEL_TYPE, &compCode;, &reason;);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the channel names and channel types. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* MQ connection handle */
          MQCMD_INQUIRE_CHANNEL, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode;, /* Completion code from the mqexecute */
          &reason;); /* Reason code from mqexecute call */

/*****
/* Check the command server is started. If not exit. */
/*****

```

```

/*****
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName=">\n");
    goto MOD_EXIT;
}

/*****
/* Check the result from mqExecute call. If successful find the channel */
/* types for all the channels. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each channel are in separate bags. */
    /*****
mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags,
              &compCode, &reason);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfBags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the channel attributes */
        /*****
mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &cAttrsBag,
              &compCode, &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the channel name out of the channel attributes bag */
        /*****
mqInquireString(cAttrsBag, MQCACH_CHANNEL_NAME, 0, MQ_OBJECT_NAME_LENGTH,
                 chlName, &chlNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get channel name", compCode, reason);

        /*****
        /* Get the channel type out of the channel attributes bag */
        /*****
mqInquireInteger(cAttrsBag, MQIACH_CHANNEL_TYPE, MQIND_NONE, &chlType,
                 &compCode, &reason);
        CheckCallResult("Get type", compCode, reason);

        /*****
        /* Use mqTrim to prepare the channel name for printing. */
        /* Print the result. */
        /*****
mqTrim(MQ_CHANNEL_NAME_LENGTH, chlName, chlName, &compCode, &reason);
        sprintf(OutputBuffer, "%-20s%-9s", chlName, ChlType2String(chlType));
        WRITEOUTFILE(outfp, OutputBuffer, 29)
    }
}

else /* Failed mqExecute */
{
    printf("Call to get channel attributes failed: Cc = %ld : Rc = %ld\n",
           compCode, reason);
    /*****
    /* If the command fails get the system bag handle out of the mqexecute */
    /* response bag. This bag contains the reason from the command server */
    /* why the command failed. */
    /*****
if (reason == MQRCCF_COMMAND_FAILED)
{
    mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag,
                 &compCode, &reason);
    CheckCallResult("Get the result bag handle", compCode, reason);

    /*****
    /* Get the completion code and reason code, returned by the command */
    /* server, from the embedded error bag. */
    /*****
mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                 &compCode, &reason);
    CheckCallResult("Get the completion code from the result bag",
                    compCode, reason);
    mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                     &compCode, &reason);
    CheckCallResult("Get the reason code from the result bag",

```

```

        compCode, reason);
    printf("Error returned by the command server: Cc = %ld : Rc = %ld\n",
        mqExecuteCC, mqExecuteRC);
    }
}

MOD_EXIT:
/*****
/* Delete the admin bag if successfully created.
*****/
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created.
*****/
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/*****
/* Disconnect from the queue manager if not already connected
*****/
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
}

/*****
/* Close the output file if open
*****/
if(outfp != NULL)
    CLOSEOUTFILE(outfp);

return 0;
}

/*****
/*
/* Function: CheckCallResult
/*
*****/
/*
/* Input Parameters:  Description of call
/*                    Completion code
/*                    Reason code
/*
/* Output Parameters: None
/*
/* Logic: Display the description of the call, the completion code and the
/*        reason code if the completion code is not successful
/*
*****/
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %ld : Reason = %ld\n", callText,
            cc, rc);
}

```

Programa C de amostra para consultar filas e informações de impressão (amqsailq.c)

O programa de amostra C amqsailq.c consulta a profundidade atual das filas locais usando o MQAI.

```

/*****
/*
/* Program name: AMQSAILQ.C
/*
/* Description: Sample C program to inquire the current depth of the local
*****/

```

```

/*          queues using the IBM MQ Administration Interface (MQAI)          */
/*          */                                                                */
/* Statement: Licensed Materials - Property of IBM                          */
/*          */                                                                */
/*          84H2000, 5765-B73                                                */
/*          84H2001, 5639-B42                                                */
/*          84H2002, 5765-B74                                                */
/*          84H2003, 5765-B75                                                */
/*          84H2004, 5639-B43                                                */
/*          */                                                                */
/*          (C) Copyright IBM Corp. 1999, 2024                              */
/*          */                                                                */
/*****
/*
/* Function:
/* AMQSAILQ is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires the current depths of all the local queues.
/*
/* - A PCF command is built by placing items into an MQAI administration
/* bag.
/* These are:-
/* - The generic queue name "*"
/* - The type of queue required. In this sample we want to
/* inquire local queues.
/* - The attribute to be inquired. In this sample we want the
/* current depths.
/*
/* - The mqExecute call is executed with the command MQCMD_INQUIRE_Q.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply from the MQCMD_INQUIRE_Q command is read from the
/* temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server, then the code returned by
/* command server is retrieved from the system bag that has been
/* embedded in the response bag to the mqExecute call.
/*
/* - If the call is successful, the depth of each local queue is placed
/* in system bags embedded in the response bag of the mqExecute call.
/* The name and depth of each queue is obtained from each of the bags
/* and the result displayed on the screen.
/*
/* Note: The command server must be running.
/*
/*****
/*
/* AMQSAILQ has 1 parameter - the queue manager name (optional)
/*
/*****
/*****
/* Includes
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>          /* MQI          */
#include <cmqcfh.h>       /* PCF          */
#include <cmqbc.h>        /* MQAI         */

/*****
/* Function prototypes
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* Function: main
/*****
int main(int argc, char *argv[])
{
    /*****
    /* MQAI variables
    /*****
    MQHCONN hConn;          /* handle to IBM MQ connection
    MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name

```

```

MQLONG reason; /* reason code */
MQLONG connReason; /* MQCONN reason code */
MQLONG compCode; /* completion code */
MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
MQHBAG qAttrBag; /* bag containing q attributes */
MQHBAG errorBag; /* bag containing cmd server error */
MQLONG mqExecuteCC; /* mqExecute completion code */
MQLONG mqExecuteRC; /* mqExecute reason code */
MQLONG qNameLength; /* Actual length of q name */
MQLONG qDepth; /* depth of queue */
MQLONG i; /* loop counter */
MQLONG numberOfBags; /* number of bags in response bag */
MQCHAR qName[MQ_Q_NAME_LENGTH+1]; /* name of queue extracted from bag*/

printf("Display current depths of local queues\n\n");

/*****
/* Connect to the queue manager */
/*****
if (argc > 1)
    strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(qmName, &hConn, &compCode, &connReason);

/*****
/* Report the reason and stop if the connection failed. */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("Queue Manager connection", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Create an admin bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
CheckCallResult("Create admin bag", compCode, reason);
/*****
/* Create a response bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
CheckCallResult("Create response bag", compCode, reason);

/*****
/* Put the generic queue name into the admin bag */
/*****
mqAddString(adminBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, "*",
            &compCode, &reason);
CheckCallResult("Add q name", compCode, reason);

/*****
/* Put the local queue type into the admin bag */
/*****
mqAddInteger(adminBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type", compCode, reason);

/*****
/* Add an inquiry for current queue depths */
/*****
mqAddInquiry(adminBag, MQIA_CURRENT_Q_DEPTH, &compCode, &reason);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the local queue names and queue depths. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* IBM MQ connection handle */
          MQCMD_INQUIRE_Q, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response*/
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE*/
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqExecute */
          &reason); /* Reason code from mqExecute call */

```

```

/*****
/* Check the command server is started. If not exit. */
/*****
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n");
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call. If successful find the current */
/* depths of all the local queues. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each queue are in a separate bag. */
    /*****
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags, &compCode,
        &reason);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfBags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the queue attributes */
        /*****
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &qAttrsBag, &compCode,
            &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the queue name out of the queue attributes bag */
        /*****
        mqInquireString(qAttrsBag, MQCA_Q_NAME, 0, MQ_Q_NAME_LENGTH, qName,
            &qNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get queue name", compCode, reason);

        /*****
        /* Get the depth out of the queue attributes bag */
        /*****
        mqInquireInteger(qAttrsBag, MQIA_CURRENT_Q_DEPTH, MQIND_NONE, &qDepth,
            &compCode, &reason);
        CheckCallResult("Get depth", compCode, reason);

        /*****
        /* Use mqTrim to prepare the queue name for printing. */
        /* Print the result. */
        /*****
        mqTrim(MQ_Q_NAME_LENGTH, qName, qName, &compCode, &reason);
        printf("%4d %-48s\n", qDepth, qName);
    }
}

else /* Failed mqExecute */
{
    printf("Call to get queue attributes failed: Completion Code = %d :
        Reason = %d\n", compCode, reason);

    /*****
    /* If the command fails get the system bag handle out of the mqExecute */
    /* response bag. This bag contains the reason from the command server */
    /* why the command failed. */
    /*****
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag, &compCode,
            &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        /*****
        mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
            &compCode, &reason );
        CheckCallResult("Get the completion code from the result bag",
            compCode, reason);
    }
}

```

```

mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                 &compCode, &reason);
CheckCallResult("Get the reason code from the result bag",
                compCode, reason);
printf("Error returned by the command server: Completion Code = %d :
       Reason = %d\n", mqExecuteCC, mqExecuteRC);
}
}

/*****
/* Delete the admin bag if successfully created.          */
/*****
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created.      */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from queue manager", compCode, reason);
}
return 0;
}

*****/
*
* Function: CheckCallResult                                */
*
* *****/
* Input Parameters:  Description of call                    */
*                   Completion code                       */
*                   Reason code                           */
*
* Output Parameters: None                                  */
*
* Logic: Display the description of the call, the completion code and the
*        reason code if the completion code is not successful
*
* *****/
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
              callText, cc, rc);
}

```

Multi Pacotes de dados e o MQAI

Um pacote de dados é um meio de manipulação de propriedades ou parâmetros de objetos utilizando a interface de administração do IBM MQ (MQAI).

Pacotes de dados

- O pacote de dados contém zero ou mais *itens de dados*. Esses itens de dados são ordenados dentro do pacote conforme eles são colocados no pacote. Isto é chamado de *ordem de inserção*. Cada item de dados contém um *seletor* que identifica o item de dados e um *valor* desse item de dados que pode ser um número inteiro, um número inteiro de 64 bits, um filtro inteiro, uma sequência, um filtro de sequência, uma sequência de bytes, um filtro de sequência de bytes ou um identificador de outro

pacote. Itens de dados são descritos em detalhes em [“Tipos de item de dados disponível no MQAI”](#) na página 67

Há dois tipos de seletor: *seletores de usuário* e *seletores de sistema*. Esses são descritos em [Seletores MQAI](#). Os seletores são geralmente exclusivos, mas é possível ter diversos valores para o mesmo seletor. Neste caso, um *índice* identifica a ocorrência específica de seletor que é necessário. Os índices são descritos em [“Indexação no MQAI”](#) na página 41.

Uma hierarquia desses conceitos é mostrada em [Figura 1](#).

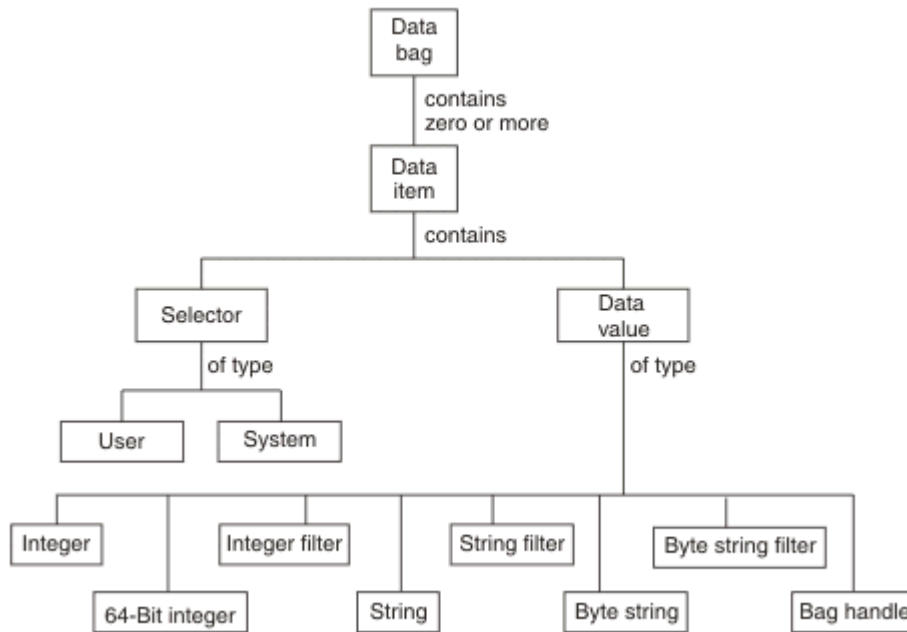


Figura 4. Hierarquia de conceitos MQAI

A hierarquia foi explicada em um parágrafo anterior.

Tipos de pacote de dados

É possível escolher o tipo de pacote de dados que você deseja criar dependendo da tarefa que deseja executar:

pacote do usuário???

Um pacote simples utilizado para dados do usuário.

pacote de administração

Um pacote criado para dados usados para administrar objetos do IBM MQ enviando mensagens administrativas para um servidor de comandos. O pacote de administração automaticamente implica determinadas opções conforme descrito em [“Criação e exclusão de pacotes de dados”](#) na página 66.

pacote de comandos???

Um pacote criado também para comandos para administrar objetos do IBM MQ. No entanto, ao contrário do pacote de administração, o pacote de comandos não implica automaticamente determinadas opções, embora estas opções estejam disponíveis. Para obter mais informações sobre as opções, consulte [“Criação e exclusão de pacotes de dados”](#) na página 66.

pacote de grupos???

Um pacote utilizado para conter um conjunto de itens de dados agrupados. Pacotes de grupos não podem ser usados para administrar objetos do IBM MQ.

Além disso, o **pacote do sistema** é criado pelo MQAI quando uma mensagem de resposta é retornada do servidor de comandos e colocada no pacote de saída de um usuário. Um pacote do sistema não pode ser modificado pelo usuário.

Usando pacote de dados As diferentes maneiras de utilizar pacotes de dados são listados neste tópico:

Usando pacotes de dados

As diferentes maneiras de utilizar pacotes de dados são mostradas na lista a seguir:

- É possível criar e excluir dados de [“Criação e exclusão de pacotes de dados”](#) na página 66.
- É possível enviar dados entre aplicativos que utilizam dados de [“Colocando e recebendo pacotes de dados usando o MQAI”](#) na página 67.
- É possível incluir itens de dados para dados de [“Incluindo itens de dados em pacotes com o MQAI”](#) na página 68.
- É possível incluir um comando de consulta dentro de um pacote de dados do [“Incluindo um comando de consulta a um pacote”](#) na página 69.
- É possível consultar dentro de pacotes de dados do [“Consultando dentro de pacotes de dados”](#) na página 69.
- É possível contar itens de dados dentro de um pacote de dados [“Contando itens de dados”](#) na página 72.
- É possível mudar as informações dentro de um pacote de dados do [“Mudando informações dentro de um pacote”](#) na página 70.
- É possível limpar um pacote de dados do [“Limpendo um pacote através da chamada mqClearBag”](#) na página 71.
- É possível truncar um pacote de dados do [“Truncando um pacote usando a chamada mqTruncateBag”](#) na página 71.
- É possível converter pacotes e buffers de [“Convertendo pacotes e buffers”](#) na página 72.

Multi

Criação e exclusão de pacotes de dados

Criando pacotes de dados

Para usar a MQAI, primeiro crie um pacote de dados usando a chamada mqCreateBag. Como entrada para essa chamada, forneça uma ou mais opções para controlar a criação do pacote.

O parâmetro **Options** da chamada MQCreateBag permite que você escolha se deseja criar um pacote do usuário, um pacote de comandos, um pacote de grupos ou um pacote de administração.

Para criar um pacote do usuário, um pacote de comandos ou um pacote de grupos, é possível escolher um ou mais opções adicionais para:

- Utilize o formulário da lista quando há duas ou mais ocorrências adjacentes da mesma seletor em um pacote.
- Reordene os itens de dados à medida que eles são incluídos em uma mensagem de PCF para assegurar que os parâmetros estão em sua ordem correta. Para obter informações adicionais sobre itens de dados, consulte [“Tipos de item de dados disponível no MQAI”](#) na página 67.
- Verifique os valores de seletores de usuário para os itens que você incluir no pacote.

Pacotes de administração automaticamente significam essas opções.

Um pacote de dados é identificado por seu identificador. O identificador de pacote é retornado a partir de mqCreateBag e deve ser fornecido em todas as outras chamadas que utilizam o pacote de dados.

Para uma descrição completa da chamada mqCreateBag, consulte [mqCreateBag](#).

Exclusão de pacotes de dados

Qualquer pacote de dados que é criado pelo usuário também deve ser excluído usando a chamada `mqDeleteBag`. Por exemplo, se um pacote é criado no código do usuário, ele também deve ser excluído no código do usuário.

Pacotes do sistema são criados e excluídos automaticamente pela MQAI. Para obter informações adicionais sobre isto, consulte [“Enviando comandos de administração para o servidor de comandos qm usando a chamada `mqExecute`” na página 74](#). O código do usuário não pode excluir um pacote do sistema.

Para obter uma descrição completa da chamada `mqDeleteBag`, consulte [mqDeleteBag](#).

Multi

Colocando e recebendo pacotes de dados usando o MQAI

Os dados também podem ser enviados entre os aplicativos colocando e obtendo pacotes de dados utilizando as chamadas `mqPutBag` e `mqGetBag`. Isso permite que a interface de administração do IBM MQ (MQAI) manipule o buffer em vez do aplicativo.

A chamada `mqPutBag` converte o conteúdo do pacote especificado em uma mensagem PCF e envia a mensagem para a fila especificada e a chamada `mqGetBag` remove a mensagem da fila especificada e converte-a de volta em um pacote de dados. Portanto, a chamada `mqPutBag` é o equivalente da chamada `mqBagToBuffer` seguido por `MQPUT` e o `mqGetBag` é o equivalente da chamada `MQGET` seguida por `mqBufferToBag`.

Para obter mais informações sobre o envio e recebimento de mensagens PCF em uma fila específica, consulte [“Enviando e recebendo mensagens PCF em uma fila especificada” na página 29](#)

Nota: Se você optar por utilizar a chamada `mqGetBag`, os detalhes de PCF dentro da mensagem devem ser corretos; se não forem, ocorrerá um erro apropriado e a mensagem PCF não será retornada.

Multi

Tipos de item de dados disponível no MQAI

Os itens de dados são usados pela interface de administração do IBM MQ (MQAI) para preencher pacotes de dados quando eles são criados. Esses itens de dados podem ser itens de usuário ou de sistema.

Esses itens do usuário contêm dados do usuário como atributos de objetos que estão sendo administrados. Itens do sistema devem ser utilizados para maior controle sobre as mensagens geradas: por exemplo, a geração de cabeçalhos da mensagem. Para obter mais informações sobre os itens do sistema, consulte [“Itens de sistema e o MQAI” na página 68](#).

Tipos de itens de dados

Quando você tiver criado um pacote de dados, é possível preenchê-lo com itens de sequência de caracteres ou de número inteiro. É possível consultar sobre todos os três tipos de item.

O item de dados pode ser um número inteiro ou itens de sequência. Aqui estão os tipos de item de dados disponível dentro do MQAI:

- Integer
- Número inteiro de 64 bits
- Filtro de número inteiro
- sequência de caracteres
- Filtragem de sequência
- Sequência de bytes
- Filtragem de sequência de bytes
- identificador de pacote

Utilizando itens de dados

Estas são as seguintes maneiras de utilizar itens de dados:

- [“Contando itens de dados” na página 72](#).

- “[Excluindo os itens de dados](#)” na página 73.
- “[Incluindo itens de dados em pacotes com o MQAI](#)” na página 68.
- “[Filtrando e consultando itens de dados](#)” na página 69.

Multi *Itens de sistema e o MQAI*

Itens de sistema podem ser usados pela interface de administração do IBM MQ (MQAI) para:

- A geração de cabeçalhos PCF. Itens de sistema podem controlar o identificador de comandos PCF, as opções de controle, o número de sequência da mensagem e o tipo de comando.
- Conversão de dados. Itens de sistema manipulam o identificador do conjunto de caracteres para os itens de sequência de caracteres no pacote.

Como todos os itens de dados, os itens do sistema consistem em um seletor e um valor. Para obter informações sobre esses seletores e o que eles são para, consulte [MQAI Seletores](#).

Itens de sistema são exclusivos. Um ou mais itens do sistema pode ser identificado por um seletor de sistema. Existe somente uma ocorrência de cada seletor de sistema.

A maioria dos itens do sistema pode ser modificada (consulte “[Mudando informações dentro de um pacote](#)” na página 70), mas as opções de criação de pacote não podem ser mudadas pelo usuário. Não é possível excluir itens do sistema. (Consulte “[Excluindo os itens de dados](#)” na página 73.)

Multi *Incluindo itens de dados em pacotes com o MQAI*

Quando um pacote de dados é criado usando a interface de administração do IBM MQ (MQAI), é possível preenchê-lo com itens de dados. Esses itens de dados podem ser itens de usuário ou de sistema.

Para obter informações sobre itens de dados, consulte “[Tipos de item de dados disponível no MQAI](#)” na página 67.

A MQAI permite que você inclua itens de número inteiro, itens de número inteiro de 64 bits, itens de filtro de número inteiro, itens de sequência de caracteres, filtragem de sequência, itens da sequência de bytes e itens de filtro da sequência de byte para pacotes e isto é mostrado em [Figura 5](#) na página 68. Os itens são identificados por um seletor. Geralmente um seletor identifica um único item, mas nem sempre é o caso. Se um item de dados com o seletor especificado já está presente no pacote, uma instância adicional desse seletor é incluída no final do pacote.

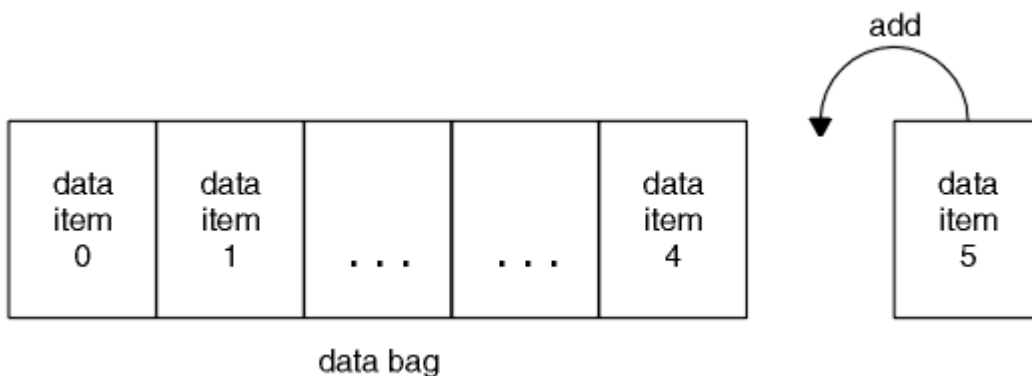


Figura 5. Incluindo itens de dados

Incluir itens de dados em um pacote usando as chamadas mqAdd*:

- Para incluir itens de número inteiro, use a chamada [mqAddInteger](#) conforme descrito em [mqAddInteger](#)
- Para incluir itens de número inteiro de 64 bits, use a chamada [mqAddInteger64](#) conforme descrito em [mqAddInteger64](#)
- Para incluir itens de filtro de número inteiro, use a chamada [mqAddIntegerFilter](#) conforme descrito em [mqAddIntegerFilter](#)

- Para incluir itens de sequência de caracteres, use a chamada `mqAddString` conforme descrito em [mqAddString](#)
- Para incluir itens de filtragem de sequência, use a chamada `mqAddStringFilter` conforme descrito em [mqAddStringFilter](#)
- Para incluir itens de sequência de bytes, use a chamada `mqAddByteString` conforme descrito em [mqAddByteString](#)
- Para incluir itens de filtragem de sequência de bytes, use a chamada `mqAddByteStringFilter` conforme descrito em [mqAddByteStringFilter](#)

Para obter mais informações sobre a inclusão de itens de dados em um pacote, consulte [“Itens de sistema e o MQAI”](#) na página 68.

Multi Incluindo um comando de consulta a um pacote

A chamada `mqAddInquiry` é utilizado para incluir um comando de consulta a um pacote. A chamada é especificamente para fins de administração, para que ele possa ser utilizado somente com pacotes de administração. Ele permite especificar os seletores de atributos nos quais você deseja consultar a partir do IBM MQ.

Para obter uma descrição integral da chamada `mqAddInquiry`, consulte [mqAddInquiry](#).

Multi Filtrando e consultando itens de dados

Ao utilizar o MQAI para pesquisar sobre os atributos de objetos do IBM MQ, é possível controlar os dados que são retornados para seu programa de duas maneiras.

- É possível **filtrar** os dados que são retornados usando as chamadas `mqAddInteger` e `mqAddString`. Esta abordagem permite especificar um par de *Selector* e *ItemValue*, por exemplo:

```
mqAddInteger(inputbag, MQIA_Q_TYPE, MQQT_LOCAL)
```

Este exemplo especifica que o tipo de fila (*Selector*) deve ser local (*ItemValue*) e esta especificação deve corresponder aos atributos do objeto (nesse caso, uma fila) sobre o qual você está consultando.

Outros atributos que podem ser filtrados correspondem aos PCF Inquire* comandos que podem ser localizados em [“Introdução aos formatos de comando programável do IBM MQ”](#) na página 26. Por exemplo, para consultar sobre os atributos de um canal, consulte o comando `Inquire Channel` nesta documentação do produto. Os "parâmetros obrigatórios" e "parâmetros opcionais" do comando `Inquire Channel` identificam os seletores que podem ser utilizados para filtragem.

- É possível **consultar** atributos específicos de um objeto utilizando a chamada `mqAddInquiry`. Isso especifica o seletor no qual você está interessado. Se você não especificar o seletor, todos os atributos do objeto são retornados.

Aqui está um exemplo de filtragem e consulta os atributos de uma fila:

```
/* Request information about all queues */
mqAddString(adminbag, MQCA_Q_NAME, "*")

/* Filter attributes so that local queues only are returned */
mqAddInteger(adminbag, MQIA_Q_TYPE, MQQT_LOCAL)

/* Query the names and current depths of the local queues */
mqAddInquiry(adminbag, MQCA_Q_NAME)
mqAddInquiry(adminbag, MQIA_CURRENT_Q_DEPTH)

/* Send inquiry to the command server and wait for reply */
mqExecute(MQCMD_INQUIRE_Q, ...)
```

Multi Consultando dentro de pacotes de dados

É possível consultar sobre:

- O valor de um item de número inteiro utilizando a chamada `mqInquireInteger`. Consulte [mqInquireInteger](#).
- O valor de um item de inteiro 64-bit usando a chamada `mqInquireInteger64`. Consulte [mqInquireInteger64](#).
- O valor de um item do filtro de número inteiro usando a chamada `mqInquireIntegerFilter`. Consulte [mqInquireIntegerFilter](#).
- O valor de um item de sequência usando a chamada `mqInquireString`. Consulte [mqInquireString](#).
- O valor de um item de filtragem de sequência usando a chamada `mqInquireStringFilter`. Consulte [mqInquireStringFilter](#).
- O valor de um item de sequência de bytes utilizando a chamada `mqInquireByteString`. Consulte [mqInquireByteString](#).
- O valor de um item de filtragem de sequência de byte usando a chamada `mqInquireByteStringFilter`. Consulte [mqInquireByteStringFilter](#).
- O valor de um identificador de pacote através da chamada `mqInquireBag`. Consulte [mqInquireBag](#).

Também é possível consultar sobre o tipo (número inteiro, inteiro de 64 bits, filtro de número inteiro, sequência de caracteres, filtragem de sequência, sequência de bytes, filtro de filtragem de sequência ou identificador de pacote) de um item específico usando a chamada `mqInquireItemInfo`. Consulte [mqInquireItemInfo](#).

Multi Mudando informações dentro de um pacote

O MQAI permite que você mude informações em um pacote usando as chamadas `mqSet*`. Você pode:

1. Modificar itens de dados dentro de um pacote. O índice permite que uma instância individual de um parâmetro seja substituída identificando a ocorrência do item a ser modificado (consulte [Figura 6](#) na página 70).

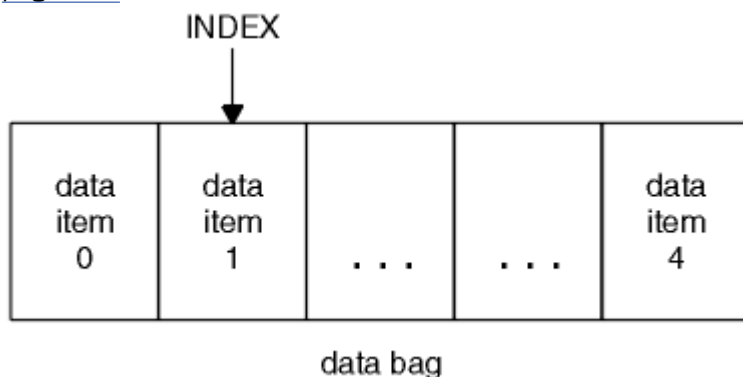


Figura 6. Modificando um item de dados único

2. Excluir todas as ocorrências existentes do seletor especificado e incluir uma nova ocorrência no final do pacote. (Consulte [Figura 7](#) na página 71.) Um valor de índice especial permite que **todas** as instâncias de um parâmetro sejam substituídas.

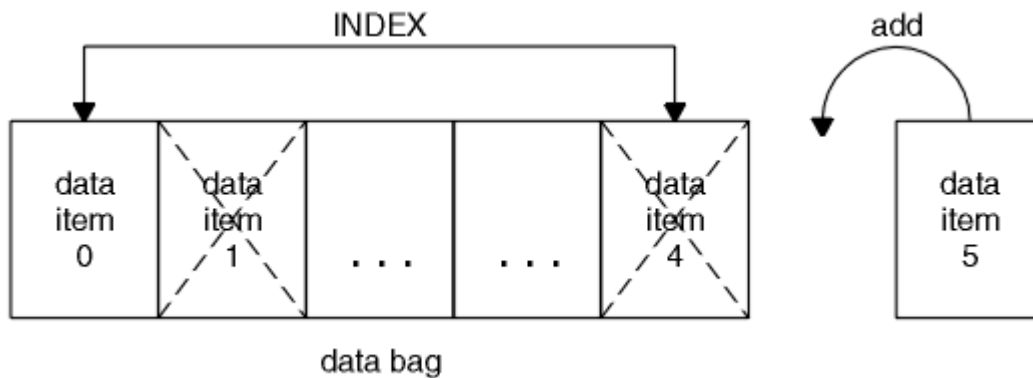


Figura 7. Modificando todos os itens de dados

Nota: O índice preserva a ordem de inserção dentro do pacote, mas pode afetar os índices de outros itens de dados.

A chamada `mqSetInteger` permite que você modifique itens de número inteiro dentro de um pacote. A chamada `mqSetInteger64` permite que você modifique itens de número inteiro de 64 bits. A chamada `mqSetIntegerFilter` permite que você modifique itens de filtro de número inteiro. A chamada `mqSetString` permite que você modifique os itens de sequência de caracteres. A chamada `mqSetStringFilter` permite que você modifique os itens de filtragem de sequência. A chamada `mqSetByteString` permite que você modifique os itens de sequência de bytes. A chamada `mqSetByteStringFilter` permite que você modifique os itens de filtragem de sequência de bytes. Como alternativa, é possível usar essas chamadas para excluir todas as ocorrências existentes do seletor especificado e incluir uma nova ocorrência no final do pacote. O item de dados pode ser um item do usuário ou um item de sistema.

Para ver uma descrição completa dessas chamadas, consulte:

- [mqSetInteger](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [mqSetSequência](#)
- [mqSetStringFilter](#)
- [mqSetByteString](#)
- [mqSetByteStringFiltro](#)

Multi Limpando um pacote através da chamada `mqClearBag`

A chamada `mqClearBag` removerá todos os itens do usuário a partir de um pacote do usuário e reconfigura os itens do sistema para seus valores iniciais. Pacotes do sistema contidos dentro do pacote também são excluídos.

Para obter uma descrição completa da chamada `mqClearBag`, consulte [mqClearBag](#).

Multi Truncando um pacote usando a chamada `mqTruncateBag`

A chamada `mqTruncateBag` reduz o número de itens do usuário em um pacote do usuário através da exclusão dos itens do final do pacote, começando com o item incluído mais recentemente. Por exemplo, ele pode ser utilizado quando utilizar as informações do cabeçalho mesmo para gerar mais de uma mensagem.

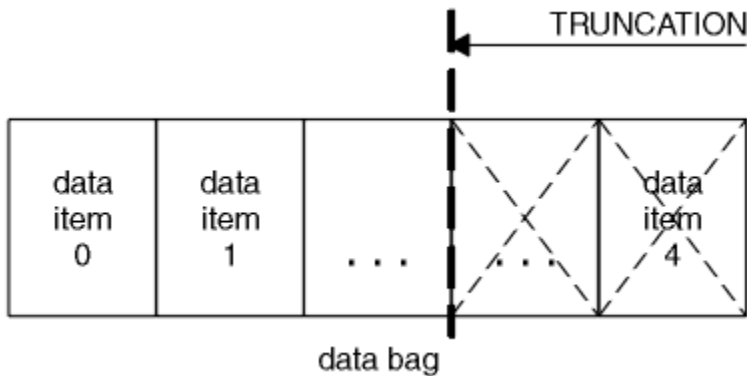


Figura 8. Truncando um pacote

Para obter uma descrição completa da chamada `mqTruncateBag`, consulte [mqTruncateBag](#).

Multi Convertendo pacotes e buffers

Para enviar dados entre aplicativos, primeiramente os dados da mensagem são colocados em um pacote. Em seguida, os dados no pacote são convertidos em uma mensagem PCF usando a chamada `mqBagToBuffer`. A mensagem PCF é enviada para a fila requerida usando a chamada `MQPUT`. Isso é mostrado na Figura do Figura 9 na página 72. Para obter uma descrição completa da chamada `mqBagToBuffer`, consulte [mqBagToBuffer](#).

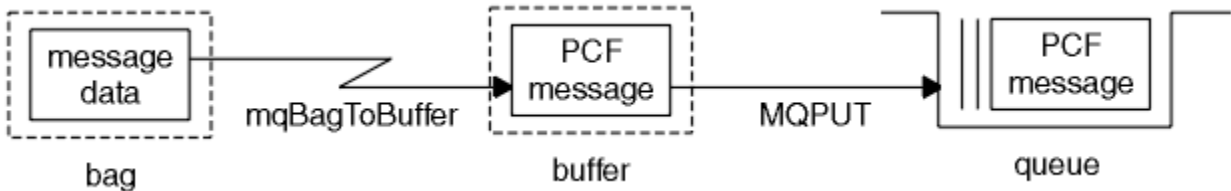


Figura 9. Convertendo pacotes em mensagens PCF

Para receber dados, a mensagem será recebida em um buffer usando a chamada `MQGET`. Os dados no buffer são, então, convertidos em um pacote usando a chamada `mqBufferToBag`, fornecendo o buffer que contém uma mensagem PCF válida. Isso é mostrado na Figura do Figura 10 na página 72. Para obter uma descrição completa da chamada `mqBufferToBag`, consulte [mqBufferToBag](#).

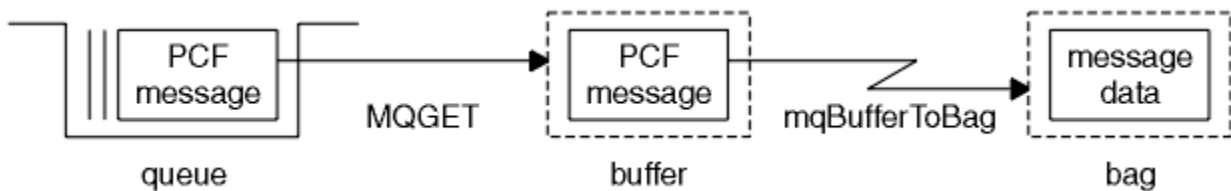


Figura 10. Convertendo para mensagens PCF do pacote

Multi Contando itens de dados

A chamada `mqCountItems` conta o número de itens do usuário, itens do sistema ou ambos, que são armazenados em um pacote de dados e retorna esse número. Por exemplo, `mqCountItems (Bag , 7 , . . .)`, retorna o número de itens no pacote com um seletor de 7. Ele pode contar itens por seletor individual, por seletores de usuário, por seletores de sistema ou por todos os seletores.

Nota: Essa chamada conta o número de itens de dados, não o número de seletores exclusivos no pacote. Um seletor pode ocorrer várias vezes, portanto, pode haver seletores menos exclusivo no pacote de itens de dados.

Para obter uma descrição completa da chamada `mqCountItems`, consulte [mqCountItems](#).

Multi Excluindo os itens de dados

É possível excluir itens de pacotes de várias maneiras. Você pode:

- Remover um ou mais itens do usuário a partir de um pacote. Para obter informações detalhadas, consulte [“Excluindo os itens de dados de um pacote usando a chamada mqDeleteItem”](#) na página 73.
- Exclua todos os itens do usuário de um pacote, ou seja, limpe um pacote. Para obter informações detalhadas, consulte [“Limpendo um pacote através da chamada mqClearBag”](#) na página 71.
- Exclua os itens do usuário do fim de um pacote, ou seja, trunque um pacote. Para obter informações detalhadas, consulte [“Truncando um pacote usando a chamada mqTruncateBag”](#) na página 71.

Multi Excluindo os itens de dados de um pacote usando a chamada mqDeleteItem

A chamada `mqDeleteItem` remove um ou mais itens do usuário a partir de um pacote. O índice é utilizado para excluir um:

1. Uma única ocorrência do seletor especificado. (Consulte [Figura 11](#) na página 73.)

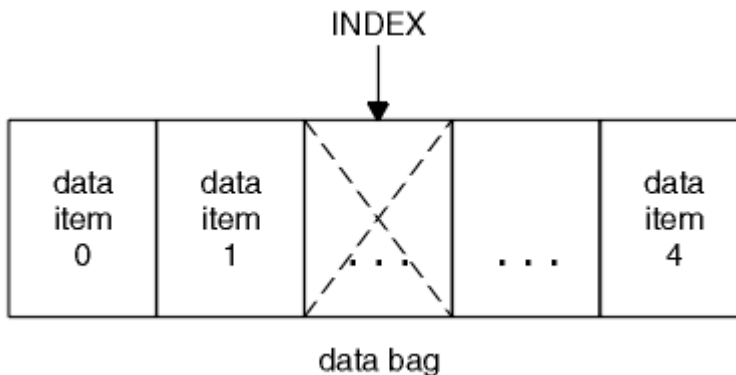


Figura 11. Excluindo um item de dados único

ou

2. Todas as ocorrências do seletor especificado. (Consulte [Figura 12](#) na página 73.)

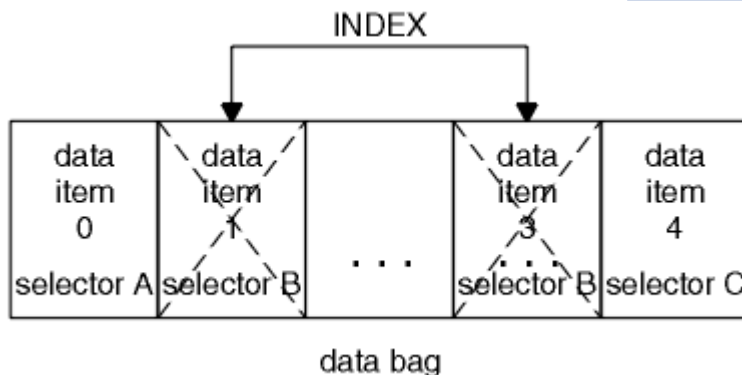


Figura 12. Excluindo todos os itens de dados

Nota: O índice preserva a ordem de inserção dentro do pacote, mas pode afetar os índices de outros itens de dados. Por exemplo, a chamada `mqDeleteItem` não preserva os valores de índice dos itens de dados que seguem o item excluído porque os índices estão reorganizados para preencher a lacuna que permanece do item excluído.

Para obter uma descrição integral da chamada `mqDeleteItem`, consulte [mqDeleteItem](#).

Enviando comandos de administração para o servidor de comandos qm usando a chamada mqExecute

Quando um pacote de dados tiver sido criado e preenchido, uma mensagem de comando administrativo pode ser enviada para o servidor de comandos de um gerenciador de filas usando a chamada mqExecute. Isso controla a troca com o servidor de comandos e retorna respostas em um pacote.

Após ter criado e preenchido o pacote de dados, é possível enviar uma mensagem de comando de administração para o servidor de comandos de um gerenciador de filas. A maneira mais fácil de fazer isso é usando a chamada mqExecute. A chamada mqExecute envia uma mensagem de comando de administração como uma mensagem não persistente e aguarda as respostas. Respostas são retornadas em um pacote de respostas. Estes pode conter informações sobre os atributos relacionados a vários objetos do IBM MQ ou uma série de mensagens de resposta de erro PCF, por exemplo. Portanto, o pacote de respostas poderia conter um código de retorno somente ou poderia conter *pacotes aninhados*.

As mensagens de resposta são colocadas em pacotes do sistema que são criados pelo sistema. Por exemplo, para consultas sobre os nomes de objetos, um pacote do sistema é criado para conter esses nomes de objeto e o pacote é inserido no pacote do usuário. Identificadores para esses pacotes são inseridos no pacote de respostas e o pacote aninhado pode ser acessado pelo seletor MQHA_BAG_HANDLE. O pacote do sistema permanece no armazenamento, se ele não for excluído, até que o pacote de respostas seja excluído.

O conceito de *aninhando* será mostrado em [Figura 13 na página 74](#).

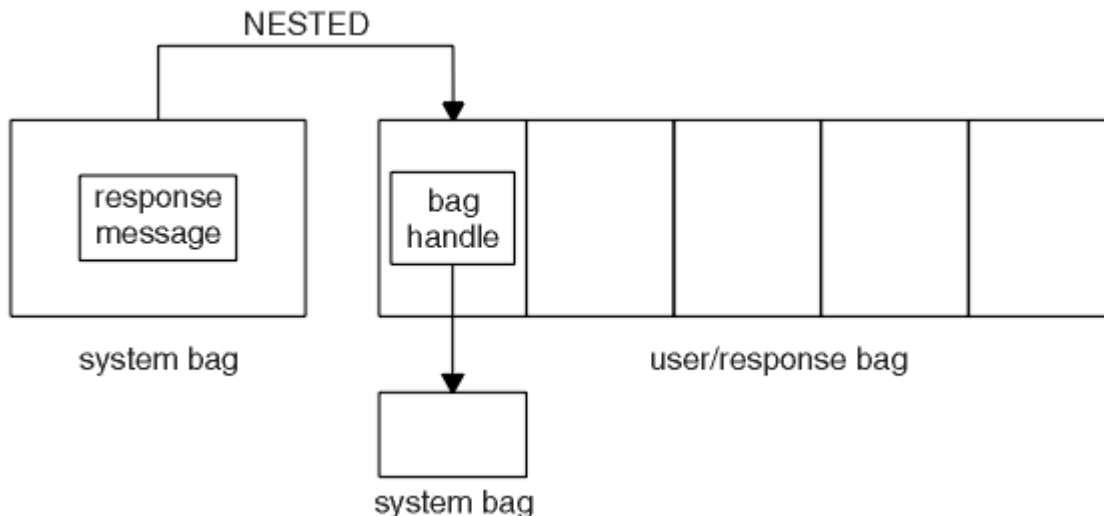


Figura 13. Agrupando

Como entrada para a chamada mqExecute, deve-se fornecer:

- Um MQI identificador de conexão.
- O comando a ser executado. Isso deve ser um dos valores MQCMD_*

Nota: Se este valor não for reconhecido pelo MQAI, o valor ainda será aceito. No entanto, se a chamada mqAddInquiry foi utilizada para inserir valores no pacote, esse parâmetro deve ser um comando INQUIRE reconhecido pelo MQAI. Ou seja, o parâmetro deve ter a forma MQCMD_INQUIRE_*.

- Opcionalmente, um identificador do pacote que contém opções que controlam o processamento da chamada. Este é também onde é possível especificar o tempo máximo em milissegundos que a MQAI deve aguardar para cada mensagem de resposta.
- Um identificador do pacote de administração que contém detalhes do comando de administração a ser emitido.
- Um identificador do pacote de respostas que recebe as mensagens de resposta.

Os identificadores a seguir são opcionais:

- Um identificador de objeto da fila para onde o comando de administração deve ser colocado.

Se nenhum identificador de objeto for especificado, o comando de administração é colocada na SYSTEM.ADMIN.COMMAND.QUEUE que pertence ao gerenciador de filas atualmente conectados. Esse é o padrão.

- Um identificador de objeto da fila para onde as mensagens de resposta devem ser colocadas.


É possível optar por colocar as mensagens de resposta em uma fila dinâmica que é criada automaticamente pelo MQAI. A fila criada existe para a duração da chamada somente e é excluída pelo MQAI na saída da chamada mqExecute.

Para obter exemplos de usos da chamada mqExecute, consulte [Código de exemplo](#)

Administração usando o REST API

É possível usar a administrative REST API para administrar objetos do IBM MQ, como gerenciadores de filas e filas e agentes do Managed File Transfer e transferências. As informações são enviadas para a administrative REST API e recebidas dela no formato JSON. Essas APIs RESTful podem ajudar a integrar a administração do IBM MQ ao conjunto de ferramentas popular de DevOps e automação.

Antes de começar

Nota:  O administrative REST API não está disponível em uma instalação independente do IBM MQ Web Server. Para obter mais informações sobre as opções de instalação para o componente IBM MQ que executa o administrative REST API, consulte [O IBM MQ Console e REST API](#).

Para obter informações de referência sobre os recursos REST disponíveis, consulte [A referência do administrative REST API](#).

Procedimento



- [“Introdução ao administrative REST API” na página 75](#)
- [“Usando o administrative REST API” na página 79](#)
- [“Administração remota usando a REST API” na página 80](#)
- [“Registros de data e hora da REST API” na página 85](#)
- [“REST API manipulação de erros” na página 85](#)
- [“Descoberta de REST API” na página 88](#)
- [“Suporte ao idioma nacional da REST API” na página 89](#)

Introdução ao administrative REST API

Comece a usar rapidamente a administrative REST API e experimente algumas solicitações de exemplo usando o cURL para criar, atualizar, visualizar e excluir uma fila.

Antes de começar

Para começar a usar a administrative REST API, os exemplos nesta tarefa possuem os requisitos a seguir:

- Os exemplos usam o cURL para fazer solicitações REST para exibir informações sobre gerenciadores de filas no sistema e para criar uma fila, atualizar, visualizar e excluir uma fila. Portanto, para concluir esta tarefa, será necessário ter o cURL instalado no sistema.
- Para concluir essa tarefa, deve-se ser um usuário com determinados privilégios para que seja possível usar o comando **dspmweb**:
 -  No z/OS, deve-se ter autoridade para executar o comando **dspmweb** e acesso de gravação ao arquivo mqwebuser.xml.
 -  Em todos os outros sistemas operacionais, deve-se ser um [usuário privilegiado](#).

Procedimento

1. Assegure-se de que você tenha configurado o servidor mqweb para uso pelo administrative REST API, o administrative REST API para MFT, o messaging REST API ou IBM MQ Console.

Para obter mais informações sobre como configurar o servidor mqweb com um registro básico, consulte [Configuração básica para o servidor mqweb](#)

2. 

Em z/OS, configure a variável de ambiente WLP_USER_DIR para que seja possível usar o comando **dspmqweb**. Configure a variável para apontar para a configuração do servidor mqweb inserindo o comando a seguir:

```
export WLP_USER_DIR=WLP_user_directory
```

em que *WLP_user_directory* é o nome do diretório transmitido para crtmqweb. Por exemplo:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Para obter mais informações, consulte [Criando o servidor mqweb](#).

3. Determinar o REST API URL digitando o seguinte comando:

```
dspmqweb status
```

Os exemplos nas etapas a seguir pressupõem que seu REST API URL é o URL padrão `https://localhost:9443/ibmmq/rest/v1/`. Se a URL for diferente do padrão, substitua-a nas etapas a seguir.

4. Experimente uma solicitação GET no recurso `qmgr` usando a autenticação básica com o usuário `mqadmin`:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/qmgr -X GET -u mqadmin:mqadmin
```

5. Crie, exiba, altere e exclua uma fila usando o recurso `mqsc`:

Este exemplo usa um gerenciador de filas QM1. Crie um gerenciador de filas com o mesmo nome ou substitua um gerenciador de filas existente no sistema.

- a) Faça uma solicitação de POST no recurso `mqsc` para criar a fila local:

No corpo da solicitação, o nome da nova fila é configurado para Q1. A autenticação básica é usada e um cabeçalho de HTTP `ibm-mq-rest-csrf-token` com um valor arbitrário é configurado na solicitação REST de cURL. Esse cabeçalho adicional é necessário para solicitações POST, PATCH e DELETE:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "define", "qualifier": "qlocal", "name": "Q1"}'
```

- b) Faça uma solicitação de POST no recurso `mqsc` para visualizar a fila local criada na etapa [“5.a”](#) na [página 76](#):

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1"}'
```

- c) Faça uma solicitação de POST no recurso `mqsc` para atualizar a descrição da fila:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --
```

```
data {"type": "runCommandJSON", "command": "alter", "qualifier": "qlocal",
"name": "Q1", "parameters": {"descr": "new description" }}
```

- d) Faça uma solicitação de POST no recurso mqsc para visualizar a nova descrição da fila. Especifique o atributo **responseParameters** no corpo da solicitação para que a resposta inclua o campo de descrição:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u
mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --
data {"type": "runCommandJSON", "command": "display", "qualifier": "qlocal",
"name": "Q1", "responseParameters": [{"descr"}]}
```

- e) Faça uma solicitação de POST no recurso mqsc para excluir a fila:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u
mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --
data {"type": "runCommandJSON", "command": "delete", "qualifier": "qlocal",
"name": "Q1" }
```

- f) Faça uma solicitação de POST no recurso mqsc para provar que a fila foi excluída:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u
mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --
data {"type": "runCommandJSON", "command": "display", "qualifier": "qlocal",
"name": "Q1" }
```

Como proceder a seguir

- Os exemplos usam autenticação básica para proteger a solicitação. Em vez disso, é possível usar autenticação baseada em token ou autenticação baseada em cliente. Para obter mais informações, consulte [Usando a autenticação de certificado de cliente com o REST API e o IBM MQ Console](#) e [Usando a autenticação baseada em token com o REST API](#).
- Saiba mais sobre como usar o administrative REST API e construir URLs com parâmetros de consulta: [“Usando o administrative REST API” na página 79](#).
- Procure as informações de referência para os recursos administrative REST API disponíveis e todos os parâmetros de consulta opcionais disponíveis: [Referência administrative REST API](#).
- Saiba como usar a administrative REST API para administrar objetos do IBM MQ em sistemas remotos: [“Administração remota usando a REST API” na página 80](#).
- Aprenda a usar o administrative REST API com MFT: [“Iniciando com o REST API para MFT” na página 77](#).
- Descubra o messaging REST API, uma interface RESTful para o sistema de mensagens do IBM MQ: [Sistema de mensagens usando o REST API](#).
- Descubra o IBM MQ Console, uma GUI baseada em navegador: [“Administração usando o IBM MQ Console” na página 92](#).

Iniciando com o REST API para MFT

Comece a usar rapidamente a administrative REST API para o Managed File Transfer e experimente algumas solicitações de exemplo para visualizar o status do agente do MFT e para visualizar uma lista de transferências.

Antes de começar

- Os exemplos usam o cURL para enviar solicitações REST para visualizar uma lista de transferências e visualizar o status do agente do MFT. Portanto, para concluir esta tarefa, será necessário ter o cURL instalado no sistema.
- Para concluir essa tarefa, deve-se ser um usuário com determinados privilégios para que seja possível usar o comando [dsprmweb](#):

- **z/OS** No z/OS, deve-se ter autoridade para executar o comando **dspmweb** e acesso de gravação ao arquivo `mqwebuser.xml`.
- **Multi** Em todos os outros sistemas operacionais, deve-se ser um usuário privilegiado.

Procedimento

1. Assegure-se de que o servidor mqweb esteja configurado para a administrative REST API para o MFT:

- Assegure-se de que você tenha configurado o servidor mqweb para uso pelo administrative REST API, o administrative REST API para MFT, o messaging REST API ou IBM MQ Console. Para obter mais informações sobre como configurar o servidor mqweb com um registro básico, consulte [Configuração básica para o servidor mqweb](#)
- Se o servidor mqweb estiver configurado, assegure-se de que a etapa 8 da [Configuração básica para o servidor mqweb](#) tenha sido concluída para ativar a administrative REST API para o MFT.

2. **z/OS**

Em z/OS, configure a variável de ambiente `WLP_USER_DIR` para que seja possível usar o comando **dspmweb**. Configure a variável para apontar para a configuração do servidor mqweb inserindo o comando a seguir:

```
export WLP_USER_DIR=WLP_user_directory
```

em que `WLP_user_directory` é o nome do diretório transmitido para `crtmqweb`. Por exemplo:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Para obter mais informações, consulte [Criando o servidor mqweb](#).

3. Determinar o REST API URL digitando o seguinte comando:

```
dspmweb status
```

Os exemplos nas etapas a seguir pressupõem que seu REST API URL é o URL padrão `https://localhost:9443/ibmmq/rest/v1/`. Se a URL for diferente do padrão, substitua-a nas etapas a seguir.

4. Faça uma solicitação GET no recurso `agent` para retornar detalhes básicos sobre todos os agentes, incluindo o nome, o tipo e o estado:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/agent/ -X GET -u mftadmin:mftadmin
```

5. Crie algumas transferências para serem exibidas usando o comando **fteCreateTransfer**.

O servidor mqweb armazena em cache informações sobre transferências e retorna essas informações quando uma solicitação é feita. Esse cache é reconfigurado quando o servidor mqweb é reiniciado. É possível ver se o servidor foi reiniciado visualizando os arquivos `console.log` e `messages.log` ou z/OS, consultando a saída da tarefa iniciada.

6. Faça uma solicitação GET no recurso `transfer` para retornar detalhes de até quatro transferências que foram feitas desde que o servidor mqweb foi iniciado:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/transfer?limit=4 -X GET -u mftadmin:mftadmin
```

Como proceder a seguir

- Os exemplos usam autenticação básica para proteger a solicitação. Em vez disso, é possível usar autenticação baseada em token ou autenticação baseada em cliente. Para obter mais informações, consulte [Usando a autenticação baseada em token com a REST API](#) e [Usando a autenticação por certificado de cliente com a REST API](#) e o IBM MQ Console.

- Saiba mais sobre como usar o administrative REST API e construir URLs com parâmetros de consulta: [“Usando o administrative REST API” na página 79.](#)
- Procure, nas informações de referência, os recursos da administrative REST API for MFT disponíveis e todos os parâmetros de consulta opcionais disponíveis: [Referência da administrative REST API.](#)
- Descubra o messaging REST API, uma interface RESTful para o sistema de mensagens do IBM MQ: [Sistema de mensagens usando o REST API.](#)
- Descubra o IBM MQ Console, uma GUI baseada em navegador: [“Administração usando o IBM MQ Console” na página 92.](#)

Usando o administrative REST API

Quando você usa a administrative REST API, você chama métodos de HTTP em URLs que representam os vários objetos do IBM MQ, como gerenciadores de filas ou filas. O método de HTTP, por exemplo, POST, representa o tipo de ação a ser executado no objeto representado pela URL. Informações adicionais sobre a ação podem ser fornecidas em JSON como parte da carga útil do método de HTTP ou codificadas em parâmetros de consulta. Informações sobre o resultado da execução da ação podem ser retornadas como o corpo da resposta HTTP.

Antes de começar

Considere estas coisas antes de usar o administrative REST API:

- Você deve se autenticar no servidor mqweb para usar o administrative REST API. É possível autenticar usando a autenticação básica HTTP, a autenticação por certificado de cliente ou autenticação baseada em token. Para obter mais informações sobre como usar esses métodos, consulte [IBM MQ Console e REST API security.](#)
- A REST API faz distinção entre maiúsculas e minúsculas. Por exemplo, um HTTP GET na URL a seguir não exibe informações se o gerenciador de filas é chamado qmgr1.

```
/ibmmq/rest/v1/admin/qmgr/QMGR1
```

- Nem todos os caracteres que podem ser usados em nomes de objetos IBM MQ podem ser codificados diretamente em uma URL. Para codificar esses caracteres corretamente, deve-se usar a codificação de URL apropriada:
 - Uma barra, /, deve ser codificada como %2F.
 - Um sinal de porcentagem, %, deve ser codificado como %25.
- Devido ao comportamento de alguns navegadores, não nomeie objetos usando apenas caracteres de ponto ou barra.

Sobre esta tarefa

Quando você usa a REST API para executar uma ação em um objeto, você primeiro precisa construir uma URL para representar esse objeto. Cada URL se inicia com um prefixo, que descreve o nome do host e a porta para os quais enviar a solicitação. O restante da URL descreve um objeto específico ou um conjunto de objetos, conhecido como recurso.

A ação a ser executada no recurso define se a URL precisará de parâmetros de consulta ou não. Ela também define o método de HTTP usado e se informações adicionais serão enviadas para a URL ou retornadas dela no formato JSON. As informações adicionais podem formar parte da solicitação de HTTP ou serem retornadas como parte da resposta HTTP.

Depois de construir a URL e criar uma carga útil JSON opcional para enviar na solicitação de HTTP, é possível enviar a solicitação de HTTP para o IBM MQ. É possível enviar a solicitação usando a implementação HTTP que é construída na linguagem de programação de sua escolha. Também é possível enviar as solicitações usando ferramentas de linha de comandos, como cURL, ou um navegador da web ou o complemento de navegador da web.

Importante: Deve-se, no mínimo, realizar as etapas [“1.a” na página 80](#) e [“1.b” na página 80](#).

Procedimento

1. Construa a URL:

- a) Determine a URL do prefixo inserindo o comando a seguir:

```
dspmweb status
```

A URL que você deseja usar inclui a frase `/ibmmq/rest/`.

- b) Inclua o recurso no caminho da URL.

Os recursos do IBM MQ a seguir estão disponíveis:

- [/admin/installation](#)
- [/admin/qmgr](#)
- [/admin/queue](#)
- [/admin/subscription](#)
- [/admin/channel](#)
- [/action/qmgr/{/mqsc](#)

Os recursos do Managed File Transfer a seguir estão disponíveis:

- [/admin/agente](#)
- [/admin/transfer](#)
- [/admin/monitor](#)

Por exemplo, para interagir com gerenciadores de filas, inclua `/qmgr` na URL do prefixo para criar a URL a seguir:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr
```

- c) Opcional: Inclua qualquer segmento de caminho opcional adicional na URL.

Nas informações de referência para cada tipo de objeto, os segmentos opcionais podem ser identificados na URL pelas chaves que a circundam `{ }`.

Por exemplo, inclua o nome do gerenciador de filas QM1 na URL para criar a URL a seguir:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr/QM1
```

- d) Opcional: Inclua um parâmetro de consulta opcional na URL.

Inclua um ponto de interrogação, `?`, nome da variável, sinal de igual `=` e um valor ou lista de valores para a URL.

Por exemplo, para solicitar todos os atributos do gerenciador de filas QM1, crie a URL a seguir:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr/QM1?attributes=*
```

- e) Inclua parâmetros de consulta opcionais adicionais na URL.

Inclua um e comercial, `&`, na URL e repita [step d.](#)

2. Chame o método de HTTP relevante na URL. Especifique qualquer carga útil JSON opcional e forneça as credenciais de segurança apropriadas para autenticação. Por exemplo:

- Use a implementação HTTP/REST de sua linguagem de programação escolhida.
- Use uma ferramenta, como um complemento de navegador do cliente REST ou cURL.

Administração remota usando a REST API

É possível usar a REST API para administrar gerenciadores de filas remotas e os objetos do IBM MQ que estão associados a esses gerenciadores de filas. Essa administração remota inclui os gerenciadores de filas que estão no mesmo sistema, mas não na mesma instalação do IBM MQ que o servidor mqweb.

Portanto, é possível usar a REST API para administrar toda a sua rede do IBM MQ com apenas uma instalação que executa o servidor mqweb. Para administrar gerenciadores de filas remotas, deve-se configurar o gateway da administrative REST API para que pelo menos um gerenciador de filas na mesma instalação que o servidor mqweb aja como um gerenciador de filas de gateway. Em seguida, é possível especificar o gerenciador de filas remotas na URL de recurso da REST API para executar a ação administrativa especificada.

Antes de começar

É possível evitar a administração remota desativando o gateway da administrative REST API. Para obter mais informações, consulte [Configurando o gateway da administrative REST API](#).

Para usar o gateway da administrative REST API, as condições a seguir devem ser atendidas:

- O servidor mqweb deve ser configurado e iniciado. Para obter mais informações sobre como configurar e iniciar o servidor mqweb, veja [“Introdução ao administrative REST API”](#) na página 75.
- O gerenciador de filas que você deseja configurar como o gerenciador de filas de gateway deve estar na mesma instalação que o servidor mqweb.
- O gerenciador de filas remotas que você deseja administrar deve ser o IBM MQ 8.0 ou mais recente.
- Deve-se assegurar que quaisquer atributos especificados em sua solicitação sejam válidos para o sistema ao qual você está enviando a solicitação. Por exemplo, se o gerenciador de filas de gateway está no Windows e o gerenciador de filas remotas está no z/OS, não é possível solicitar que o atributo `dataCollection.statistics` seja retornado para uma solicitação de HTTP GET no recurso `queue`.
- Deve-se assegurar que quaisquer atributos especificados em sua solicitação sejam válidos para o nível do IBM MQ para o qual você está enviando a solicitação. Por exemplo, se o gerenciador de filas remotas está executando o IBM MQ 8.0, não é possível solicitar que o atributo `extended.enableMediaImageOperations` seja retornado para uma solicitação de HTTP GET no recurso `queue`.
- Deve-se usar um destes recursos REST suportados:
 - `/queue`
 - `/subscription`
 - `/channel`
 - `/mqsc`
 - `/qmgr`

O recurso `/qmgr` retorna somente um subconjunto dos atributos quando você consulta um gerenciador de filas remotas: `name`, `status.started`, `status.channelInitiatorState`, `status.ldapConnectionState`, `status.connectionCount` e `status.publishSubscribeState`.

Sobre esta tarefa

Para usar o gateway da administrative REST API para administrar gerenciadores de filas remotas, deve-se preparar os gerenciadores de filas para administração remota. Ou seja, deve-se configurar filas de transmissão, listeners e canais emissores e receptores entre o gerenciador de filas de gateway e o gerenciador de filas remotas. É possível então enviar uma solicitação REST para o gerenciador de filas remotas, especificando o gerenciador de filas na URL do recurso. O gerenciador de filas de gateway é especificado usando o comando **setmqweb** para configurar o atributo `mqRestGatewayQmgr` para o nome do gerenciador de filas de gateway ou enviando o nome do gerenciador de filas de gateway em um cabeçalho que é enviado com a solicitação. A solicitação é enviada por meio do gerenciador de filas de gateway para o gerenciador de filas remotas. A resposta é retornada com um cabeçalho que indica o gerenciador de filas que foi usado como o gerenciador de filas de gateway.

Procedimento

1. Configure as comunicações entre o gerenciador de filas de gateway e os gerenciadores de filas remotas que você deseja administrar. Essas etapas de configuração são as mesmas etapas que são necessárias para configurar a administração remota por runmqsc e PCF.
Para obter informações adicionais sobre essas etapas, consulte [“Configurando Gerenciadores de Filas para Administração Remota”](#) na página 200.
2. Configure a segurança nos gerenciadores de filas remotas:
 - a) Assegure-se de que os IDs de usuário relevantes existam no sistema em que o gerenciador de filas remotas é executado. O ID do usuário que deve existir no sistema remoto depende da função do usuário da REST API:
 - Se o usuário da REST API estiver no grupo MQWebAdmin ou MQWebAdminRO, o ID do usuário que iniciou o servidor mqweb deverá existir no sistema remoto. No IBM MQ Appliance, o usuário que inicia o servidor mqweb é mqsystem.
 - Se o usuário da REST API estiver no grupo MQWebUser, esse ID do usuário da REST API deverá existir no sistema remoto.
 - b) Assegure-se de que os IDs de usuário relevantes recebam os níveis necessários de autoridade para acessar os recursos apropriados da REST API no gerenciador de filas remotas:
 - Autoridade para colocar mensagens no SYSTEM.ADMIN.COMMAND.QUEUE.
 - Autoridade para colocar mensagens no SYSTEM.REST.REPLY.QUEUE.
 - Autoridade para acessar as filas de transmissão que são definidas para administração remota.
 - Autoridade para exibir atributos do gerenciador de filas.
 - Autoridade para executar as solicitações REST. Para obter mais informações, consulte a seção [Requisitos de segurança dos tópicos de referência de recursos da REST API](#).
3. Configure qual gerenciador de filas locais é usado como o gateway. É possível configurar um gerenciador de filas de gateway padrão, especificar o gerenciador de filas de gateway em um cabeçalho HTTP ou usar uma combinação de ambas as abordagens:
 - Configure um gerenciador de filas de gateway padrão usando o comando **setmqweb**:

```
setmqweb properties -k mqRestGatewayQmgr -v qmgrName
```

em que *qmgrName* é o nome do gerenciador de filas de gateway.
Esse gerenciador de filas de gateway é usado quando ambas as instruções a seguir são verdadeiras:
 - Um gerenciador de filas não está especificado no cabeçalho `ibm-mq-rest-gateway-qmgr` de uma solicitação REST.
 - O gerenciador de filas especificado na URL do recurso REST API não é um gerenciador de filas local.
 - Configure o gerenciador de filas de gateway em cada solicitação REST configurando o cabeçalho de HTTP `ibm-mq-rest-gateway-qmgr` para o nome do gerenciador de filas de gateway.
4. Inclua o nome do gerenciador de filas remotas que você deseja administrar na URL do recurso.
Por exemplo, para obter uma lista de filas do gerenciador de filas remotas `remoteQM`, use a URL a seguir:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/remoteQM/queue
```

Resultados

Um cabeçalho `ibm-mq-rest-gateway-qmgr` é retornado com a resposta REST. Esse cabeçalho especifica qual gerenciador de filas foi usado como o gerenciador de filas de gateway.

Se você tiver dificuldade em usar a administrative REST API para administrar gerenciadores de filas remotas:

- Verifique se o gerenciador de filas remotas está em execução.
- Verifique se o servidor de comandos está em execução no sistema remoto.
- Verifique se o intervalo de desconexão do canal não expirou. Por exemplo, se um canal foi iniciado, mas foi encerrado após algum tempo. Isso é especialmente importante se você iniciar os canais manualmente.

Exemplo

No exemplo a seguir, há três instalações do IBM MQ em duas máquinas. No Machine 1, há um Installation 1 e um Installation 2. No Machine 2, há um Installation 3. Um servidor mqweb é configurado para Installation 1. Há um único gerenciador de filas em cada instalação e esses gerenciadores de filas são configurados para administração remota. Ou seja, os listeners, canais e filas a seguir são configurados e iniciados:

- No gerenciador de filas QM1, em Installation 1, em Machine 1:
 - Canal emissor QM1.to.QM2
 - Canal receptor QM2.to.QM1
 - Canal emissor QM1.to.QM3
 - Canal receptor QM3.to.QM1
 - Fila de transmissão QM2
 - Fila de transmissão QM3
 - Um listener configurado na porta 1414
- No gerenciador de filas QM2, em Installation 2, em Machine 1:
 - Canal emissor QM2.to.QM1
 - Canal receptor QM1.to.QM2
 - Fila de transmissão QM1
 - Um listener configurado na porta 1415
- No gerenciador de filas QM3, em Installation 3, em Machine 2:
 - Canal emissor QM3.to.QM1
 - Canal receptor QM1.to.QM3
 - Fila de transmissão QM1
 - O listener padrão

Uma fila, Qon2 é definida em QM2 e uma fila Qon3 é definida em QM3.

O usuário mquser é definido em ambas as máquinas, é concedida a ele a função MQWebAdmin na REST API e a autoridade para acessar as filas apropriadas em cada gerenciador de filas.

O comando setmqweb é usado para configurar o gerenciador de filas QM1 como o gerenciador de filas de gateway padrão.

O diagrama a seguir mostra esta configuração:

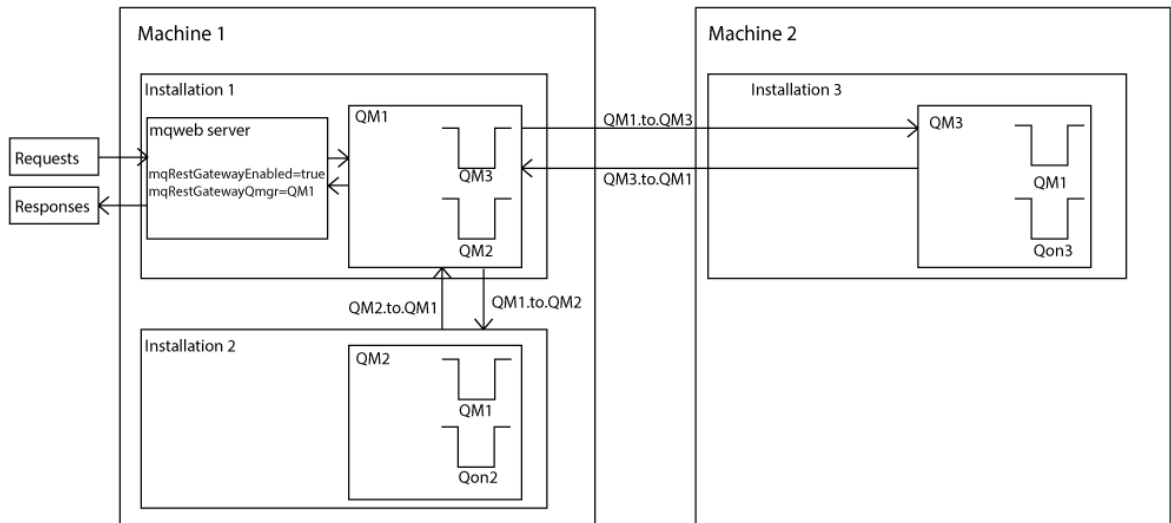


Figura 14. Diagrama de configuração de exemplo para administração remota usando a REST API.

A solicitação REST a seguir é enviada para o servidor mqweb:

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM2/queue?
attributes=general.isTransmissionQueue
```

A resposta a seguir é recebida:

```
{
  "queue" :
  [ {
    "general": {
      "isTransmissionQueue": true
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false
    },
    "name" : "Qon2",
    "type" : "local"
  }
]
```

A solicitação REST a seguir é enviada para o servidor mqweb:

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM3/queue?
attributes=general.isTransmissionQueue,general.description
```

A resposta a seguir é recebida:

```
{
  "queue" :
  [ {
    "general": {
      "isTransmissionQueue": true,
      "description": "Transmission queue for remote admin."
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false,
      "description": "A queue on QM3."
    }
  }
]
```

```
    },
    "name" : "Qon3",
    "type" : "local"
  }
}
```

Registros de data e hora da REST API

Quando as informações de data e hora são retornadas pela administrative REST API, elas são retornadas na Hora Universal Coordenada (UTC) e em um formato configurado.

A data e hora é retornada no formato do registro de data e hora a seguir:

```
YYYY-MM-DDTHH:mm:ss:sssZ
```

Por exemplo, 2012-04-23T18:25:43.000Z, em que Z indica que o fuso horário é Hora Universal Coordenada (UTC).

A precisão desse registro de data e hora não é garantida. Por exemplo, se o servidor mqweb não for iniciado no mesmo fuso horário que o gerenciador de filas que está especificado na URL do recurso, o registro de data e hora poderá não ser exato. Além disso, se os ajustes de horário de verão forem necessários, o registro de data e hora poderá não ser exato.

REST API manipulação de erros

A REST API relata erros retornando um código de resposta HTTP apropriado, por exemplo, 404 (Não localizado) e uma resposta JSON. Qualquer código de resposta de HTTP que não está no intervalo 200 - 299 é considerado um erro.

O formato da resposta de erro

A resposta está no formato JSON em codificação UTF-8. Ela contém objetos JSON aninhados:

- Um objeto JSON externo que contém uma única matriz JSON chamada `error`.
- Cada elemento na matriz é um objeto JSON que representa informações sobre um erro. Cada objeto JSON contém as propriedades a seguir:

Tipo

Sequência.

O tipo de erro.

messageId

Sequência.

Um identificador exclusivo para a mensagem no formato MQWBnnnnX. Esse identificador tem os elementos a seguir:

MQWB

Um prefixo que mostra que a mensagem foi originada na API de REST do IBM MQ.

nnnn

Um número exclusivo que identifica a mensagem.

X

Uma única letra que denota a gravidade da mensagem:

- I se uma mensagem é puramente informativa.
- W se uma mensagem é aviso de um problema.
- E se uma mensagem indica que um erro ocorreu.
- S se uma mensagem indica que um erro grave ocorreu.

mensagem

Sequência.

Uma descrição do erro.

explicação

Sequência.


Uma explicação do erro.

ação

Sequência.

Uma descrição das etapas que podem ser usadas para resolver o erro.

qmgrName

 Este campo está disponível somente para o z/OS no qual o gerenciador de filas é um membro do grupo de filas compartilhadas. Deve-se ter especificado o parâmetro de consulta opcional **commandScope** ou o atributo **queueSharingGroupDisposition**.

Sequência.

O nome do gerenciador de filas que teve o erro.

Este campo não é aplicável para a messaging REST API.

completionCode

Este campo está disponível somente quando **type** é `pcf`, `java` ou `rest`.

Número.

O código de conclusão do MQ associado à falha.

reasonCode

Este campo está disponível somente quando **type** é `pcf`, `java` ou `rest`.

Número.

O código de razão do MQ associado à falha.

exceções

Este campo está disponível somente quando **type** é `java`.

Matriz.

Uma matriz de exceções de cadeia Java ou JMS Cada elemento da matriz de exceções contém uma matriz de sequência **stackTrace**.

A matriz de sequência **stackTrace** contém os detalhes de cada exceção divididos em linhas.

Erros com grupos de filas compartilhadas



Em um grupo de filas compartilhadas, é possível especificar um parâmetro de consulta opcional de **commandScope** para determinados comandos. Esse parâmetro permite que o comando seja propagado para outros gerenciadores de filas no grupo de filas compartilhadas. Qualquer um desses comandos pode falhar independentemente, resultando em alguns comandos bem-sucedidos e alguns comandos com falha para o grupo de filas compartilhadas.

Nos casos em que um comando falha parcialmente, um código de erro HTTP 500 é retornado. Para cada gerenciador de filas que gerou uma falha, as informações sobre essa falha são retornadas como um elemento na matriz JSON `error`. Para cada gerenciador de filas que executou o comando com sucesso, o nome do gerenciador de filas é retornado como um elemento em uma matriz JSON `success`.

Exemplos

- O exemplo a seguir mostra a resposta de erro para uma tentativa de obter informações sobre um gerenciador de filas que não existe:

```
"error": [
  {
    "type": "rest",
    "messageId": "MQWB0009E",
    "message": "MQWB0009E: Could not query the queue manager 'QM1'",
    "explanation": "The MQ REST API was invoked specifying a queue manager name which
```

```
cannot be located.",
  "action": "Resubmit the request with a valid queue manager name or no queue manager
name, to retrieve a list of queue managers. "
}
]
```

- ▶ **z/OS** O exemplo a seguir mostra a resposta de erro para uma tentativa de excluir uma fila em um grupo de filas compartilhadas que não existe para alguns gerenciadores de filas:

```
"error" : [
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason
code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be
located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name
to retrieve a list of queues.",
    "qmgrName": "QM1"
  },
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason
code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be
located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name
to retrieve a list of queues.",
    "qmgrName": "QM2"
  }
],
"success" : [{"qmgrName": "QM3"}, {"qmgrName": "QM4"}]
```

Erros com Solicitações MFT

Se os serviços do MFT REST API não estiverem ativados e você chamar o MFT REST API, a exceção a seguir será recebida:

```
{"error": [{
  "action": "Enable the Managed File Transfer REST API and resubmit the request.",
  "completionCode": 0,
  "explanation": "Managed File Transfer REST calls are not permitted as the service is
disabled.",
  "message": "MQWB0400E: Managed File Transfer REST API is not enabled.",
  "msgId": "MQWB0400E",
  "reasonCode": 0,
  "type": "rest"
}]}
```

Se os serviços MFT REST API estiverem ativados e o gerenciador de filas de coordenação não estiver configurado no arquivo `mqwebuser.xml`, você receberá a seguinte exceção:

```
{"error": [{
  "action": "Set the coordination queue manager name and restart the mqweb server.",
  "completionCode": 0,
  "explanation": "Coordination queue manager name must be set before using Managed File
Transfer REST services.",
  "message": "MQWB0402E: Coordination queue manager name is not set.",
  "msgId": "MQWB0402E",
  "reasonCode": 0,
  "type": "rest"
}]}
```

Descoberta de REST API

A documentação da REST API está disponível na IBM Documentation e no formato Swagger. Swagger é uma abordagem usada geralmente para documentar APIs de REST. A documentação do Swagger para o REST API pode ser visualizada ativando o recurso API Discovery (apiDiscovery) no servidor mqweb.

Antes de começar

Stabilized

Importante: O recurso apiDiscovery foi estabilizado.. Ainda é possível usar esse recurso No momento, o IBM MQ não suporta o uso do recurso mpOpenAPI

Deve-se ativar a segurança do servidor mqweb para visualizar a documentação do Swagger usando a descoberta de API. Para obter mais informações sobre as etapas necessárias para ativar a segurança, consulte [Segurança do IBM MQ Console e da REST API](#).

Procedimento

1. Localize o arquivo mqwebuser.xml em um dos seguintes diretórios:

- ALW `MQ_DATA_PATH/web/installations/installationName/servers/mqweb`
- z/OS `WLP_user_directory/servers/mqweb`

Em que `WLP_user_directory` é o diretório que foi especificado quando o script `crtmqweb` foi executado para criar a definição do servidor mqweb.

2. Inclua o XML apropriado no arquivo mqwebuser.xml:

- Se as tags `<featureManager>` existirem no arquivo mqwebuser.xml, inclua o XML a seguir dentro das tags `<featureManager>`:
- Se as tags `<featureManager>` não existirem em seu arquivo mqwebuser.xml, inclua o XML a seguir dentro das tags `<server>`:

```
<featureManager>
  <feature>apiDiscovery-1.0</feature>
</featureManager>
```

3. Visualize a documentação do Swagger usando um dos métodos a seguir:

- Exiba uma página da web que você pode procurar e teste a REST API inserindo a URL a seguir em um navegador:

```
https://host:port/ibm/api/explorer
```

Além de autenticar cada solicitação, deve-se incluir um cabeçalho `ibm-mq-rest-csrf-token` para cada solicitação POST, PATCH ou DELETE. Os conteúdos desse cabeçalho podem ser qualquer sequência, incluindo em branco.

Esse cabeçalho da solicitação é usado para confirmar que as credenciais que estão sendo usadas para autenticar a solicitação estão sendo usadas pelo proprietário das credenciais. Ou seja, o token é usado para evitar ataques de falsificação de solicitação entre sites.

- Recupere um único documento do Swagger 2 que descreva toda a REST API emitindo um HTTP GET para a URL a seguir:

```
https://host:port/ibm/api/docs
```

Esse documento pode ser usado para aplicativos nos quais você deseja navegar programaticamente pelas APIs disponíveis.

host

Especifica o nome do host ou endereço IP no qual a REST API está disponível.

O valor padrão é localhost.

port

Especifica o número da porta HTTPS usada pela administrative REST API.

O valor padrão é 9443.

Se o nome do host ou número da porta mudar de padrão, será possível determinar os valores corretos na URL da REST API. Use o comando **dspmqweb status** para visualizar a URL.

Informações relacionadas

[Status dspmqweb \(exibir status do servidor mqweb\)](#)

Suporte ao idioma nacional da REST API

A REST API suporta, com determinadas qualificações, a capacidade para especificar idiomas nacionais como parte de uma solicitação de HTTP.

Plano de fundo

Cabeçalhos de HTTP permitem que um determinado comportamento seja especificado em solicitações e que informações adicionais sejam fornecidas em respostas.

Incluída nos cabeçalhos de HTTP está a capacidade de solicitar que as informações sejam retornadas em um idioma nacional. A REST API honra esse cabeçalho sempre que possível.

Especificando um idioma nacional

No cabeçalho de HTTP ACCEPT-LANGUAGE, uma ou mais tags de idioma podem ser fornecidas. É possível associar, opcionalmente, uma classificação com as tags, permitindo a especificação de uma lista ordenada por preferência. [Esta página tem uma discussão útil sobre o princípio.](#)

A REST API honra esse cabeçalho, selecionando um idioma do cabeçalho ACCEPT-LANGUAGE e retornando as mensagens nesse idioma. Quando o cabeçalho ACCEPT-LANGUAGE não contiver nenhum idioma que a REST API possa suportar, as mensagens serão retornadas em um idioma padrão. Esse idioma padrão corresponde ao código padrão de idioma do servidor da web da REST API.

A seção [“Quais dados são traduzidos?”](#) na página 89 explica quais dados são traduzidos.

Indicando o idioma aplicável em respostas

O cabeçalho de HTTP CONTENT-LANGUAGE em respostas da REST API indica o idioma no qual as mensagens são retornadas.

Quais dados são traduzidos?

Erro e mensagens informativas são traduzidas, outro texto não é.

- Os dados que são retornados de um gerenciador de filas não são traduzidos - por exemplo, no caso de executar um comando do MQSC por meio da REST API, as respostas do gerenciador de filas estarão no código de idioma do gerenciador de filas.
- A documentação gerada (Swagger) para a REST API, conforme exposta por meio do recurso `apiDiscovery`, está em inglês.

Quais idiomas são suportados?

Além do inglês, as mensagens de erro e informativas da REST API são traduzidas para os idiomas a seguir.

Chinês (Simplificado)

Denotado pela tag de idioma zh_CN

Chinês (Tradicional)

Denotado pela tag de idioma zh_TW

Tcheco

Denotado pela tag de idioma `cs`

Francês

Denotado pela tag de idioma `fr`

Húngaro

Denotado pela tag de idioma `hu`

Italiano

Denotado pela tag de idioma `it`

Japonês

Denotado pela tag de idioma `ja`

Coreano

Denotado pela tag de idioma `ko`

Polonês

Denotado pela tag de idioma `pl`

Português (brasileiro)

Denotado pela tag de idioma `pt_BR`

Russo

Denotado pela tag de idioma `ru`

Espanhol

Denotado pela tag de idioma `es`

Exemplos

Nos exemplos, o servidor da web tem um código de idioma padrão em inglês.

Especificando um único idioma suportado

Nos cabeçalhos da solicitação, `ACCEPT-LANGUAGE` é configurado como `fr`. Esta configuração especifica que francês é o idioma preferencial para texto traduzível.

Nos cabeçalhos de resposta, `CONTENT-LANGUAGE` é configurado como `fr`. Esta configuração indica que um erro e mensagens informativas na resposta estão em francês.

Especificando uma lista de idiomas

Nos cabeçalhos da solicitação, `ACCEPT-LANGUAGE` é configurado como `am, fr`. Esta configuração especifica que amárico e francês são idiomas aceitáveis para texto traduzível e que amárico é o idioma preferencial para texto traduzível.

Nos cabeçalhos de resposta, `CONTENT-LANGUAGE` é configurado como `fr`. Essa configuração indica que as mensagens de erro e informativas na resposta estão em francês, visto que a REST API não suporta amárico.

Especificando um único idioma não suportado

Nos cabeçalhos da solicitação, `ACCEPT-LANGUAGE` é configurado como `am`. Esta configuração especifica que amárico é o idioma preferencial para texto traduzível.

Nos cabeçalhos de resposta, `CONTENT-LANGUAGE` é configurado como `en`. Essa configuração indica que as mensagens de erro e informativas na resposta estão em inglês, visto que a REST API não suporta amárico.

Versões do REST API

O número da versão da REST API é parte da URL base para solicitações REST. Por exemplo, `https://localhost:9443/ibmmq/rest/v2/admin/installation`. O número da versão é usado para isolar clientes de mudanças na REST API que podem ser introduzidas em liberações futuras.

O IBM MQ 9.2.0 apresenta a versão 2 do REST API. Este aumento de versão aplica-se ao administrative REST API, messaging REST API e MFT REST API. Esse aumento de versão muda a URL de recurso que é usada para a REST API. O prefixo da URL para as URLs de recursos na versão 2 é a URL a seguir:

```
https://host:port/ibmmq/rest/v2/
```

Stabilized Algumas mudanças que são introduzidas na REST API podem mudar a função existente da REST API de modo que os clientes que usam a REST API podem precisar ser atualizados. Para evitar que tais mudanças forcem os clientes a serem atualizados, o número da versão da REST API é aumentado e a função existente estabilizada no número anterior. A nova função que pode mudar a função existente é incluída na REST API no novo número da versão. Portanto, os clientes podem continuar a usar a REST API na versão anterior sem serem atualizados.

As mudanças da REST API que podem resultar na exigência de uma atualização do cliente incluem as mudanças a seguir:

- Remoção de suporte para um atributo existente no JSON que é enviado para, ou retornado de, a REST API.
- Remoção de uma URL, verbo HTTP ou cabeçalho. Por exemplo, se uma URL ou um cabeçalho for renomeado ou se um verbo diferente for usado.
- Adição de um novo atributo JSON obrigatório para dados que são enviados a uma URL existente.
- Adição de um novo cabeçalho de HTTP obrigatório para dados que são enviados a uma URL existente.
- Adição de um novo parâmetro de consulta obrigatório a uma URL existente.

Quando esse tipo de mudança é introduzido na função REST API que existia em uma liberação Long Term Support (LTS), o número da versão da REST API é aumentado para a primeira dessas mudanças. Quaisquer mudanças subsequentes que forem feitas em uma liberação Continuous Delivery (CD) que possam requerer mudanças em clientes que usam a REST API usarão o novo número da versão.

Esse número da versão permanece o mesmo em liberações CD subsequentes até a próxima liberação LTS. Portanto, o número da versão aumenta no máximo uma vez entre as liberações LTS.

Stabilized Quando o número da versão é aumentado, a função REST API existente é estabilizada no número da versão antiga. Ou seja, a função REST API existente que estava disponível na liberação LTS permanece disponível no número da versão antiga, mas nenhuma mudança adicional é feita para essa versão. Qualquer nova função que for incluída na REST API será incluída na nova versão da REST API. No entanto, não há garantia de que quaisquer adições feitas na REST API em liberações CD antes do aumento da versão serão incluídas na versão mais antiga da REST API.

Deprecated Os clientes existentes podem continuar a usar a REST API no número da versão antiga sem precisar de quaisquer mudanças. As versões mais antigas da REST API podem ser descontinuadas e, eventualmente, removidas.

Algumas mudanças não requerem mudanças em clientes que usam a REST API. Essas mudanças não resultam em um aumento do número da versão. Portanto, assegure-se de que qualquer cliente que use a REST API não precise ser atualizado quando esses tipos de mudanças forem introduzidos. Essas mudanças na REST API podem incluir as mudanças a seguir:

- Adição de um novo atributo JSON a dados existentes que são retornados da REST API.
- Adição de uma nova URL.
- Adição de um novo verbo HTTP a uma URL existente.
- Adição um novo código de status a uma URL existente.
- Adição de novos atributos JSON opcionais para dados que são enviados a uma URL existente.
- Adição de novos parâmetros de consulta a uma URL existente.
- Adição de novos cabeçalhos para dados que são enviados a uma URL existente.
- Retorno de cabeçalhos novos do REST API.

Mudanças na nova função da API REST do Continuous Delivery

Para a nova função REST API que é incluída em uma liberação CD, quaisquer mudanças que são feitas nessa nova função que podem requerer mudanças em clientes da REST API não aumentam o número da versão. Ou seja, a nova função pode mudar antes da próxima liberação LTS sem aumentar o número da versão. Quando a função é incluída em uma liberação LTS, quaisquer mudanças subsequentes que podem requerer mudanças nos clientes da REST API aumentam o número da versão.

exemplo

1. Na liberação LTS X, a REST API está na versão 1.
2. Na liberação CD X.0.1, o suporte para uma nova URL é incluído. Essa mudança não requer mudanças em clientes que usam a REST API. Portanto, o REST API permanece na versão 1.
3. No CD X.0.2, o suporte para uma nova URL é incluído. Essa mudança não requer mudanças em clientes que usam a API de REST. Portanto, o REST API permanece na versão 1.
4. Na liberação LTS Y, a REST API está na versão 1.
5. Na liberação CD Y.0.1, uma URL existente é renomeada. Essa mudança pode requerer mudanças em clientes que usam a REST API. Portanto, uma nova versão do REST API é criada como versão 2. A URL renomeada é incluída na versão 2 do REST API, juntamente com toda a função existente. Qualquer nova função que é incluída no REST API é incluída na versão 2. A versão 1 permanece estabilizada no nível na liberação Y do LTS.
6. Na liberação CD Y.0.2, outra URL existente é renomeada. Como a versão já está aumentada na liberação Y do CD, o REST API permanece na versão 2. A versão 1 permanece estabilizada no nível na liberação Y do LTS.
7. Na liberação Z do LTS, o REST API permanece na versão 2. A versão 1 permanece estabilizada no nível na liberação Y do LTS.

Administração usando o IBM MQ Console

É possível executar tarefas básicas de administração usando o IBM MQ Console

Nota: Não desative o servidor de comandos em nenhum dos gerenciadores de filas quando usar o IBM MQ Console. Se o servidor de comandos estiver desativado para um gerenciador de filas:

- O IBM MQ Console se torna não responsivo, com longos atrasos no processamento de comandos
- Quaisquer comandos que são emitidos para o gerenciador de fila de tempo limite.

Tarefas relacionadas



[Rastreamento do IBM MQ Console](#)

Introdução ao IBM MQ Console

Configure o servidor mqweb; determine o URI para o IBM MQ Console; conecte-se ao console; efetue login no console

Antes de começar

Para concluir essa tarefa, deve-se ser um usuário com determinados privilégios para que seja possível usar o comando **dspmweb**:

-  No z/OS, deve-se ter autoridade para executar o comando **dspmweb** e acesso de gravação ao arquivo mqwebuser.xml.
-  Em todos os outros sistemas operacionais, deve-se ser um usuário privilegiado.

 Em IBM i, os comandos devem estar em execução em QSHLL

Sobre esta tarefa

Esteja ciente sobre as seguintes restrições:

- ▶ **z/OS**
 - Os gerenciadores de filas no z/OS não podem ser criados, excluídos, iniciados ou interrompidos.
 - Os inicializadores de canais no z/OS não podem ser iniciados ou interrompidos, e o status do inicializador de canais não é exibido.
 - Os listeners não podem ser exibidos ou administrados.
 - Iniciar, executar ping, resolver e reconfigurar comandos de canal só podem ser emitidos com CHLDISP(DEFAULT).
 - Os objetos definidos com QSGDISP(GROUP) não podem ser exibidos nem gerenciados.
 - Não é possível gerenciar a segurança do gerenciador de filas.
 - O uso de recursos do sistema não pode ser monitorado.
- ▶ **Multi**
 - Não é possível usar o IBM MQ Console para trabalhar com canais AMQP.
 - Não é possível usar o IBM MQ Console para trabalhar com canais MQTT.

Procedimento

1. Se o servidor mqweb ainda não estiver configurado para uso pelo IBM MQ Console, configure o servidor mqweb.

Para obter mais informações sobre como configurar o servidor mqweb com um registro básico, consulte [Configuração básica para o servidor mqweb](#)

2. ▶ **z/OS**

Em z/OS, configure a variável de ambiente WLP_USER_DIR para que seja possível usar o comando **dspmweb**. Configure a variável para apontar para a configuração do servidor mqweb inserindo o comando a seguir:

```
export WLP_USER_DIR=WLP_user_directory
```

em que *WLP_user_directory* é o nome do diretório transmitido para crtmqweb. Por exemplo:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Para obter mais informações, consulte [Criando o servidor mqweb](#).

3. Determine o URI para o IBM MQ Console inserindo o comando a seguir:

```
dspmweb status
```

O comando gera uma saída semelhante à seguinte:

```
MQWB1124I: Server 'mqweb' is running.  
URLS:  
https://localhost:9443/ibmmq/rest/v1/  
https://localhost:9443/ibmmq/console/
```

O URI para o IBM MQ Console termina com o sufixo console/.

4. Conecte-se ao IBM MQ Console inserindo a URL da etapa anterior em um navegador.

Uma exceção de segurança pode ser produzida pelo navegador porque o certificado padrão que é fornecido com o servidor mqweb não é um certificado confiável. Escolha para continuar com o IBM MQ Console.

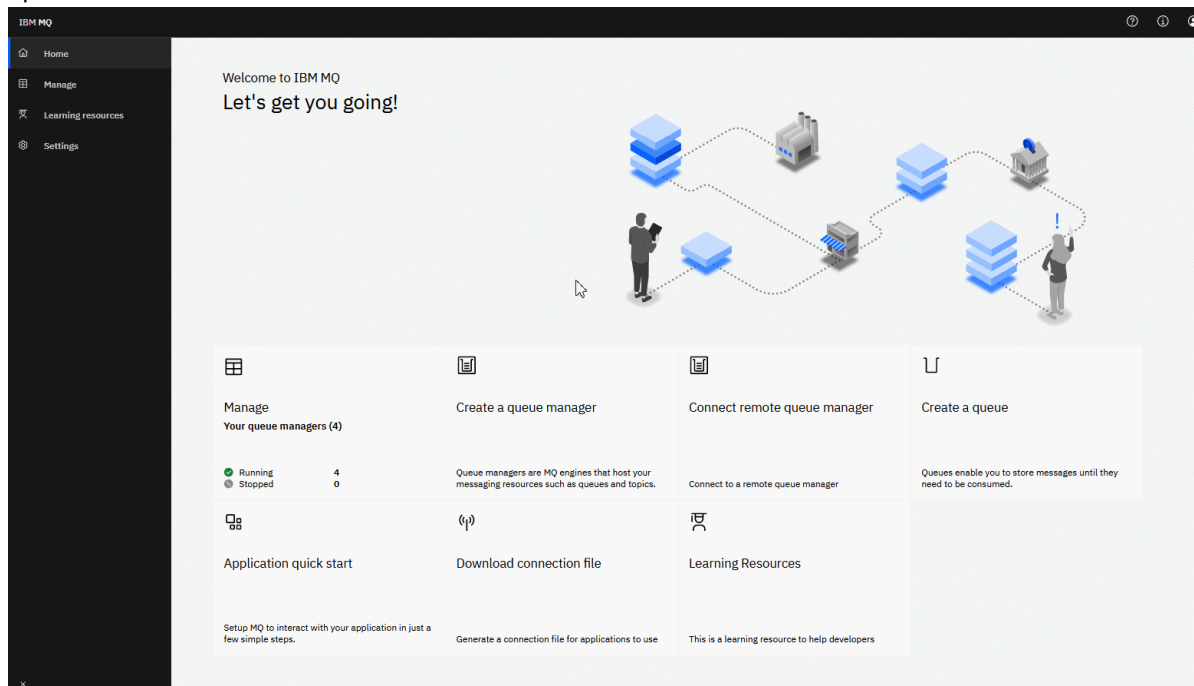
5. Efetue login no IBM MQ Console. Use o nome do usuário mqadmin e a senha mqadmin.

Como proceder a seguir

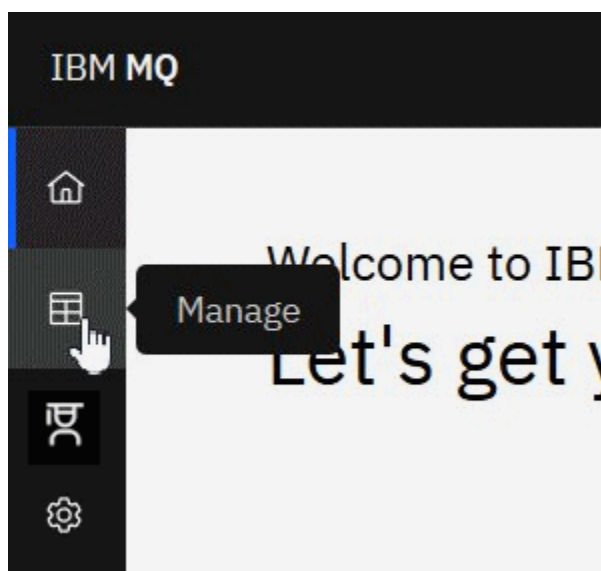
Por padrão, o IBM MQ Console usa a autenticação baseada em token para autenticar usuários. Também é possível usar a autenticação por certificado de cliente. Para obter mais informações, veja [Usando a autenticação por certificado de cliente com a REST API e o IBM MQ Console](#).

V 9.4.0 Tour rápido do IBM MQ Console

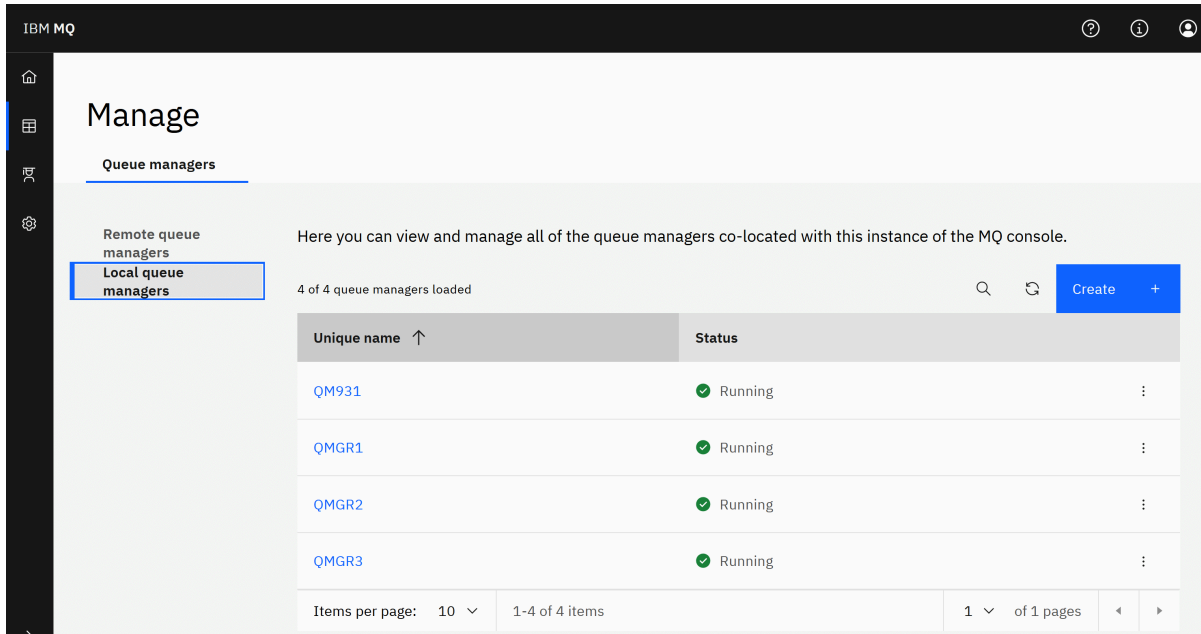
Ao efetuar login pela primeira vez no IBM MQ Console, você é levado para a página de entrada. Aqui, é possível escolher gerenciar os gerenciadores de filas existentes, criar um gerenciador de filas ou uma fila, navegar para alguns tópicos de educação ou abrir as informações do produto IBM MQ no IBM Documentation. Também é possível ativar a iniciação rápida do aplicativo, que o orienta durante o processo de configuração rápida e fácil do sistema de mensagens entre gerenciadores de filas e aplicativos novos ou existentes.



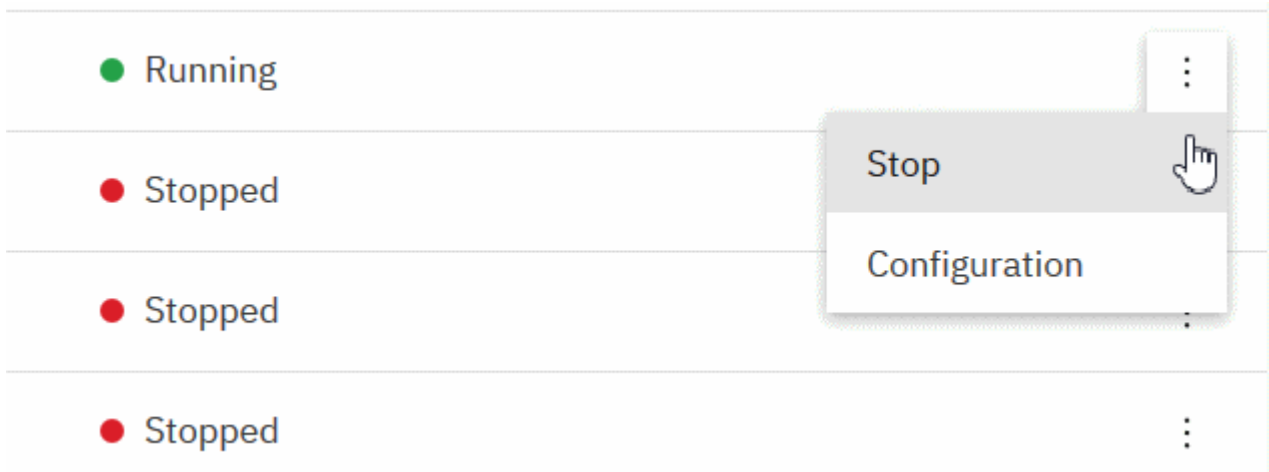
Como alternativa, basta clicar no ícone Gerenciar para começar a gerenciar os objetos do IBM MQ imediatamente.



A visualização de gerenciamento mostra inicialmente os gerenciadores de filas e os seus estados atuais. Também é possível criar novos gerenciadores e conectar gerenciadores de filas remotas.

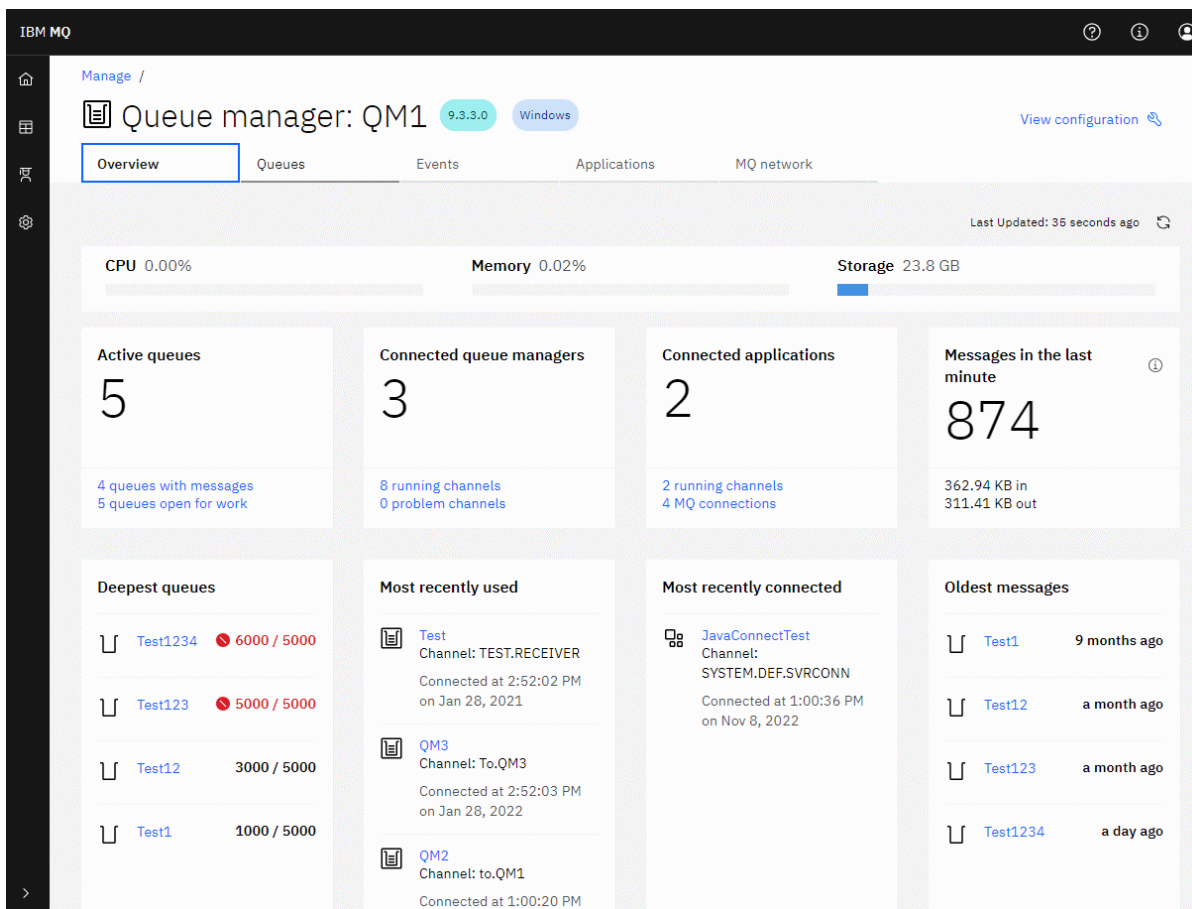


Cada gerenciador de filas tem um menu que permite parar ou configurar um gerenciador de filas em execução ou iniciar ou excluir um gerenciador de filas interrompido.



Registros de autoridade, objetos de Informações sobre autenticação e registros de Autenticação de canal do gerenciador de filas podem ser localizados na guia **Segurança** da página **Configuração** do gerenciador de filas, na qual é possível criar e incluir novos.

Clique no nome de um gerenciador de filas em execução para abrir o seu painel.



No painel do gerenciador de filas, é possível concluir as ações a seguir:

V 9.4.0 Na guia **Visão geral**, visualize as informações a seguir:

CPU

Estimativa de porcentagem de uso de CPU pelo gerenciador de fila (Não aplicável em z/OS.)

Memory

Estimativa de porcentagem de uso de memória pelo gerenciador de filas (Não aplicável em z/OS ou Windows.)

Armazenamento

Estimativa percentual do espaço livre do disco no qual o gerenciador de filas reside. (Não aplicável em z/OS.)

Filas ativas

Contagem de filas que possuem mensagens ou que estão abertas para entrada ou saída

Gerenciadores de filas conectados

Contagem de gerenciadores de filas conectados atualmente, como derivados de canais ativos

Aplicativos conectados

Contagem de aplicativos conectados no momento

Mensagens no último minuto

Exibe um resumo dos tópicos do sistema PUT/GET que mostram o rendimento da mensagem a cada 10 segundos. (Não aplicável em z/OS.)

Assinaturas

Exibe uma contagem de assinaturas. Visível somente no z/OS e em outras plataformas nas quais o monitoramento de tópicos do sistema é inibido (consulte [setmqweb properties](#)).

Filas mais profundas

Lista as filas em ordem de profundidade Mostra a profundidade da fila atual e a profundidade máxima da fila

Usados mais recentemente

Lista os gerenciadores de filas conectados atualmente, ordenados por data da última mensagem

Mais recentemente conectado

Lista os aplicativos atualmente conectados como derivados dos canais de conexão do servidor ativos, ordenados por data e hora de início do canal

Mensagens mais antigas

Lista as filas ordenados pela data e hora da mensagem mais antiga

As informações exibidas na guia **Visão Geral** são derivadas do monitoramento de tópicos do sistema (consulte [Métricas publicadas nos tópicos do sistema](#))... O z/OS não suporta o monitoramento de tópico do sistema e o monitoramento para propósitos de exibição do console pode ser desativado em outras plataformas (consulte [propriedades setmqweb](#)). Nesses casos, a guia **Visão Geral** exibe informações mais limitadas e sua aparência é semelhante ao exemplo a seguir:

IBM MQ

Manage / Queue manager: remote-MQ53 [View configuration](#)

Overview Queues Events Applications MQ network

Last Updated: a few seconds ago


Active queues 0 0 queues with messages 0 queues open for work	Connected queue managers 0 0 running channels 0 problem channels	Connected applications 0 6 running channels 0 MQ connections	Subscriptions 0
Deepest queues No queues with messages found	Most recently used No connected queue managers After you connect queue managers, their connections will display here.	Most recently connected No connected applications After you connect applications, their connections will display here.	Oldest messages No queues with messages found

Na guia **Filas**:

- Criar novas filas
- Clique em um nome de fila para visualizar mensagens existentes e criar novas e configurar a fila.

Na guia **Eventos** :

tópicos

- Criar novos tópicos
- Configurar tópicos existentes 
- Clique em um nome de tópico para visualizar as assinaturas correspondentes

Assinaturas

- Criar novas assinaturas gerenciadas ou não gerenciadas

- Configurar assinaturas existentes



V 9.4.0 Na guia **Aplicativos** :

Visão Geral

Contém ladrilhos que fornecem visões gerais das seguintes estatísticas:

Aplicativos conectados

Exibe uma contagem do número de aplicativos conectados.. Fornece links para as seguintes guias:

- **Instâncias do aplicativo**
- **Conexões**

Executando instâncias do canal

Exibe uma contagem do número de instâncias do canal SVRCONN e desses links para aquelas que são definidas ou interrompidas na guia **Canal do aplicativo** .

Conexões

Exibe uma contagem do número de conexões.. Fornece links para as informações a seguir na guia **Conexões** :

- Conexões locais (aquelas sem um nome de canal).
- Conexões remotas (aquelas com um nome de canal)

Aplicativos mais comuns

Exibe uma lista de aplicativos frequentes, ordenados pelo número de conexões que eles estão usando

Canais mais comuns

Exibe uma lista de canais frequentes, ordenados pelo número de instâncias ativas.

Transações mais antigas

Exibe uma lista de transações mais antigas por nome do aplicativo Essas transações têm conexões com unidades de trabalho abertas e são ordenadas pela data e hora de início da UOW

Versões conectadas remotas

Exibe uma lista de versões do IBM MQ conectadas comuns, ou seja as instâncias do canal que possuem um REMOTE_VERSION especificado....

Segurança do canal de aplicativo

Exibe uma lista de protocolos de segurança de canal conectados comuns, ou seja aquelas instâncias de canal que possuem um SECURITY_PROTOCOL especificado.

Taxas de transferência de canal

Exibe uma lista de canais comuns ordenados pelas taxas de transferência de mensagens e bytes Usa a data e hora de início do canal para calcular a duração e usa MSGS e MQIACH_BYTES_SENT/ MQIACH_BYTES_RCVD para calcular a taxa.

Aplicativos

Visualizar informações sobre aplicativos que estão conectados ao gerenciador de filas.

Canais

Visualizar atividade em canais conectados a aplicativos.

Canais do aplicativo

- Iniciar, parar, executar ping e configurar canais
- Criar novos canais
- Reconfigurar canais

Instâncias do canal do aplicativo

- Visualizar status de instâncias do canal de aplicativo

- Resolver mensagens em dúvida nos canais

V 9.4.0

Na guia **Rede doMQ** :

Visão Geral

Contém ladrilhos que fornecem visões gerais das seguintes estatísticas:

Executando instâncias do canal de gerenciador de filas

Exibe uma contagem do número de instâncias do canal não SVRCONN. Exibe links para os seguintes tipos de instâncias de canal na guia **Gerenciadores de filas conectados** :

- Canais definidos
- Canais interrompidos

Gerenciadores de filas conectados

Exibe uma contagem de gerenciadores de filas conectados por MQCA_REMOTE_Q_MGR_NAME Também fornece uma contagem de gerenciadores de filas retornados por MQCMD_INQUIRE_CLUSTER_Q_MGR.

Associação de cluster

Se houver apenas um cluster do gerenciador de filas, exibirá o nome do cluster e se o gerenciador de fila é um repositório completo ou parcial Exibe quantos gerenciadores de filas estão visíveis no cluster Se houver mais de um cluster, então será exibido o número de clusters mais uma contagem de gerenciadores de filas de repositório completos e parciais em cada cluster

Falha nos canais do gerenciador de filas

Exibe uma lista de canais que estão em um estado de repetição (não pausado / em execução). Calcula o número de novas tentativas restantes se estiverem no estado de nova tentativa A lista contém canais com os seguintes tipos de status:

- MQCHS_PAUSED
- MQCHS_RETENTANDO

Maiores atrasos de mensagens

Exibe uma lista de canais que possuem um indicador de tempo de XMIT (longo período)

Filas de transmissão não assistidas

Exibe uma lista de filas de transmissão que possuem uma profundidade de fila diferente de zero e nenhuma manipulação associada.

Versões de conexão remota

Exibe uma lista de versões do IBM MQ conectadas comuns, ou seja as instâncias do canal que possuem um REMOTE_VERSION especificado....

Segurança do canal de gerenciador de filas

Exibe uma lista de protocolos de segurança de canal conectados comuns, ou seja aquelas instâncias de canal que possuem um SECURITY_PROTOCOL especificado.

Funcionamento do cluster

Exibe várias estatísticas independentes relacionadas ao funcionamento do cluster. O status inclui:

- O número de objetos do cluster (filas, tópicos e gerenciadores de filas).
- O número de gerenciadores de filas suspensos (MQIACF_SUSPEND configurado como YES)
- A profundidade de SYSTEM do SYSTEM.CLUSTER.COMMAND.QUEUE fila.
- O número de entradas do gerenciador de filas do cluster começando com SYSTEM.TEMP.

Se todos esses valores forem zero, esse ladrilho não será exibido e o ladrilho **Listeners** será mostrado no lugar.


Listeners

Exibe uma lista de listeners e se eles estão em um estado de execução Mostrado apenas se o bloco **Funcionamento do cluster** não for exibido.


Gerenciadores de filas conectados

Visualize os detalhes dos gerenciadores de filas atualmente conectados a este gerenciador de filas

Canais do gerenciador de filas

- Iniciar, parar, executar ping e configurar canais 
- Criar novos canais
- Reconfigurar canais

Instâncias do canal gerenciador de filas


- Visualizar status de instâncias do canal do gerenciador de filas
- Resolver mensagens em dúvida nos canais 


V 9.4.0 IBM MQ Console: Trabalhando com Gerenciadores de Filas Locais

Você cria, configura e controla os gerenciadores de filas locais por meio do nível superior da visualização



Sobre esta tarefa

 Multi A visualização Gerenciar lista os gerenciadores de filas locais que são incluídos na instalação do IBM MQ por meio da qual o IBM MQ Console está em execução. Os gerenciadores de filas associados a diferentes instalações do IBM MQ no mesmo sistema não são listados.

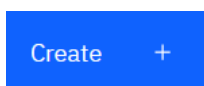
 z/OS No z/OS, a visualização Gerenciar lista os gerenciadores de filas que estão na mesma versão que o IBM MQ Console, e são definidos no sistema em que o IBM MQ Console está executando. Os gerenciadores de filas em uma versão diferente para o IBM MQ Console não são listados.

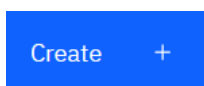

É possível selecionar gerenciadores de filas individuais na lista, com os quais trabalhar.





Nota: O IBM MQ Console pode se conectar a um gerenciador de filas do RDQM local quando ele está ativo (ou seja, tem a função primária), mas não oferece nenhum recurso específico do RDQM

Procedimento

- Para criar um novo gerenciador de filas locais:



- Clique no botão criar  na visualização da lista do gerenciador de filas.
 - Insira um nome para o novo gerenciador de filas. O nome pode conter até 48 caracteres. Os caracteres válidos são letras e números e os caracteres ".", "/", "_" e "%".
 - Opcional: Insira uma porta TCP/IP disponível na qual o gerenciador de filas possa atender. O número da porta não deve exceder 65535.
 - Clique em **Criar**. O novo gerenciador de filas é criado e iniciado.
- Para iniciar um gerenciador de filas locais:
 - Localize na lista o gerenciador de filas que você deseja iniciar.
 - Selecione **Iniciar** no menu .
 - Para parar um gerenciador de filas locais:
 - No widget de gerenciador de filas locais, selecione o gerenciador de filas que você deseja parar na lista.

- b) Selecione **Parar** no menu  .
- Para excluir um gerenciador de filas locais:
 - a) Se o gerenciador de filas estiver em execução, pare-o.
 - b) Selecione **Visualizar configuração** no menu  e selecione **Excluir gerenciador de filas**.
 - c) Confirme se você deseja excluir o gerenciador de filas inserindo o nome dele na janela de confirmação. O gerenciador de filas e todos os objetos associados são excluídos.
- Para visualizar e editar as propriedades de um gerenciador de filas locais:
 - a) Assegure-se de que o gerenciador de filas esteja em execução e localize-o na lista do gerenciador de filas.
 - b) Selecione **Visualizar configuração** no menu  .
 - c) Assegure-se de que a guia **Propriedades** esteja selecionada. Visualize as propriedades e edite-as, conforme necessário. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos. Para obter informações sobre uma propriedade, é possível visualizar as informações da propriedade em [Propriedades do gerenciador de filas](#).
- Para trabalhar com configurações de segurança para o gerenciador de filas locais:
 - a) Assegure-se de que o gerenciador de filas esteja em execução e selecione-o na lista do gerenciador de filas.
 - b) Selecione **Visualizar configuração** no menu  .
 - c) Assegure-se de que a guia **Segurança** esteja selecionada.
 - d) É possível trabalhar com objetos de autenticação, registros de autorização ou objetos de autenticação de canal. Visite os tópicos a seguir para obter mais informações:
 - [“IBM MQ Console: Trabalhando com objetos de informações sobre autenticação”](#) na página 101
 - [“IBM MQ Console: Trabalhando com Registros de Autoridade do Gerenciador de Filas”](#) na página 103
 - [“IBM MQ Console: Trabalhando com Registros de Autenticação de Canal”](#) na página 104

V 9.4.0 IBM MQ Console: Trabalhando com objetos de informações sobre autenticação

É possível usar o console para incluir e excluir objetos de informações de autenticação em um gerenciador de filas. Também é possível visualizar e configurar as propriedades, além de gerenciar registros de autoridade para os objetos.


Sobre esta tarefa

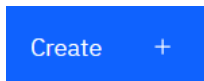
A visualização de informações de autenticação lista as informações de autenticação que existem para um gerenciador de filas específico. É possível selecionar informações sobre autenticação individuais na lista, com as quais trabalhar.

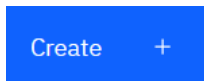
As informações sobre autenticação do gerenciador de filas formam parte do suporte do IBM MQ para Segurança da Camada de Transporte (TLS). Esses objetos contêm as definições necessárias para executar a verificação de revogação de certificados, usando OCSP ou as Listas de Revogação de Certificado (CRLs) em servidores LDAP, e as definições necessárias para permitir a verificação do ID de usuário e da senha.


Procedimento

- Para visualizar as informações sobre autenticação para um gerenciador de filas:


- a) Assegure-se de que o gerenciador de filas esteja em execução e selecione-o na lista do gerenciador de filas.
 - b) Selecione **Visualizar configuração** no menu 
 - c) Assegure-se de que a guia **Segurança** esteja selecionada.
 - d) Selecione **Informações sobre autenticação** no painel de navegação.
- Para incluir um objeto de informações sobre autenticação:

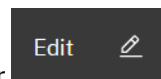


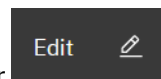
- a) Clique no botão Criar  na visualização da lista de informações de autenticação.
 - b) Especifique o nome do objeto de informações sobre autenticação. Os caracteres válidos são letras e números e os caracteres ".", "/", "_", e "%".
 - c) Especifique o tipo de objeto de informações sobre autenticação.
 - d) Especifique informações adicionais apropriadas para o tipo de objeto:
 - Para **CRL LDAP**, especifique o **nome do servidor LDAP**. Esse nome é o nome do host, o endereço decimal pontuado IPv4 ou a notação hexadecimal IPv6 do host no qual o servidor LDAP está em execução, com um número de porta opcional. Opcionalmente, é possível especificar um nome de usuário e uma senha para o usuário acessar o servidor LDAP.
 - Para **OCSP**, especifique a **URL do respondente do OCSP**. Essa é a URL do respondente que é usada para verificar a revogação de certificado. Este valor deve ser uma URL HTTP contendo o nome do host e número da porta do respondente OCSP. Se o respondente OCSP estiver usando a porta 80, que é o padrão para HTTP, em seguida, o número da porta pode ser omitido. URLs HTTP são definidas no RFC 1738.
 - Para **IDPW OS**, não há requisitos adicionais embora seja possível, opcionalmente, especificar outras opções para este tipo de autenticação.
 - Para **IDPW LDAP**, especifique o **nome do servidor LDAP** e o nome **Curto do usuário**. O nome do servidor LDAP é o nome do host, o endereço decimal pontuado IPv4 ou a notação hexadecimal IPv6 do host no qual o servidor LDAP está em execução, com um número de porta opcional. O nome curto do usuário é o campo no registro do usuário LDAP usado como um nome abreviado para a conexão. Opcionalmente, é possível especificar mais opções para este tipo de autenticação.
 - e) Clique em **Incluir**.
- Para excluir um objeto de informações sobre autenticação:



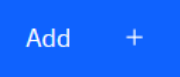
- a) Selecione o ícone de chave inglesa  para o objeto de informações sobre autenticação que você deseja excluir da lista.
- b) Na visualização de propriedades do objeto, clique em **Excluir objeto de informações sobre autenticação**.
- c) Confirme que você deseja excluir o objeto de informações sobre autenticação, clicando em **Excluir**. O objeto é excluído.

- Para visualizar e editar as propriedades de um objeto de informações sobre autenticação:

- a) Selecione o ícone de chave inglesa  para o objeto de informações sobre autenticação que você deseja visualizar da lista.



- b) Para editar as propriedades exibidas, clique no botão Editar 
- c) Edite as propriedades conforme necessárias. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos.

- d) Clique em **Salvar** para salvar as alterações.
- Para visualizar e editar os registros de autoridade de um objeto de informações sobre autenticação:
 - a) Selecione o ícone de chave inglesa  para o objeto de informações sobre autenticação para o qual deseja visualizar o registro de autoridade na lista.
 - b) Selecione a guia **Segurança**.
 - c) Para editar ou excluir um registro de autoridade existente, selecione **Editar** ou **Excluir** no menu .
 - d) Para incluir um novo registro de autoridade, clique no botão **Incluir** , forneça os detalhes do novo registro de autoridade e clique em **Criar**.


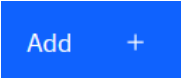
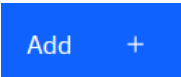
V9.4.0 **IBM MQ Console: Trabalhando com Registros de Autoridade do Gerenciador de Filas**

É possível controlar o acesso que os usuários e os grupos têm aos gerenciadores de filas especificando um registro de autoridade para esse usuário ou grupo.

Sobre esta tarefa

É possível ajustar o acesso que um usuário do sistema de mensagens ou grupo de usuários do sistema de mensagens tem a um gerenciador de filas específico usando registros de autoridade. Há dois tipos de registros de autoridade: os registros de **acesso do gerenciador de filas** que controlam autoridades gerais e a **permissão para criar** registros que controlam quais usuários e grupos podem criar objetos para o gerenciador de filas.

Procedimento

- Para visualizar os registros de autoridade para um gerenciador de filas:
 - a) Assegure-se de que o gerenciador de filas esteja em execução e selecione-o na lista do gerenciador de filas.
 - b) Selecione **Visualizar configuração** no menu .
 - c) Assegure-se de que a guia **Segurança** esteja selecionada.
 - d) Selecione **Registros de autoridade** no painel de navegação. A visualização mostra os registros de autoridade em duas áreas de janela, permitindo que você trabalhe com registros de autoridade gerais e com registros de autoridade de criação.
- Para incluir um registro de autoridade geral:
 - a) Clique no botão **Incluir**  na visualização de lista **Acesso do Gerenciador de filas**.
 - b) Escolha se você deseja incluir um registro de autoridade para um usuário ou grupo.
 - c) Especifique o nome do usuário ou grupo para o qual você está adicionando um registro de autoridade (o registro de autoridade usa isso como nome).
 - d) Selecione as autoridades que deseja conceder.
 - e) Clique em **Criar**.
- Para incluir um registro de autoridade de criação:
 - a) Clique no botão **Incluir**  na visualização de lista **Permissão para criar**.

- b) Escolha se você deseja incluir um registro de autoridade para um usuário ou grupo.
 - c) Especifique o nome do usuário ou grupo para o qual você está adicionando um registro de autoridade (o registro de autoridade usa isso como nome).
 - d) Selecione os tipos de objeto aos quais você está concedendo autoridade de criação.
 - e) Clique em **Criar**.
- Para excluir um registro de autoridade:
 - a) Selecione o registro de autoridade que você deseja excluir e selecione **Excluir**.
 - b) Confirme que você deseja excluir o objeto de informações sobre autenticação, clicando em **Excluir**. O objeto é excluído.
 - Para visualizar e editar as propriedades de um registro de autoridade:
 - a) Clique no registro de autoridade que você deseja visualizar
 - b) Mude as configurações conforme necessário e clique em **Salvar** para salvar as suas mudanças.

V 9.4.0 IBM MQ Console: Trabalhando com Registros de Autenticação de Canal


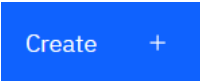
É possível usar o IBM MQ Console para incluir e excluir registros de autenticação de canal em um gerenciador de filas. Também é possível visualizar e configurar as propriedades de registros de autenticação de canal.

Sobre esta tarefa


Para exercer controle mais preciso sobre o acesso concedido à conexão de sistemas em um nível de canal, é possível usar registros de autenticação de canal.


Para aplicar segurança, é possível usar registros de autenticação de canal de bloqueio para bloquear o acesso a seus canais. Também é possível usar registros de autenticação de canal do mapa de endereços para permitir acesso aos usuários especificados. Para saber mais sobre registros de autenticação de canal, consulte [Registros de autenticação de canal](#).

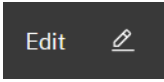
Procedimento

- Para visualizar as informações de autenticação de canal para um gerenciador de filas:
 - a) Assegure-se de que o gerenciador de filas esteja em execução e selecione-o na lista do gerenciador de filas.
 - b) Selecione **Visualizar configuração** no menu 
 - c) Assegure-se de que a guia **Segurança** esteja selecionada.
 - d) Selecione **Autenticação de canal** no painel de navegação.
- Para incluir um registro de autenticação de canal:
 - a) Clique no botão Criar  na visualização de lista de informações de autenticação de canal.
 - b) Escolha o tipo de regra que você deseja utilizar. Selecione um **Permitir**, **Bloquear** ou **Avisar**.
 - c) Escolha o tipo de identidade para a qual você está configurando uma regra de autenticação de canal. Diferentes tipos de identidade estão disponíveis, dependendo do tipo de regra que você selecionou.
 - d) Forneça as informações necessárias para a identidade que você está especificando. Por padrão, as propriedades mínimas recomendadas são exibidas para que você forneça valores para elas. É possível visualizar todas as propriedades disponíveis selecionando **Criação customizada**.
 - e) Clique em **Criar** para criar o registro da autenticação de canal

Para obter mais informações sobre as configurações disponíveis para registros de autenticação de canal, consulte [Registros de autenticação de canal](#) e [SET CHLAUTH](#)

- Para excluir um registro de autenticação de canal:
 - a) Clique no ícone de chave inglesa  ao lado do registro de autenticação de canal que você deseja excluir.
 - b) Na visualização Editar autenticação de canal, clique em **Excluir objeto de autenticação de canal**.
 - c) Confirme que você deseja excluir o registro de autenticação de canal, clicando em **Excluir**. O registro de autenticação de canal é excluído.
- Para visualizar e editar as propriedades de um registro de autenticação de canal:

- a) Clique no ícone de chave inglesa  ao lado do registro de autenticação de canal que deseja editar ou visualizar. As propriedades são exibidas.

- b) Clique no botão Editar 
- c) Edite as propriedades conforme necessárias. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos.
- d) Clique em **Salvar** para salvar as alterações.


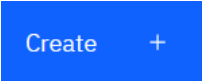

IBM MQ Console: Trabalhando com listeners



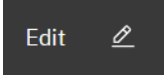


É possível usar o IBM MQ Console para incluir e excluir listeners, iniciar e parar listeners, visualizar e configurar propriedades do listener e gerenciar os registros de autoridade para um listener.

Sobre esta tarefa

A visualização de listeners exibe os listeners que existem para um gerenciador de filas específico. É possível selecionar listeners individuais com os quais trabalhar.

Procedimento

- Para visualizar os listeners de um gerenciador de filas:
 - a) Assegure-se de que o gerenciador de filas esteja em execução e selecione-o na lista do gerenciador de filas.
 - b) Selecione **Visualizar configuração** no menu 
 - c) Selecione a guia **Listeners**.
- Para criar um listener:
 - a) Clique no botão Criar 
 - b) Forneça as informações necessárias para o listener que você está criando.
 - c) Clique em **Criar**. O novo listener é criado.
- Para iniciar um listener:
 - a) Localize o listener que você deseja iniciar na lista.
 - b) Selecione **Iniciar** no menu 
- Para parar um listener:
 - a) Localize o listener que você deseja iniciar na lista.

- b) Selecione **Parar** no menu  .
- Para visualizar e editar as propriedades de um listener:
 - a) Localize o listener na lista.
 - b) Selecione **Visualizar configuração** no menu .
 - c) Assegure-se de que a guia **Propriedades** esteja selecionada. Para editar as propriedades, clique no botão Editar 
 - d) Edite as propriedades conforme necessárias. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos. Para obter mais informações sobre as propriedades, consulte [Propriedades do listener](#) na documentação do MQ Explorer.
 - e) Clique em **Salvar** para salvar as alterações.
- Para visualizar e editar os registros de autoridade de um listener:
 - a) Localize o listener na lista.
 - b) Selecione **Visualizar configuração** no menu .
 - c) Clique na guia **Segurança**.
 - d) Trabalhe com os registros de autoridade conforme descrito para os registros de autoridade do gerenciador de filas. Consulte [“IBM MQ Console: Trabalhando com Registros de Autoridade do Gerenciador de Filas”](#) na página 103.
- Para excluir um listener:
 - a) Localize o listener na lista.
 - b) Selecione **Visualizar configuração** no menu .
 - c) Clique em **Excluir listener**.

IBM MQ Console: Incluindo um gerenciador de filas remotas

É possível usar o IBM MQ Console para administrar um gerenciador de filas em execução em um sistema remoto.

Antes de começar

- Deve-se preparar o gerenciador de fila no sistema remoto para que ele possa ser administrado remotamente, consulte a etapa [“1” na página 108](#), [“2” na página 109](#), [“3” na página 109](#) e [“4” na página 109](#) de [“Incluindo um gerenciador de filas remotas no IBM MQ Console usando a linha de comandos”](#) na página 108
- Você também deve ativar conexões remotas a partir do IBM MQ Console. Para obter mais informações, consulte [Configurando o comportamento da conexão do gerenciador de filas remotas](#).

Sobre esta tarefa

Você usa uma tabela de definição de conexão do cliente (CCDT) em formato JSON para especificar os detalhes da conexão remota. É possível criar um JSON CCDT usando um editor de texto (consulte a etapa [“5” na página 109](#) de [“Incluindo um gerenciador de filas remotas no IBM MQ Console usando a linha de comandos”](#) na página 108) ou criar um usando o IBM MQ Console.

Alternativamente, é possível criar o CCDT por meio do IBM MQ Console especificando detalhes da conexão diretamente à medida que o gerenciador de filas remotas é incluído.

Também é possível conectar um gerenciador de filas remotas ao IBM MQ Console usando a linha de comandos para todas as tarefas necessárias (além de preparar o gerenciador de filas remotas e criar um CCDT). Consulte [“Incluindo um gerenciador de filas remotas no IBM MQ Console usando a linha de comandos”](#) na página 108.



Atenção: se você receber as seguintes mensagens:

```
MQWB2026E: The request to connect to the remote queue manager 'rqmgr-qmgr_name' failed
with the error message:
'JMSSC0051: The property 'JMS_IBM_MQMD_AccountingToken' should be set using type '[B',
not 'java.lang.Object'.'
```

você está tentando transmitir um `java.lang.Object` para o token de contabilidade, quando um tipo de objeto java byte [] é esperado

Procedimento

- Para incluir um gerenciador de filas remotas, especifique um CCDT existente:
 - a) Na página inicial, clique em **Conectar gerenciador de filas remotas**.
 - b) Especifique o nome do gerenciador de filas remotas.
 - c) Opcionalmente, especifique um nome exclusivo para o gerenciador de filas. Se você não especificar um nome exclusivo, o nome real será usado com o prefixo "remote-" incluído.
 - d) Certifique-se de que **Conectar usando um JSON CCDT** esteja selecionado.
 - e) Clique em **Procurar** e selecione o arquivo contendo o JSON CCDT que você deseja utilizar.
 - f) Clique em **Avançar** para se mover para a página do usuário e opcionalmente especificar um nome de usuário e senha para conectar ao gerenciador de filas remotas. Se você não especificar essas informações, as informações sobre autenticação serão obtidas do arquivo de configuração de conexão remota.
 - g) Clique em **Avançar** para se mover para a página Certificado. Se o seu CCDT especificar informações de "transmissionSecurity", estas informações serão usadas. Opcionalmente, é possível colar um certificado (como uma chave pública codificada em Base64) e incluí-lo no armazenamento confiável global.

O certificado é armazenado temporariamente em `WLP_USER_DIR/generated.crts/uniqueName-qmgrName.crt` antes de ser incluído no armazenamento de confiança. Quando a conexão é incluída com sucesso, o certificado é excluído deste local.
 - h) Clique em **Avançar** para visualizar a página de resumo. É possível usar o botão **Voltar** para visitar páginas anteriores e fazer correções. Se você estiver feliz com as informações, clique em **Conectar** para se conectar ao gerenciador de filas remotas.
- Para incluir um gerenciador de filas remotas e especificar as informações de conexão manualmente:
 - a) Na página inicial, clique em **Conectar gerenciador de filas remotas**.
 - b) Especifique o nome do gerenciador de filas remotas.
 - c) Opcionalmente, especifique um nome exclusivo para o gerenciador de filas. Se você não especificar um nome exclusivo, o nome real será usado com o prefixo "remote-" incluído.
 - d) Selecione **Entrada manual**.
 - e) Insira o nome do canal de conexão do cliente que a conexão utilizará.
 - f) Especifique o nome do host no qual o gerenciador de filas remotas está em execução. Se instalações remotas do MQ forem detectadas, os nomes do host serão exibidos e você poderá selecionar o host do gerenciador de filas remotas com o qual você deseja se conectar. Em algumas configurações de rede, não é possível detectar instâncias remotas do MQ. Nesse caso, inclua o nome do host e a porta manualmente.
 - g) Clique em **Avançar** para se mover para a página do usuário e opcionalmente especificar um nome de usuário e senha para conectar ao gerenciador de filas remotas. Se você não especificar essas informações, as informações sobre autenticação serão obtidas do arquivo de configuração de conexão remota.

- h) Clique em **Avançar** para se mover para a página Certificado. É possível selecionar um SSL CipherSpec na lista suspensa. Opcionalmente, é possível colar um certificado (como uma chave pública codificada em Base64) e inclui-lo no armazenamento confiável global.

O certificado é armazenado temporariamente em `WLP_USER_DIR/generated.crts/uniqueName-qmgrName.crt` antes de ser incluído no armazenamento de confiança. Quando a conexão é incluída com sucesso, o certificado é excluído deste local.

- i) Clique em **Avançar** para visualizar a página de resumo. É possível usar o botão **Voltar** para visitar páginas anteriores e fazer correções. Se você estiver feliz com as informações, clique em **Conectar** para se conectar ao gerenciador de filas remotas.

As informações sobre a conexão que você especificou são gravadas em arquivo CCDT em seu diretório da web. O caminho é `WLP_USER_DIR/generated.ccdt/ccdt-uniqueName`.

Resultados


O gerenciador de filas remotas aparece na lista de gerenciadores de filas remotas no IBM MQ Console. Desde que a conexão seja bem-sucedida, é possível administrar os objetos do gerenciador de filas remotas da mesma forma que você trabalha com os objetos de um gerenciador de filas locais.

Incluindo um gerenciador de filas remotas no IBM MQ Console usando a linha de comandos



É possível incluir um gerenciador de filas remotas no IBM MQ Console utilizando o comando **setmqweb remote** na linha de comandos. Um gerenciador de filas remotas pode ser um gerenciador de filas que está em execução em uma instalação diferente no mesmo sistema que o IBM MQ Console, ou um gerenciador de filas que está em execução em um sistema diferente.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos MQSC:

-  Sobre AIX, Linux, and Windows, você emite comandos MQSC de um **runmqsc** prompt de comando. Ver [Executando comandos MQSC interativamente em runmqsc](#) e [Executando comandos MQSC a partir de arquivos de texto em runmqsc](#). Para esta tarefa, se você estiver executando no AIX, Linux, and Windows, abra um prompt de comandos `runmqsc` que usa QM1:

```
runmqsc QM1
```

-  Sobre IBM i, você cria uma lista de comandos em um arquivo de script e, em seguida, executa o arquivo usando o comando **STRMQMMQSC** comando. Ver [Administração usando comandos MQSC em IBM i](#).
-  Sobre z/OS, os comandos MQSC podem ser emitidos a partir de diversas fontes, dependendo do comando. Ver [Fontes das quais é possível emitir comandos MQSC e PCF em IBM MQ for z/OS](#).

Certifique-se de que o servidor `mqweb` esteja configurado para permitir conexões do gerenciador de filas remotas ao IBM MQ Console. Para obter mais informações, consulte [Configurando o comportamento de conexões remotas gerenciador de filas](#).

Procedimento

1. No gerenciador de filas remotas, crie um canal de conexão do servidor para permitir a administração remota do gerenciador de filas usando o comando MQSC **DEFINE CHANNEL**. Por exemplo, para criar um canal de conexão do servidor QM1.SVRCONN para o gerenciador de filas QM1, insira o comando MQSC a seguir:

```
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

Para obter mais informações sobre o **DEFINE CHANNEL** e as opções disponíveis, consulte [DEFINIR CANAL](#).

2. Assegure-se de que um usuário apropriado esteja autorizado a administrar o gerenciador de filas e os objetos do MQ associados ao gerenciador de filas.

- **ALW** No AIX, Linux, and Windows , use o comando de controle **setmqaut** em uma linha de comandos padrão
- **z/OS** No z/OS, defina perfis RACF para fornecer ao usuário autorizado acesso ao gerenciador de filas.

Por exemplo, em AIX, Linux, and Windows, para autorizar o usuário `exampleUser` a acessar o gerenciador de filas QM1. Insira o comando de controle a seguir:

```
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

Esse usuário autorizado pode ser um dos seguintes usuários:

- Um ID do usuário que é igual ao ID do usuário que inicia o servidor mqweb que executa o IBM MQ Console no sistema do qual você deseja administrar remotamente esse gerenciador de filas.
- Um ID do usuário que corresponde a um ID do usuário e senha que é, então, incluído no comando **setmqweb remote** na etapa “7” na [página 110](#). Incluindo o ID do usuário e a senha no comando **setmqweb remote** , esse ID do usuário e a senha são usados para autenticação quando o IBM MQ Console se conecta ao gerenciador de filas.
- Um ID do usuário determinado pelas regras de segurança do canal. Por exemplo, é possível configurar uma regra de autenticação de canal no canal de conexão do servidor para permitir conexões do endereço IP do qual você usa o IBM MQ Console para administração remota e mapear todas essas conexões para um ID do usuário específico que está autorizado a usar o gerenciador de filas. Para obter mais informações, consulte [Criando novas regras de CHLAUTH para canais](#)

3. **ALW**

Se não houver listener em execução no gerenciador de filas remotas, crie um listener para aceitar conexões de rede recebidas usando o comando MQSC **DEFINE LISTENER** .

Por exemplo, para criar um listener REMOTE . LISTENER na porta 1414 para o gerenciador de filas remotas QM1, insira o comando MQSC a seguir:

```
runmqsc QM1
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
end
```

4. Assegure que o listener esteja em execução usando o comando MQSC **START LISTENER** .

ALW Por exemplo, no AIX, Linux, and Windows para iniciar o listener REMOTE . LISTENER para o gerenciador de filas QM1, insira o comando MQSC a seguir:

```
runmqsc QM1
START LISTENER(REMOTE.LISTENER)
end
```

z/OS Por exemplo, em z/OS, para iniciar o listener, insira o comando MQSC a seguir:

```
/cpf START LISTENER TRPTYPE(TCP) PORT(1414)
```

Note que o espaço de endereço do iniciador do canal deve ser iniciado antes de você poder iniciar um listener no z/OS.

5. Crie um arquivo JSON CCDT que contenha as informações de conexão do gerenciador de filas remotas:

- Gere um arquivo CCDT usando o IBM MQ Console que está associado à mesma instalação do gerenciador de filas ao qual você deseja se conectar remotamente.

No painel **Inicial**, clique no arquivo **Fazer download do arquivo de conexão**.

- Crie um arquivo CCDT no formato JSON que defina a conexão. Para obter mais informações sobre a criação de um CCDT em formato JSON, consulte [Configurando um CCDT em formato JSON](#).

O arquivo CCDT deve incluir as informações `name`, `clientConnection` e `type`. Opcionalmente, é possível incluir informações adicionais como `transmissionSecurity`. Para obter mais informações sobre todas as definições de atributo de canal CCDT, consulte a [Lista completa de definições de atributo de canal CCDT](#).

O exemplo a seguir mostra um arquivo JSON CCDT para uma conexão de gerenciador de filas remotas. Ele define o nome do canal como o mesmo nome do canal de conexão do servidor de exemplo criado na etapa “1” na página 108, e a porta da conexão como o mesmo valor da porta usada pelo listener. O host de conexão é configurado com o nome do host do sistema no qual o gerenciador de filas remotas de exemplo, QM1, está em execução:

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }]
}
```

6. Copie o arquivo JSON CCDT para o sistema no qual o IBM MQ Console está em execução
7. Na instalação que está executando o IBM MQ Console, use o comando **setmqweb remote** para incluir as informações do gerenciador de filas remotas na configuração do IBM MQ Console .

No mínimo, para incluir um gerenciador de filas remoto no IBM MQ Console, deve-se fornecer o nome do gerenciador de filas, um nome exclusivo para o gerenciador de filas (para diferenciar entre outros gerenciadores de filas remotos que podem ter o mesmo nome do gerenciador de filas) e a URL CCDT para o gerenciador de filas. O nome exclusivo é o nome de exibição no IBM MQ Console; portanto, especifique um nome que torne claro que esse é um gerenciador de fila remoto, por exemplo, "remote-QM2". Existem várias opções adicionais que podem ser especificadas, como o nome de usuário e a senha para usar para a conexão do gerenciador de filas remotas, ou detalhes do armazenamento confiável e do armazenamento de chaves. Para uma lista completa de parâmetros que podem ser especificados com o comando **setmqweb remote** , consulte [setmqweb remote](#).

Por exemplo, para incluir o gerenciador de filas remotas de exemplo QM1, usando o arquivo CCDT de exemplo, insira o comando a seguir:

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL "c:\myccdt\ccdt.json"
```

Resultados

O gerenciador de filas remotas aparece na lista de gerenciadores de filas remotas no IBM MQ Console quando a próxima lista de conexões remotas é atualizada. Desde que a conexão seja bem-sucedida, é possível administrar os objetos do gerenciador de filas remotas da mesma forma que você trabalha com os objetos de um gerenciador de filas locais.

Exemplo

O exemplo a seguir configura a conexão do gerenciador de filas remotas para um gerenciador de fila QM1.. O IBM MQ Console está autorizado a administrar o gerenciador de filas com base na autorização fornecida para o usuário `exampleUser`. As credenciais desse usuário são fornecidas para o IBM MQ Console quando o comando **setmqweb remote** é usado para configurar as informações de conexão do gerenciador de filas remotas

1. No sistema em que o gerenciador de filas remotas QM1 é, um canal de conexão do servidor e um listener são criados.. O listener é iniciado e a autorização é fornecida para o usuário `exampleUser`

para administrar o gerenciador de fila Por exemplo, no AIX, Linux, and Windows, execute os seguintes comandos:

```
runmqsc QM1
#Define the server connection channel that will accept connections from the Console
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
# Define the listener to use for the connection from the Console
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
# Start the listener
START LISTENER(REMOTE.LISTENER)
end

#Set mq authorization for exampleUser to access the queue manager
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

2. No sistema no qual o IBM MQ Console está em execução, um arquivo QM1_ccdt.json é criado com as seguintes informações de conexão:

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }]
}
```

3. No sistema no qual o IBM MQ Console está em execução, as informações de conexão do gerenciador de filas remotas para o gerenciador de filas QM1 são incluídas no servidor mqWeb. As credenciais para exampleUser são incluídas nas informações de conexão:

```
setmqweb remote add -uniqueName "remote-QM1" -qmgrName "QM1" -ccdtURL
"c:\myccdt\QM1_ccdt.json" -username "exampleUser" -password "password"
```

4. O IBM MQ Console mostra o gerenciador de filas remotas QM1.

V 9.4.0 IBM MQ Console: Trabalhando com objetos

Cada gerenciador de filas do IBM MQ tem vários tipos diferentes de objeto associados a ele.

Sobre esta tarefa

É possível usar o console para trabalhar com os tipos de objeto do IBM MQ a seguir:

- Filas
- Objetos de eventos:
 - tópicos
 - Assinaturas
- Objetos de aplicativos:
 - Conexões
 - Canais do aplicativo
 - Instâncias do canal do aplicativo
- Objetos de rede do MQ :
 - Gerenciadores de filas conectados
 - Canais do gerenciador de filas
 - Instâncias do canal gerenciador de filas

Procedimento

Para trabalhar com um objeto do IBM MQ:

1. Na visualização da lista do gerenciador de filas, clique no gerenciador de filas que possui os objetos com os quais você deseja trabalhar.
2. Clique na guia de rede Filas, Eventos, Aplicativos ou MQ para selecionar o tipo de objeto com o qual você deseja trabalhar
3. Consulte um dos seguintes tópicos para obter instruções detalhadas para trabalhar com os objetos.

IBM MQ Console: Trabalhando com filas

É possível visualizar as filas que existem para um gerenciador de filas específico na guia **Filas**. É possível incluir e excluir filas, incluir e limpar mensagens em uma fila, procurar mensagens, visualizar e configurar as propriedades de uma fila e gerenciar os registros de autoridade de uma fila.

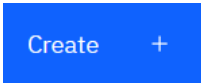
Sobre esta tarefa

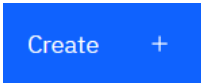
A visualização de filas lista as filas que existem para um gerenciador de filas específico. Você acessa a lista de filas clicando em um gerenciador de filas e selecionando a guia **Filas**. É possível selecionar filas individuais na lista, com as quais trabalhar.

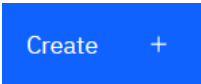
 Não é possível visualizar nem editar registros de autoridade para filas no z/OS.

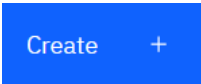

Procedimento

- Para adicionar uma fila:




- a) Na guia **Filas**, clique no botão Criar .
 - b) Selecione o tipo de fila que você deseja criar:
 - Fila local - armazena mensagem dentro do gerenciador de filas ao qual pertence.
 - Fila de alias - um ponteiro para outra fila no mesmo gerenciador de filas.
 - Fila remota - um ponteiro para outra fila em um gerenciador de filas diferente.
 - Fila modelo - um modelo para uma fila usada quando um gerenciador de filas dinâmico é criado.
 - c) Forneça as informações necessárias para o tipo de fila que você está criando. Por padrão, as propriedades mínimas recomendadas são exibidas para que você forneça valores para elas. É possível visualizar todas as propriedades disponíveis selecionando **Criação customizada**.
 - d) Clique em **Criar**. A nova fila é criada.
- Para colocar mensagens em uma fila:
 - a) Clique na fila na qual você deseja incluir mensagens na lista na visualização de lista de filas. Não é possível selecionar uma fila modelo.

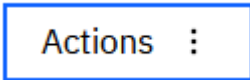



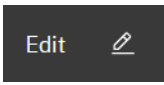

- b) Clique no botão Criar .
 - c) Insira a mensagem que você deseja colocar na fila.
 - d) Clique em **Criar**.
- Para limpar mensagens de uma fila:
 - a) Clique na fila local da qual você deseja limpar mensagens na lista de fila
 - b) Clique no ícone Limpar fila 
 - c) Confirme que você deseja limpar a fila, clicando em **Limpar fila**.

• **V 9.4.0**

Para excluir uma mensagem individual de uma fila:


- a) Localize a mensagem que você deseja excluir
 - b) Clique no ícone excluir próximo à mensagem .
 - c) Confirme que deseja limpar a mensagem clicando em **Excluir**.
- Para pesquisar mensagens em uma fila, clique na fila na visualização de lista de filas. Uma lista das mensagens nessa fila é exibida.
- Para excluir uma fila:
- a) Clique na fila local que você deseja excluir na lista de filas



- b) Clique no botão Ações e selecione **Excluir fila**.
 - c) Confirme que você deseja excluir a fila, clicando em **Excluir**. A fila é excluída.
- Para visualizar e editar as propriedades de uma fila:
- a) Selecione **Visualizar configuração** no menu  ao lado da fila que você deseja editar
- 
- b) Clique no botão Editar
 - c) Edite as propriedades conforme necessárias. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos. Para obter informações sobre as propriedades, consulte [Propriedades da Fila](#) na IBM MQ Explorer documentação
 - d) Clique em **Salvar** para salvar as alterações.
- Para visualizar e editar os registros de autoridade de uma fila:
- a) Selecione **Visualizar configuração** no menu  ao lado da fila que você deseja editar
 - b) Clique na guia **Segurança**.
 - c) Trabalhe com os registros de autoridade conforme descrito para os registros de autoridade do gerenciador de filas. Consulte [“IBM MQ Console: Trabalhando com Registros de Autoridade do Gerenciador de Filas”](#) na página 103.

• **V 9.4.0**

Para visualizar os objetos IBM MQ associados a uma fila:

- a) Selecione **Visualizar objetos associados** no menu  ao lado da fila que você deseja visualizar
- b) Visualizar os objetos no painel que aparece. Clique nos links para visualizar mais detalhes sobre cada um dos objetos listados

É possível usar o painel para visualizar quais aplicativos estão colocando mensagens em filas, e ver os relacionamentos entre diferentes filas. Isso pode ajudar a identificar e resolver problemas.

• **V 9.4.0** **IBM MQ Console: Trabalhando com tópicos**

É possível usar o IBM MQ Console para incluir e excluir tópicos e visualizar e configurar as propriedades de um tópico.

Sobre esta tarefa

A visualização de tópicos lista os tópicos que existem para um gerenciador de filas específico. Você acessa os tópicos da guia **Eventos** do gerenciador de filas. É possível selecionar tópicos individuais na lista, com os quais trabalhar.

 Não é possível visualizar nem editar registros de autoridade para um tópico no z/OS.

Procedimento

- Para incluir um tópico:

a) Na visualização do gerenciador de filas, abra a guia **Eventos** e clique em **Tópicos**.

b) Clique no botão Criar .

c) Forneça as informações necessárias para o tópico que está sendo criado. Por padrão, as propriedades mínimas recomendadas são exibidas para que você forneça valores para elas. É possível visualizar todas as propriedades disponíveis selecionando **Criação customizada**.

d) Clique em **Criar**. O novo tópico é criado.

- Para excluir um tópico:

a) Clique no ícone de Chave inglesa  ao lado do tópico que você deseja excluir.

b) Na visualização Editar fila, clique em **Excluir tópico**.

c) Confirme que você deseja excluir o tópico, clicando em **Excluir**. O tópico é excluído.

- Para visualizar e editar as propriedades de um tópico:

a) Clique no ícone de Chave inglesa  ao lado do tópico que você deseja editar.

b) Clique no botão Editar .

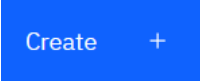
c) Edite as propriedades conforme necessárias. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos. Para obter informações sobre as propriedades, consulte [Propriedades do tópico](#) na documentação do MQ Explorer.

d) Clique em **Salvar** para salvar as alterações.

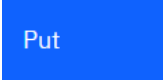
- Para publicar uma mensagem em um tópico, deve-se ter pelo menos uma assinatura correspondente.

a) Clique no tópico no qual deseja publicar na lista de tópicos.

b) Clique no nome da assinatura correspondente..


c) Clique no botão Criar .

d) Insira a mensagem que você deseja publicar.

e) Clique no botão Colocar . A mensagem é gravada em todas as assinaturas correspondentes.

- Para assinar um tópico, consulte [“IBM MQ Console: Trabalhando com assinaturas”](#) na página 115:

- Para visualizar e editar os registros de autoridade de um tópico:

- a) Clique no ícone de Chave inglesa  ao lado do tópico para o qual você deseja editar registros de autoridade.
- b) Clique na guia **Segurança**.
- c) Trabalhe com os registros de autoridade conforme descrito para os registros de autoridade do gerenciador de filas. Consulte “[IBM MQ Console: Trabalhando com Registros de Autoridade do Gerenciador de Filas](#)” na página 103.

V 9.4.0 IBM MQ Console: Trabalhando com assinaturas

É possível usar o IBM MQ Console para incluir e excluir assinaturas e visualizar e configurar as propriedades de uma assinatura.



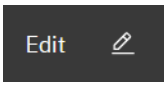
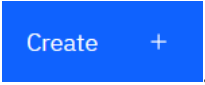
Sobre esta tarefa

A visualização de assinaturas lista as assinaturas que existem para um gerenciador de filas específico. Você acessa as assinaturas na guia **Eventos** do gerenciador de filas. É possível selecionar tópicos individuais na lista, com os quais trabalhar. É possível selecionar assinaturas individuais da lista com a qual trabalhar.

Para obter mais informações sobre assinaturas, consulte [Assinantes e assinaturas](#) e [DEFINE SUB](#).

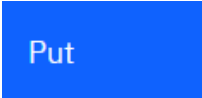
 **z/OS** Não é possível visualizar ou editar registros de autoridade para uma assinatura no z/OS.

Procedimento

- Para incluir uma assinatura:
 - a) Na visualização do gerenciador de filas, abra a guia **Eventos** e clique em **Assinaturas**.
 - b) Escolha se você deseja criar uma assinatura gerenciada ou não gerenciada.
 - c) Forneça as informações necessárias para a assinatura que você está criando. Por padrão, as propriedades mínimas recomendadas são exibidas para que você forneça valores para elas. É possível visualizar todas as propriedades disponíveis selecionando **Criação customizada**.
 - d) Clique em **Criar**. A nova assinatura é criada.
- Para excluir uma assinatura:
 - a) Clique no ícone de Chave inglesa  ao lado da assinatura que você deseja excluir.
 - b) Na visualização Editar fila, clique em **Excluir assinatura**.
 - c) Confirme se você deseja excluir a assinatura clicando em **Excluir**. A assinatura será excluída.
- Para visualizar e editar as propriedades de uma assinatura:
 - a) Clique no ícone de Chave inglesa  ao lado da assinatura que você deseja editar.
 - b) Clique no botão Editar .
 - c) Edite as propriedades conforme necessárias. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos.
 - d) Clique em **Salvar** para salvar as alterações.
- Para publicar uma mensagem no tópico para o qual a assinatura é inscrita:
 - a) Clique na assinatura em cujo tópico você deseja publicar na lista de assinaturas.
 - b) Clique no botão Criar .

c) Insira a mensagem que você deseja publicar.

A blue rectangular button with the word "Put" written in white text.


d) Clique no botão Colocar . A mensagem é gravada em todas as assinaturas que correspondem ao tópico no qual você publicou.

IBM MQ Console: Trabalhando com canais do gerenciador de filas

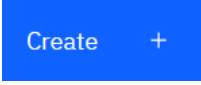





É possível usar o IBM MQ Console para trabalhar com canais do gerenciador de filas: é possível incluir e excluir canais do gerenciador de filas, iniciar e parar canais, reconfigurar e resolver canais e efetuar ping dos canais. Também é possível visualizar e configurar as propriedades de um canal do gerenciador de filas e gerenciar os registros de autoridade para ele.






Sobre esta tarefa

Um canal de gerenciador de filas é um link de comunicação lógica para transmissão de mensagens entre os gerenciadores de filas em uma rede. A visualização do canal do gerenciador de filas inclui um painel que mostra uma visualização rápida de quantos canais estão em execução, quantos estão realizando novas tentativas e quantos pararam.

 Não é possível visualizar nem editar registros de autoridade para um canal no z/OS.

Procedimento

- Para incluir um canal do gerenciador de filas:
 - a) Na visualização do gerenciador de filas, abra a guia **MQ** e clique em **Canais do gerenciador de filas**
e clique no botão Criar  ..
 - b) Selecione o tipo de canal do gerenciador de filas que você deseja criar e clique no botão Avançar .
 - c) Forneça as informações necessárias para o canal que você está criando. Por padrão, as propriedades mínimas recomendadas são exibidas para que você forneça valores para elas. É possível visualizar todas as propriedades disponíveis selecionando **Criação customizada**.
 - d) Clique em **Criar**. O novo canal é criado com o status **inativo**.
- Para iniciar um canal do gerenciador de filas:
 - a) Localize na lista o canal que você deseja iniciar.
 - b) Selecione **Iniciar** no menu .
- Para parar um canal do gerenciador de filas:
 - a) Localize na lista o canal que você deseja parar.
 - b) Selecione **Parar** no menu .
- Para visualizar as propriedades de um canal do gerenciador de filas:
 - a) Localize o canal na lista.
 - b) Selecione **Visualizar configurações** no menu .
 - c) Assegure-se de que a guia **Propriedades** esteja selecionada. Para editar as propriedades, clique no botão Editar 

- d) Edite as propriedades conforme necessárias. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos. Para obter mais informações sobre as propriedades, consulte [Propriedades do canal](#) na documentação do MQ Explorer.
- e) Clique em **Salvar** para salvar as alterações.
- Para reconfigurar um canal do gerenciador de filas:
 - a) Localize o canal na lista.
 - b) Selecione **Avançado** no menu .
 - c) Na seção **Reconfigurar**, especifique um número de sequência de mensagem.
Será necessário reconfigurar um canal, se ele não iniciar porque as duas extremidades discordam sobre o número de sequência da próxima mensagem a ser enviada. O número de sequência da mensagem especifica esse número.
 - d) Clique em **Reconfigurar canal**.
- Para resolver um canal do remetente ou do servidor:
 - a) Localize o canal na lista.
 - b) Selecione **Avançado** no menu .
 - c) Na seção **Resolver**, escolha se você deseja confirmar ou recuperar o lote atual de mensagens clicando em **Restaurar mensagens na fila de transmissão** ou em **Descartar mensagens**.
- Para efetuar ping em um canal do gerenciador de filas:
 - a) Localize o canal na lista.
 - b) Selecione **Ping** no menu .
- Para visualizar e editar registros de autoridade para um canal do gerenciador de filas:
 - a) Localize o canal na lista.
 - b) Selecione **Visualizar configuração** no menu .
 - c) Clique na guia **Segurança**.
 - d) Trabalhe com os registros de autoridade conforme descrito para os registros de autoridade do gerenciador de filas. Consulte [“IBM MQ Console: Trabalhando com Registros de Autoridade do Gerenciador de Filas”](#) na página 103.
- Para excluir um canal do gerenciador de filas:
 - a) Localize o canal na lista.
 - b) Selecione **Configurar** no menu .
 - c) Clique em **Excluir canal**.

IBM MQ Console: Trabalhando com canais de aplicativos

É possível usar o IBM MQ Console para trabalhar com canais do aplicativo: é possível incluir e excluir canais, iniciar e parar canais, reconfigurar e resolver canais e efetuar ping de canais. Também é possível visualizar e configurar as propriedades de um canal do aplicativo, além de gerenciar registros de autoridade do canal.

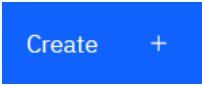






Sobre esta tarefa





Um canal de aplicativo é um link de comunicação lógica usado por aplicativos para conectar-se a um gerenciador de filas em uma rede. A visualização do canal de aplicativo inclui um painel que mostra uma

visualização rápida de quantos canais estão em execução, quantos estão realizando novas tentativas e quantos pararam.

 Não é possível visualizar nem editar registros de autoridade para um canal no z/OS.

Procedimento

- Para incluir um canal de aplicativo:
 - a) Na visualização do gerenciador de filas, abra a guia **Aplicativos** e clique em **Canais do aplicativo** e clique no botão Criar 
 - b) Clique no botão Avançar 
 - c) Forneça as informações necessárias para o canal que você está criando. Por padrão, as propriedades mínimas recomendadas são exibidas para que você forneça valores para elas. É possível visualizar todas as propriedades disponíveis selecionando **Criação customizada**.
 - d) Clique em **Criar**. O novo canal é criado com o status **inativo**.
- Para iniciar um canal de aplicativo:
 - a) Localize na lista o canal que você deseja iniciar.
 - b) Selecione **Iniciar** no menu 
- Para parar um canal de aplicativo:
 - a) Localize na lista o canal que você deseja parar.
 - b) Selecione **Parar** no menu 
- Para visualizar as propriedades de um canal de aplicativo:
 - a) Localize o canal na lista.
 - b) Selecione **Visualizar configuração** no menu 
 - c) Assegure-se de que a guia **Propriedades** esteja selecionada. Para editar as propriedades, clique no botão Editar 
 - d) Edite as propriedades conforme necessárias. Se a caixa de texto da propriedade estiver desativada, a propriedade será somente leitura ou poderá ser editada somente a partir da linha de comandos. Para obter mais informações sobre as propriedades, consulte [Propriedades do canal](#) na documentação do MQ Explorer.
 - e) Clique em **Salvar** para salvar as alterações.
- Para reconfigurar um canal de aplicativo:
 - a) Localize o canal na lista.
 - b) Selecione **Avançado** no menu 
 - c) Na seção **Reconfigurar**, especifique um número de sequência de mensagem. Será necessário reconfigurar um canal, se ele não iniciar porque as duas extremidades discordam sobre o número de sequência da próxima mensagem a ser enviada. O número de sequência da mensagem especifica esse número.
 - d) Clique em **Reconfigurar canal**.

- Para resolver um canal do remetente ou do servidor:
 - a) Localize o canal na lista.
 - b) Selecione **Avançado** no menu  .
 - c) Na seção **Resolver**, escolha se você deseja confirmar ou recuperar o lote atual de mensagens clicando em **Restaurar mensagens na fila de transmissão** ou em **Descartar mensagens**.
- Para executar ping em um canal:
 - a) Localize o canal na lista.
 - b) Selecione **Ping** no menu  .
- Para visualizar e editar registros de autoridade para um canal de aplicativo:
 - a) Localize o canal na lista.
 - b) Selecione **Configurar** no menu  .
 - c) Clique na guia **Segurança**.
 - d) Trabalhe com os registros de autoridade conforme descrito para os registros de autoridade do gerenciador de filas. Consulte [“IBM MQ Console: Trabalhando com Registros de Autoridade do Gerenciador de Filas”](#) na página 103.
- Para excluir um canal de aplicativo:
 - a) Localize o canal na lista.
 - b) Selecione **Configurar** no menu  .
 - c) Clique em **Excluir canal**.

IBM MQ Console: Trabalhando com aplicativos

É possível usar o IBM MQ Console para visualizar informações sobre aplicativos conectados a um gerenciador de filas..

Sobre esta tarefa

Um aplicativo é conectado a um gerenciador de filas em uma rede usando um canal de conexão do servidor. A visualização de aplicativos inclui um painel que mostra uma visualização rápida de quantos aplicativos estão conectados a um gerente da fila.

Procedimento

- Para visualizar informações do aplicativo:
 - a) Na visualização do gerenciador de filas, abra a guia **Aplicativos** .
 - b) Clique em **Aplicativos Conectados** para abrir a visualização de aplicativos
 - c) Se houver diversas instâncias de um aplicativo, clique na flecha para baixo para visualizar detalhes de cada instância
 - d) Clique em objetos na visualização para obter mais detalhes

IBM MQ Console: Working with storage classes

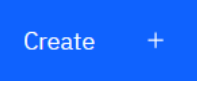
You can use the IBM MQ Console to add, view, delete and update storage classes on z/OS queue managers.

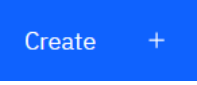
About this task

The storage classes view lists the storage classes that exist for a specific queue manager. You access **Storage classes** from the sidebar on the queue manager **Queues** tab.

See [Storage classes for IBM MQ for z/OS](#) and [DEFINE STGLASS](#) for more information about storage classes.


Procedure

- To add a storage class:
 - From the queue manager view, open the **Queues** tab, and click **Storage classes**.
 - On the **Storage classes** screen, click the **Create**  button.
 - Provide the required information for the storage class you are creating.
By default, the minimum recommended properties you need to provide values for are displayed. You can view all of the available properties by selecting **Custom create**.

- Click the **Create**  button.
The new storage class is created.

- To delete a storage class:

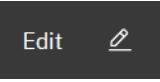


- Click the spanner button  next to the storage class that you want to delete.
- In the Edit storage class view, click **Delete storage class**.
- Confirm that you want to delete the queue by clicking **Delete**. The storage class is deleted.

- To view and edit the properties of a storage class:



- Click the spanner button  next to the storage class that you want to edit.

- Click the Edit button .
- Edit the properties as required. If the property text box is disabled, the property is read-only, or can be set only at the time of creation.
- Click **Save** to save your changes.

IBM MQ Console: Working with page sets and buffer pools

You can use the IBM MQ Console to view page sets and buffer pools on z/OS queue managers.

About this task

The page sets and buffer pools views list the page sets and buffer pools that exist for a specific queue manager. You access the **Page sets** and **Buffer pools** views from the sidebar of the queue manager **Queues** tab

See [Page sets for IBM MQ for z/OS](#) for more information about page sets, and [Buffers and buffer pools for IBM MQ for z/OS](#) for more information about buffer pools.

Procedure

- To view the properties of a page set



Click the spanner button next to the page set that you want to view.


- To view the properties of a buffer pool



Click the spanner button next to the buffer pool that you want to view.

Configurações IBM MQ Console

É possível especificar algumas configurações gerais para o IBM MQ Console

Clique no ícone de configurações  para alternar para a visualização de configurações do IBM MQ Console

Use as configurações para controlar os recursos a seguir:

- Atualização automática de gerenciadores de filas a cada dez segundos. Esse recurso pode ser ligado ou desligado.
- Se os objetos do sistema são exibidos. É possível especificá-lo para todos os tipos de objeto ou selecionar os tipos de objeto individualmente.
- Se as informações de rastreamento são coletadas ou não.

Windows

Linux

Administração usando o IBM MQ Explorer

O IBM MQ Explorer permite executar administração local ou remota da sua rede a partir de um computador que esteja executando Windows ou Linux x86-64 somente.

IBM MQ for Windows e IBM MQ para Linux x86-64 fornecem uma interface de administração denominada IBM MQ Explorer para executar tarefas de administração como uma alternativa para o uso de comandos de controle ou MQSC. [Comparando Conjuntos de Comandos](#) mostra o que pode ser feito usando o IBM MQ Explorer.

O IBM MQ Explorer permite desempenhar a administração local ou remota de sua rede a partir de um computador que está executando o Windows x86-64 do Linux, apontando o IBM MQ Explorer para os gerenciadores de filas e clusters que você está interessado. É possível conectar-se remotamente aos gerenciadores de filas que estão em execução em qualquer plataforma suportada, incluindo z/OS, ativando todo seu backbone do sistema de mensagens para ser exibido, explorado e alterado a partir do console.

Para configurar os gerenciadores de fila IBM MQ remotos para que o IBM MQ Explorer possa administrá-los, consulte [“Software obrigatório e definições para IBM MQ Explorer”](#) na página 123.

Ele permite que você execute tarefas, tipicamente associadas com a configuração e ajuste fino do ambiente de trabalho para o IBM MQ, localmente ou remotamente em um domínio de sistema Windows ou Linux x86-64.

No Linux, o IBM MQ Explorer pode falhar ao ser iniciado se você tiver mais de uma instalação Eclipse. Se isso acontecer, inicie o IBM MQ Explorer usando um ID de usuário diferente daquele que você usa para a outra instalação Eclipse.

No Linux, para iniciar o IBM MQ Explorer com êxito, deve-se conseguir gravar um arquivo para o seu diretório inicial e o diretório inicial deve existir.

IBM MQ Explorer pode ser instalado a partir do download IBM MQ Explorer independente disponível em Fix Central. Para obter mais informações, consulte [Instalando e desinstalando o IBM MQ Explorer como um aplicativo independente em Linux e Windows](#)

Windows

Linux

O que Você Pode Fazer com IBM MQ Explorer

É possível usar o IBM MQ Explorer para executar tarefas de administração usando vários diálogos de Visualizações de conteúdo e Propriedades. Também é possível estender o IBM MQ Explorer gravando um ou mais plug-ins do Eclipse.

Tarefas do IBM MQ Explorer

Com o IBM MQ Explorer, é possível executar as tarefas a seguir:

- Criar e excluir um gerenciador de filas (apenas em sua máquina local).
- Iniciar e parar um gerenciador de filas (apenas em sua máquina local).
- Definir, exibir e alterar as definições de objetos do IBM MQ, tais como filas e canais.
- Procurar as mensagens em uma fila.
- Iniciar e parar um canal.
- Visualizar informações de status sobre um canal, um listener, uma fila ou objetos de serviço.
- Visualizar gerenciadores de filas em um cluster.
- Verificar quais aplicativos, usuários ou canais têm uma fila aberta específica.
- Criar um novo cluster de gerenciador de filas usando o assistente Criar novo cluster.
- Incluir um gerenciador de filas em um cluster usando o assistente Incluir gerenciador de filas no cluster.
- Gerenciar o objeto de informações sobre autenticação usado com a segurança de canal da Segurança da Camada de Transporte (TLS).
- Criar e excluir iniciadores de canais, os monitores do acionador e listeners.
- Iniciar ou parar os servidores de comandos, os inicializadores de canais, os monitores acionadores e os listeners.
- Configurar serviços específicos para iniciar automaticamente quando um gerenciador de filas for iniciado.
- Modificar as propriedades de gerenciadores de filas.
- Mudar o gerenciador de filas padrão local.
- Criar objetos do JMS por meio de objetos do IBM MQ e objetos do IBM MQ por meio de objetos do JMS.
- Criar um Connection Factory do JMS para qualquer um dos tipos suportados atualmente.
- Modificar os parâmetros para qualquer serviço, como o número da porta TCP para um listener ou um nome de fila inicializador de canais.
- Iniciar ou parar o rastreamento de serviço.

Diálogos de Visualizações de Conteúdo e Propriedades

É possível executar tarefas de administração usando vários diálogos de Visualizações de conteúdo e Propriedades.

Visão de Conteúdo

Uma visualização Conteúdo é um painel que pode exibir o seguinte:

- Atributos e opções administrativas relativas ao próprio IBM MQ.
- Atributos e opções administrativas relativas a um ou mais objetos relacionados.
- Atributos e opções administrativas para um cluster.

Diálogos de propriedades

Um diálogo de propriedade é um painel que exibe os atributos relacionados a um objeto em uma série de campos, alguns dos quais é possível editar.

Navegue pelo IBM MQ Explorer usando a visualização Navegador. O Navegador permite que você selecione a Visualização Conteúdo que você precisa.

Estendendo o IBM MQ Explorer

O IBM MQ Explorer apresenta informações em um estilo consistente com aquele da estrutura do Eclipse e o plug-in outros aplicativos que o Eclipse suporta.

Com a extensão do IBM MQ Explorer, os administradores do sistema têm a capacidade de customizar o IBM MQ Explorer para aprimorar a forma como eles administram o IBM MQ.

Para obter mais informações, consulte [Estendendo o MQ Explorer](#).

Decidindo quando usar o IBM MQ Explorer

Ao decidir quando usar o IBM MQ Explorer em sua instalação, considere as informações listadas neste tópico.

Você precisa estar ciente dos seguintes pontos:

Nomes de Objetos

Se você usar nomes em letras minúsculas para os gerenciadores de filas e outros objetos com o IBM MQ Explorer, ao trabalhar com os objetos usando comandos MQSC, você deverá colocar os nomes do objeto entre aspas simples ou o IBM MQ não os reconhecerá.

Gerenciadores de filas grandes

O IBM MQ Explorer funciona melhor com gerenciadores de filas pequenos. Se você tiver um grande número de objetos em um único gerenciador de filas, você poderá enfrentar atrasos enquanto o IBM MQ Explorer extrai as informações necessárias a serem apresentadas em uma visualização.

Clusters

Clusters do IBM MQ pode conter potencialmente centenas ou milhares de gerenciadores de filas. O IBM MQ Explorer apresenta os gerenciadores de filas em um cluster usando uma estrutura em árvore. O tamanho físico de um cluster não afeta a velocidade do IBM MQ Explorer significativamente porque o IBM MQ Explorer não se conecta aos gerenciadores de filas no cluster até que você os selecione.

Configurando o IBM MQ Explorer

Essa seção destaca as etapas necessárias para configurar o IBM MQ Explorer.

- [“Software obrigatório e definições para IBM MQ Explorer” na página 123](#)
- [“Segurança para IBM MQ Explorer” na página 124](#)
- [“Mostrando e ocultando gerenciadores de filas e clusters no IBM MQ Explorer” na página 127](#)
- [“Associação de cluster e o IBM MQ Explorer” na página 128](#)
- [“Conversão de dados para IBM MQ Explorer” na página 129](#)

Software obrigatório e definições para IBM MQ Explorer

Certifique-se de atender os seguintes requisitos antes de tentar utilizar o IBM MQ Explorer.

O IBM MQ Explorer pode se conectar aos gerenciadores de filas remotas utilizando o protocolo de comunicação TCP/IP somente.

Verifique se:

1. Um servidor de comandos está em execução em cada gerenciador de filas administrado remotamente.
2. Um objeto de listener TCP/IP adequado deve estar em execução em cada gerenciador de filas remotas. Esse objeto pode ser o listener do IBM MQ ou, em sistemas AIX and Linux, o daemon inetd.
3. Um canal de conexão do servidor, por padrão denominado SYSTEM.ADMIN.SVRCONN, existe em todos os gerenciadores de filas remotas.

É possível criar o canal usando o comando MQSC a seguir:

```
DEFINE CHANNEL (SYSTEM.ADMIN.SVRCONN) CHLTYPE (SVRCONN)
```

Esse comando cria uma definição de canal básico. Se você deseja uma definição mais sofisticadas (para configurar a segurança, por exemplo), você precisa de parâmetros adicionais. Para obter mais informações, consulte [DEFINE CHANNEL](#).

4. O sistema de filas, SYSTEM.MQEXPLORER.REPLY.MODEL, deve existir.

Segurança para IBM MQ Explorer

Se você estiver usando o IBM MQ em um ambiente no qual seja importante controlar o acesso de usuário a objetos particulares, poderá ser necessário considerar os aspectos de segurança de uso do IBM MQ Explorer.

Autorização para usar o IBM MQ Explorer

Qualquer usuário pode usar o IBM MQ Explorer, mas é necessário que determinadas autoridades se conectem, acessem e gerenciem gerenciadores de filas.

Para executar tarefas administrativas locais usando o IBM MQ Explorer, um usuário precisa ter a autoridade necessária para executar as tarefas administrativas. Se o usuário é um membro do grupo mqm, o usuário tem autoridade para executar todas as tarefas administrativas locais.

Para se conectar a um gerenciador de filas remotas e executar tarefas administrativas remotas usando o IBM MQ Explorer, o usuário que está executando o IBM MQ Explorer precisa ter as autoridades a seguir:

- Autoridade CONNECT no objeto do gerenciador de filas de destino
- Autoridade INQUIRE no objeto do gerenciador de filas de destino
- Autoridade DISPLAY no objeto do gerenciador de filas de destino
- Autoridade INQUIRE para a fila, SYSTEM.MQEXPLORER.REPLY.MODEL
- Autoridade DISPLAY para a fila, SYSTEM.MQEXPLORER.REPLY.MODEL
- Autoridade INPUT (get) para a fila, SYSTEM.MQEXPLORER.REPLY.MODEL
- Autoridade OUTPUT (put) para a fila, SYSTEM.MQEXPLORER.REPLY.MODEL
- Autoridade OUTPUT (put) para a fila, SYSTEM.ADMIN.COMMAND.QUEUE
- Autoridade INQUIRE na fila, SYSTEM.ADMIN.COMMAND.QUEUE
- Autoridade para executar a ação selecionada

Nota: Autoridade INPUT diz respeito à entrada para o usuário a partir de uma fila (uma operação get). Autoridade OUTPUT diz respeito à saída do usuário para uma fila (uma operação put).

Para se conectar a um gerenciador de filas remotas no IBM MQ for z/OS e executar tarefas administrativas remotas usando o IBM MQ Explorer, o seguinte deve ser fornecido:

- Um perfil RACF para a fila do sistema SYSTEM.MQEXPLORER.REPLY.MODEL
- Um perfil do RACF para as filas, AMQ.MQEXPLORER.*

Além disso, o usuário que está executando o IBM MQ Explorer precisa ter as autoridades a seguir:

- RACF Autoridade UPDATE para a fila do sistema SYSTEM.MQEXPLORER.REPLY.MODEL
- RACF UPDATE autoridade para as filas, AMQ.MQEXPLORER.*
- Autoridade CONNECT no objeto do gerenciador de filas de destino

- Autoridade para executar a ação selecionada
- Autoridade READ para todos os perfis hlq.DISPLAY.object na classe MQCMDS

Para obter informações sobre como conceder autoridade a objetos do IBM MQ, consulte [Concedendo acesso a um objeto do IBM MQ em sistemas AIX, Linux, and Windows](#).

Se um usuário tenta executar uma operação que ele não está autorizado a executar, o gerenciador de filas de destino chama procedimentos falha de autorização e a operação falha.

O filtro padrão no IBM MQ Explorer é exibir todos os objetos do IBM MQ. Se houver quaisquer objetos do IBM MQ aos quais um usuário não tiver autoridade DISPLAY, falhas de autorização serão geradas. Se eventos de autoridade estão sendo registrados, restringir o intervalo de objetos que são exibidos para aqueles objetos aos quais o usuário tem autoridade DISPLAY.

Segurança para conectar-se a gerenciadores de filas remotas do IBM MQ Explorer

Deve-se proteger o canal entre o IBM MQ Explorer e cada gerenciador de filas remotas.

O IBM MQ Explorer se conecta a gerenciadores de filas remotas como um aplicativo cliente de MQI. Isso significa que cada gerenciador de filas remotas deve ter uma definição de um canal de conexão do servidor e um listener TCP/IP apropriado. Se você não proteger seu canal de conexão do servidor, é possível para um aplicativo malicioso se conectar ao canal de conexão do mesmo servidor e obter acesso a objetos do gerenciador de filas com autoridade ilimitada. A fim de proteger o canal de conexão do servidor ou especificar um valor não em branco para o atributo MCAUSER do canal, use registros de autenticação de canal ou use uma saída de segurança.

O valor padrão do atributo MCAUSER é o ID do usuário local. Se você especificar um nome de usuário que não esteja em branco como o atributo MCAUSER do canal de conexão do servidor, todos os programas que se conectam ao gerenciador de filas utilizando esse canal executam com a identidade do usuário nomeado e têm o mesmo nível de autoridade. Isso não ocorrerá se você utilizar registros de autenticação de canal.

Usando uma saída de segurança com o IBM MQ Explorer

É possível especificar uma saída de segurança padrão e as saídas de segurança específicas do gerenciador de filas usando o IBM MQ Explorer.

É possível definir uma saída de segurança padrão, que pode ser usada para todas as novas conexões do cliente a partir do IBM MQ Explorer. Esta saída padrão pode ser substituída no momento em que uma conexão é feita. Também é possível definir uma saída de segurança para um gerenciador de filas único ou um conjunto de gerenciadores de filas, que entra em vigor quando uma conexão é feita. Especifique as saídas usando o IBM MQ Explorer. Para obter mais informações, veja a Ajuda do IBM MQ Explorer.

Usando o IBM MQ Explorer para se conectar a um gerenciador de filas remotas que usa canais MQI ativados para TLS

O IBM MQ Explorer se conecta a gerenciadores de filas remotas usando um canal MQI. Se desejar assegurar o canal MQI usando a segurança TLS, você deverá estabelecer o canal usando uma tabela de definição de canal de cliente.

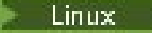


Para obter informações sobre como estabelecer um canal MQI usando uma tabela de definição de canal cliente, consulte [IBM MQ MQI clients](#).

Depois de estabelecer o canal usando uma tabela de definição de canal de cliente, é possível usar o IBM MQ Explorer para se conectar a um gerenciador de filas remotas que usa o canal MQI ativado para TLS, conforme descrito em [“Tarefas no sistema que hospeda o gerenciador de filas remotas”](#) na página 125 e [“Tarefas no sistema que hospeda o IBM MQ Explorer”](#) na página 126.

Tarefas no sistema que hospeda o gerenciador de filas remotas

No sistema que hospeda o gerenciador de filas remotas, execute as seguintes tarefas:

1. Defina uma conexão de servidor e um par de canais de conexão de cliente e especifique o valor apropriado para o atributo *SSLCIPH* na conexão do servidor em ambos os canais. Para obter mais informações sobre o atributo *SSLCIPH*, consulte [Protegendo os canais com TLS](#).
2. Envie a tabela de definição de canal AMQCLCHL .TAB, que está localizada no diretório do gerenciador de filas @ipcc, para o sistema que hospeda o arquivo IBM MQ Explorer.
3. Inicie um listener TCP/IP em uma porta designada.
4. Coloque ambos os certificados CA e TLS pessoal no diretório SSL do gerenciador de filas:

-   /var/mqm/qmgrs/+QMNAME+/SSL para sistemas AIX and Linux.
-  C:\Program Files\IBM\MQ\qmgrs\+QMNAME+\SSL para sistemas Windows.

Em que +QMNAME+ é um token representando o nome do gerenciador de filas.

5. Crie um arquivo do banco de dados de chave do tipo CMS denominado key .kdb. Armazene a senha do banco de dados de chave em um arquivo, especificando o parâmetro -stash no comando **runmqakm** que é usado para criar o banco de dados de chave.
6. Inclua os certificados de CA para o banco de dados de chaves criado na etapa anterior.
7. Importe o certificado pessoal para o gerenciador de filas para o banco de dados de chave.

Para obter informações mais detalhadas sobre como trabalhar com TLS em sistemas Windows, consulte [Trabalhando com TLS no AIX, Linux, and Windows](#).

Tarefas no sistema que hospeda o IBM MQ Explorer

No sistema que hospeda o IBM MQ Explorer, execute as seguintes tarefas:

1. Crie um arquivo de banco de dados de chave do tipo JKS denominado key .jks. Configure uma senha para este arquivo de banco de dados chave.
O keystore que o IBM MQ Explorer usa para a segurança TLS deve ser um arquivo Java keystore (JKS).
2. Inclua os certificados de CA para o banco de dados de chaves criado na etapa anterior.
3. Importe o certificado pessoal para o gerenciador de filas para o banco de dados de chave.
4. No Windows e Linux sistemas, inicie o IBM MQ Explorer usando o menu do sistema, o arquivo executável MQExplorer ou o comando **strmqcfcg**.
5. Na barra de ferramentas do IBM MQ Explorer, clique em **Janela -> Preferências** e, em seguida, expanda **IBM MQ Explorer** e clique em **Armazenamentos de certificados de clientes SSL**. Digite o nome e a senha para o arquivo JKS criado na etapa 1 de [“Tarefas no sistema que hospeda o IBM MQ Explorer”](#) na página 126, no Armazenamento de Certificados Confiáveis e Armazenamento de Certificados Pessoais e, em seguida, clique em **OK**.
6. Feche a janela **Preferências** e clique com o botão direito em **Gerenciadores de Filas**. Clique em **Mostrar/ocultar gerenciadores de filas** e, em seguida, clique em **Incluir** na tela **Mostrar/ocultar gerenciadores de filas**.
7. Digite o nome do gerenciador de filas e selecione o **Conectar Diretamente** a opção. Clique em Next.
8. Selecione **Utilizar tabela de definição de canal do cliente (CCDT)** e especifique o local do arquivo de tabela do canal que você transferiu do gerenciador de filas remotas na etapa 2 em [“Tarefas no sistema que hospeda o gerenciador de filas remotas”](#) na página 125 no sistema que hospeda o gerenciador de filas remotas.
9. Clique em **Finish**. É possível agora acessar o gerenciador de filas remotas do IBM MQ Explorer.

Conectando por meio de outro gerenciador de filas com o IBM MQ Explorer

O IBM MQ Explorer permite se conectar a um gerenciador de filas por meio de um gerenciador de filas intermediário, com o qual o IBM MQ Explorer já está conectado.

Neste caso, o IBM MQ Explorer coloca as mensagens de comando PCF no gerenciador de filas intermediário, especificando o seguinte:

- O parâmetro *ObjectQMgrName* no descritor de objetos (MQOD) como o nome do gerenciador de filas de destino. Para obter mais informações sobre a resolução de nome da fila, consulte [Resolução de Nome](#).
- O parâmetro *UserIdentifier* no descritor de mensagens (MQMD) como o ID do usuário local.

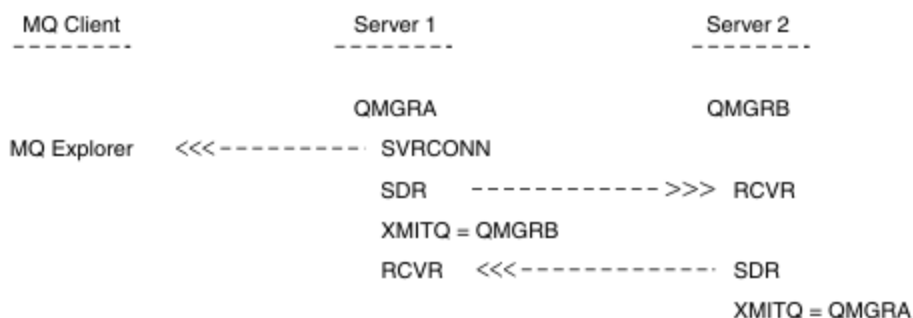
Se a conexão for, então, usada para conexão com o gerenciador de filas de destino por meio de um gerenciador de filas intermediário, o ID de usuário é transmitido no parâmetro *UserIdentifier* do descritor de mensagens (MQMD) novamente. Para que o listener MCA no gerenciador de filas de destino aceite essa mensagem, o atributo MCAUSER deve ser configurado ou o *userId* já deve existir com autoridade put.

O servidor de comandos no gerenciador de filas de destino coloca as mensagens para a fila de transmissão especificando o ID do usuário no parâmetro *UserIdentifier* no descritor de mensagens (MQMD). Para este put ser bem-sucedido, o ID do usuário já deve existir no gerenciador de filas de destino com autoridade put.

O seguinte exemplo mostra como se conectar a um gerenciador de filas, através de um gerenciador de filas intermediário, no IBM MQ Explorer.

Estabeleça uma conexão de administração remota para um gerenciador de filas. Verifique se o:

- O gerenciador de filas no servidor está ativo e tem um canal de conexão do servidor (SVRCONN) definido.
- O listener está ativo.
- O servidor de comandos está ativo.
- A fila SYSTEM.MQ EXPLORER.REPLY.MODEL foi criada e que você tenha autoridade suficiente.
- Os listeners do gerenciador de filas, servidores de comando e canais do emissor são iniciados.



Neste exemplo:

- IBM MQ Explorer está conectado ao gerenciador de filas QMGRB (em execução no Server1) usando uma conexão do cliente.
- O gerenciador de filas QMGRB no Server2 pode ser agora conectado ao IBM MQ Explorer por meio de um gerenciador de filas intermediário (QMGRB)
- Ao se conectar ao QMGRB com o IBM MQ Explorer, selecione QMGRB como o gerenciador de filas intermediário

Nesta situação, não há conexão direta ao QMGRB a partir do IBM MQ Explorer; a conexão com o QMGRB é por meio do QMGRB.

O gerenciador de filas QMGRB em Server2 é conectado ao QMGRB em Server1 utilizando os canais do emissor do receptor. O canal entre o QMGRB e QMGRB devem ser configurados de tal modo que a administração remota é possível; consulte [“Configurando Gerenciadores de Filas para Administração Remota”](#) na página 200.

Mostrando e ocultando gerenciadores de filas e clusters no IBM MQ Explorer

O IBM MQ Explorer pode exibir mais de um gerenciador de filas por vez. No painel Mostrar/Ocultar Gerenciador de Filas (selecionável a partir do menu para o nó da árvore Gerenciadores de Filas), é

possível escolher se você exibirá informações sobre outra máquina (remota). Os gerenciadores de filas locais são detectados automaticamente.

Para mostrar um gerenciador de filas remotas:

1. Clique com o botão direito do mouse no nó da árvore **Gerenciadores de Filas** e, em seguida, **Mostrar / Ocultar Gerenciadores de Filas**.
2. Clique em **Incluir**. O painel Mostrar/Ocultar Gerenciadores de Filas é exibido.
3. Digite o nome do gerenciador de filas remotas e o nome do host ou endereço IP nos campos fornecidos.

O nome do host ou o endereço IP é utilizado para estabelecer uma conexão do cliente com o gerenciador de filas remotas utilizando o seu canal de conexão do servidor padrão, SYSTEM.ADMIN.SVRCONN ou um servidor definido pelo usuário de conexão do canal.

4. Clique em **Finish**.

O painel Mostrar/Ocultar Gerenciadores de Filas também exibe uma lista de todos os gerenciadores de filas visíveis. É possível utilizar esse painel para ocultar gerenciadores de filas a partir da visualização de navegação.

Se o IBM MQ Explorer exibir um gerenciador de filas que é um membro de um cluster, o cluster será detectado e exibido automaticamente.

Para exportar a lista de gerenciadores de filas remotas a partir deste painel:

1. Feche o painel Mostrar/Ocultar Gerenciadores de Filas.
2. Clique com o botão direito no nó da árvore mais alto do **IBM MQ** na área de janela de Navegação do IBM MQ Explorer e, em seguida, selecione **Exportar IBM MQ Explorer Configurações**
3. Clique em **IBM MQ Explorer > IBM MQ Explorer Configurações**
4. Selecione **Informações de conexão > Gerenciadores de filas remotas**.
5. Selecione um arquivo para armazenar as configurações exportadas.
6. Finalmente, clique em **Concluir** para exportar as informações de conexão do gerenciador de filas remotas para o arquivo especificado.

Para importar uma lista de gerenciadores de filas remotas:

1. Clique com o botão direito no nó da árvore mais alto do **IBM MQ** na área de janela de Navegação do IBM MQ Explorer e, em seguida, selecione **Importar IBM MQ Explorer Configurações**
2. Clique em **IBM MQ Explorer > IBM MQ Explorer Configurações**
3. Clique em **Procurar** e navegue para o caminho do arquivo que contém as informações de conexão do gerenciador de filas remotas.
4. Clique em **Open**. Se o arquivo contiver uma lista de gerenciadores de filas remotas, a caixa **Informações de conexão > Gerenciadores de filas remotas** será selecionada.
5. Finalmente, clique em **Concluir** para importar as informações de conexão do gerenciador de filas remotas no IBM MQ Explorer.

Associação de cluster e o IBM MQ Explorer

IBM MQ Explorer requer informações sobre os gerenciadores de filas, membros de um cluster.

Se um gerenciador de filas for um membro de um cluster, então, o nó da árvore de clusters será preenchido automaticamente.

Se os gerenciadores de filas se tornarem membros de clusters enquanto o IBM MQ Explorer estiver em execução, você deverá manter o IBM MQ Explorer com dados de administração atualizados sobre clusters para que ele possa se comunicar efetivamente com eles e exibir informações de cluster corretas quando solicitado. Para fazer isso, o IBM MQ Explorer precisa das seguintes informações:

- O nome de um gerenciador de filas de repositório

- O nome de conexão do gerenciador de filas do repositório, se ele estiver em um gerenciador de filas remotas

Com essas informações, o IBM MQ Explorer pode:

- Utilize o gerenciador de filas do repositório para obter uma lista de gerenciadores de filas no cluster.
- Administrar os gerenciadores de filas que são membros do cluster e estão em plataformas suportadas e níveis de comando.

Administração não é possível se:

- O repositório escolhido se torna indisponível. O IBM MQ Explorer não alterna automaticamente para um repositório alternativo.
- O repositório escolhido não pode ser contactado através de TCP/IP.
- O repositório escolhido está em execução em um gerenciador de filas em execução em um nível de plataforma e de comando não suportado pelo IBM MQ Explorer.

Os membros de cluster que podem ser administrados podem ser locais ou podem ser remotos se eles podem ser contactados utilizando TCP/IP. O IBM MQ Explorer se conecta a gerenciadores de filas locais, membros de um cluster diretamente, sem usar uma conexão do cliente.

Conversão de dados para IBM MQ Explorer

O IBM MQ Explorer funciona em CCSID 1208 (UTF-8). Isso permite que o IBM MQ Explorer exiba os dados a partir de gerenciadores de filas corretamente. Quer esteja se conectando a um gerenciador de filas diretamente, quer usando um gerenciador de filas intermediário, o IBM MQ Explorer requer que todas as mensagens recebidas sejam convertidas para CCSID 1208 (UTF-8).

Uma mensagem de erro será emitida se você tentar estabelecer uma conexão entre o IBM MQ Explorer e um gerenciador de filas com um CCSID que o IBM MQ Explorer não reconhece.

Conversões suportadas são descritas em [Conversão da Página de Códigos](#).

Windows

Usando o aplicativo IBM MQ Taskbar (Windows somente)

O aplicativo IBM MQ Taskbar exibe um ícone na bandeja do sistema Windows no servidor. O ícone fornece a você o status atual do IBM MQ e um menu a partir do qual é possível executar algumas ações simples.

No Windows, o ícone IBM MQ está na bandeja do sistema no servidor e é sobreposto por um símbolo de status codificado, que pode ter um dos seguintes significados:

Linhas

Trabalhando corretamente; nenhum alerta no presente

Blue

Indeterminado; IBM MQ está sendo inicializado ou encerrado

Amarela

Alerta; um ou mais serviços estão falhando ou já falharam

Para exibir o menu, clique com o botão direito no ícone do IBM MQ. No menu, é possível executar as seguintes ações:

- Clique em **Abrir** para abrir o IBM MQ Alert Monitor.
- Clique em **Sair** para sair do aplicativo IBM MQ Taskbar.
- Clique em **IBM MQ Explorer** para iniciar o IBM MQ Explorer.
- Clique em **Parar IBM MQ** para parar o IBM MQ.
- Clique em **Sobre o IBM MQ** para exibir informações sobre o IBM MQ Alert Monitor.

O aplicativo monitor de alertas do IBM MQ (Windows somente)

O monitor de alertas do IBM MQ é uma ferramenta de detecção de erro que identifica e registra problemas com o IBM MQ em uma máquina local.

O monitor de alertas exibe informações sobre o status atual da instalação local de um servidor do IBM MQ. Também monitora o Windows Advanced Configuration and Power Interface (ACPI) e assegura que as configurações do ACPI sejam impostas.

A partir do monitor de alertas do IBM MQ, é possível:

- Acessar o IBM MQ Explorer diretamente
- Visualizar as informações relacionadas a todos os alertas pendentes
- Encerrar o serviço IBM MQ na máquina local
- Rotear mensagens de alerta pela rede para uma conta do usuário configurável ou para uma estação de trabalho ou servidor Windows

Trabalhando com objetos locais do IBM MQ

É possível administrar objetos locais do IBM MQ para suportar programas de aplicativos que usam o Message Queue Interface (MQI).

Sobre esta tarefa

Nesse contexto, a administração local significa criar, exibir, mudar, copiar e excluir os objetos do IBM MQ.

Além das abordagens descritas nesta seção, é possível usar o IBM MQ Explorer para administrar objetos locais do IBM MQ. Para obter informações adicionais, consulte [“Administração usando o IBM MQ Explorer”](#) na página 121.

Procedimento

- Use as informações nos tópicos a seguir para ajudá-lo com a administração de objetos locais do IBM MQ.
 - [Programas de aplicativo usando o MQI](#)
 - [“Administrando o IBM MQ usando comandos MQSC”](#) na página 12
 - [“Exibindo e alterando atributos do gerenciador de filas”](#) na página 138
 - [“Trabalhando com Filas Locais”](#) na página 141
 - [“Trabalhando com Filas de Alias”](#) na página 154
 - [“Trabalhando com filas modelo”](#) na página 156
 - [“Trabalhando com Serviços”](#) na página 186
 - [“Gerenciando os Objetos para Acionamento”](#) na página 193

Trabalhando com Gerenciadores de Fila

É possível usar comandos de controle para iniciar e parar um gerenciador de filas. É possível usar comandos MQSC para exibir ou alterar os atributos do gerenciador de filas.

Tarefas relacionadas

[Criando gerenciadores de filas em Multiplataformas](#)

Iniciando um Gerenciador de Filas

Quando você cria um gerenciador de filas, deve iniciá-lo para permitir que ele processe comandos ou chamadas MQI.

Sobre esta tarefa

É possível iniciar um gerenciador de filas usando o comando **strmqm**. Para uma descrição do comando **strmqm** e suas opções, veja [strmqm](#).

Windows **Linux** Como alternativa, em sistemas Windows e Linux (plataformas x86 e x86-64), é possível iniciar um gerenciador de filas usando o IBM MQ Explorer.

Windows No Windows, é possível iniciar um gerenciador de filas automaticamente quando o sistema começa a usar o IBM MQ Explorer. Para obter informações adicionais, consulte [“Administração usando o IBM MQ Explorer”](#) na página 121.

Procedimento

- Para iniciar um gerenciador de filas usando o comando **strmqm**, insira o comando seguido pelo nome do gerenciador de filas que você deseja iniciar.

Por exemplo, para iniciar um gerenciador de filas chamado QMB, insira o comando a seguir:

```
strmqm QMB
```

Nota: Deve-se usar o comando **strmqm** a partir da instalação associada ao gerenciador de filas com o qual você está trabalhando. É possível descobrir a qual instalação um gerenciador de filas está associado usando o comando `dspmqr -o installation`

O comando **strmqm** não retorna o controle até que o gerenciador de filas tenha sido iniciado e esteja pronto para aceitar solicitações de conexão.

- Windows** **Linux**
 - Para iniciar um gerenciador de filas usando o IBM MQ Explorer, conclua as etapas a seguir:
 - Abra o IBM MQ Explorer.
 - Na visualização do Navegador, selecione o gerenciador de filas.
 - Clique em **Iniciar**.

Resultados

O gerenciador de filas é iniciado.

Se a inicialização do gerenciador de filas levar mais de alguns segundos, o IBM MQ emite mensagens de informações intermitentemente detalhando o processo de inicialização.

Multi Parando um Gerenciador de Filas

É possível usar o comando **endmqm** para parar um gerenciador de filas. Esse comando fornece quatro maneiras de parar um gerenciador de filas: um encerramento controlado ou em modo quiesce, um encerramento imediato, um encerramento preventivo e um encerramento em espera. Como alternativa, no Windows e Linux, é possível parar um gerenciador de filas usando o IBM MQ Explorer.

Sobre esta tarefa

Há quatro maneiras de parar um gerenciador de filas de instância única com o comando **endmqm**:

Encerramento controlado (em modo quiesce)

Por padrão, o comando **endmqm** executa um encerramento em modo quiesce do gerenciador de filas especificado. Um encerramento em modo quiesce aguarda até que todos os aplicativos conectados tenham sido desconectados, portanto pode levar algum tempo para ser concluído.

Encerramento Imediato

Para um encerramento imediato, quaisquer chamadas MQI atuais têm permissão para concluir, mas as chamadas novas falham. Esse tipo de encerramento não aguarda os aplicativos se desconectarem do gerenciador de filas.

Encerramento Preemptivo

O gerenciador de filas é parado imediatamente. Use este tipo de encerramento somente em circunstâncias excepcionais, por exemplo, quando um gerenciador de filas não é parado como resultado de um comando **endmqm** normal.

Aguardar encerramento

Este tipo de encerramento é equivalente a um encerramento controlado, exceto se o controle for retornado a você somente depois que o gerenciador de filas tiver sido parado.

O comando **endmqm** para todas as instâncias de um gerenciador de filas de várias instâncias da mesma maneira que para um gerenciador de filas de instância única. É possível emitir **endmqm** na emitir ou uma das instâncias em espera de um gerenciador de filas de várias instâncias. No entanto, deve-se emitir **endmqm** na instância ativa para finalizar o gerenciador de filas.

Você tem a opção de terminar o gerenciador de fila em um tempo de destino de um número de segundos especificado, com ou sem interrupção de tarefas de manutenção do gerenciador de filas não essenciais, consulte [“Finalizando um gerenciador de filas em um tempo de destino” na página 134.](#)



Atenção:

- As mensagens persistentes persistirão independentemente do tipo de encerramento usado (incluindo o encerramento manual de processos IBM MQ), enquanto as mensagens não persistentes não podem ter garantia de sobreviver a qualquer tipo de encerramento.

Especificar a propriedade da fila NPMCLASS (HIGH) salva mensagens não persistentes em uma base melhor que pode ser feita. O uso de processos **endmqm -t**, **endmqm -tp**, **endmqm -pou** o término manual de IBM MQ reduz as chances de mensagens NPMCLASS (HIGH) sobreviverem a um ciclo de encerramento ou reinicialização do IBM MQ em comparação com **endmqm -w** ou **endmqm -i**

- O tempo combinado para terminar e reiniciar o gerenciador de filas pode ser mais longo como resultado do uso de um método de encerramento mais abrupto, particularmente ao usar as opções **-p** e **-tp**

Se o gerenciador de filas precisar recorrer ao término dos processos do IBM MQ para terminar o gerenciador de filas, provavelmente será necessária mais reconciliação do estado do gerenciador de filas quando o gerenciador de filas for reiniciado.

Para obter uma descrição detalhada do comando **endmqm** e suas opções, consulte [endmqm](#).

Sugestão: Problemas com o encerramento de um gerenciador de filas são frequentemente causados por aplicativos. Por exemplo, quando aplicativos:

- Não verificam códigos de retorno de MQI corretamente
- Não solicitam notificação de um quiesce
- São finalizados sem se desconectarem do gerenciador de filas (emitindo uma chamada MQDISC)

Se um problema ocorre quando você tenta parar o gerenciador de filas, é possível sair do comando **endmqm** usando Ctrl-C. É possível emitir outro comando **endmqm**, mas dessa vez com um parâmetro que especifica o tipo de encerramento requerido.



Como uma alternativa ao uso do comando **endmqm**, no Windows e Linux, é possível parar um gerenciador de filas usando o IBM MQ Explorer para realizar um encerramento controlado ou imediato.

Procedimento

- Para parar o gerenciador de filas usando o comando **endmqm**, insira o comando seguido pelo parâmetro apropriado, se necessário, e o nome do gerenciador de filas que você deseja para parar.

Nota: Deve-se usar o comando **endmqm** a partir da instalação associada ao gerenciador de filas com o qual você está trabalhando. Para descobrir com qual instalação um gerenciador de filas está associado, use o comando **dspmqr**:

```
dspmqr -o installation
```

- Para realizar um encerramento controlado (em modo quiesce), insira o comando **endmqm** conforme mostrado no exemplo a seguir, que para um gerenciador de filas chamado QMB:

```
endmqm QMB
```

Como alternativa, inserir o comando **endmqm** com o parâmetro **-c**, conforme mostrado no exemplo a seguir, é equivalente a um comando `endmqm QMB`.

```
endmqm -c QMB
```

Em ambos os casos, o controle é retornado a você imediatamente e você não é notificado quando o gerenciador de filas tiver parado. Se deseja que o comando aguarde até que todos os aplicativos tenham sido interrompidos e o gerenciador de filas tenha terminado antes de retornar o controle para você, use o parâmetro **-w** conforme mostrado no exemplo a seguir.

```
endmqm -w QMB
```

- Para realizar um encerramento imediato, insira o comando **endmqm** com o parâmetro **-i** conforme mostrado no exemplo a seguir:

```
endmqm -i QMB
```

- Para realizar um encerramento preemptivo, insira o comando **endmqm** com o parâmetro **-p** conforme mostrado no exemplo a seguir:

```
endmqm -p QMB
```



Atenção: Um encerramento preemptivo pode ter consequências imprevisíveis para aplicativos conectados. Não use essa opção, a menos que todas as outras tentativas de parar o gerenciador de filas usando um comando **endmqm** normal tenham falhado.



Se o encerramento preemptivo não funcionar, tente [“Parando um gerenciador de filas manualmente”](#) na página 135 como alternativa.

- Para solicitar a reconexão do cliente automática, insira o comando **endmqm** com o parâmetro **-r**. Este parâmetro tem o efeito de restabelecer a conectividade de clientes para outros gerenciadores de fila em seus grupos de gerenciadores de filas.

Nota: Encerrar um gerenciador de filas usando o comando padrão **endmqm** não aciona a reconexão do cliente automática.

- Para transferir para uma instância em espera de um gerenciador de filas de várias instâncias depois de encerrar a instância ativa, insira o comando **endmqm** com o parâmetro **-s** na instância ativa do gerenciador de filas de várias instâncias.
- Para encerrar a instância de espera de um gerenciador de filas de várias instâncias e deixar a instância ativa em execução, insira o comando **endmqm** com o parâmetro **-x** na instância em espera do gerenciador de filas de várias instâncias.



No Windows e Linux, para parar o gerenciador de filas usando IBM MQ Explorer, conclua as etapas a seguir:

- a) Abra o IBM MQ Explorer.
- b) Selecione o gerenciador de filas da Visualização do Navegador.

- c) Clique em **Parar**.
O painel **Terminar gerenciador de filas** é exibido.
- d) Selecione **Controlado** ou **Imediato**.
- e) Clique em **OK**.
O gerenciador de filas é parado.

Tarefas relacionadas

[Aplicando atualizações de nível de manutenção para gerenciadores de filas de várias instâncias no AIX](#)
[Aplicando atualizações de nível de manutenção para gerenciadores de filas de várias instâncias no Linux](#)
[Aplicando atualizações de nível de manutenção para gerenciadores de filas de várias instâncias no Windows](#)

Referências relacionadas

[endmqm \(terminar gerenciador de filas\)](#)

Finalizando um gerenciador de filas em um tempo de destino

É possível encerrar o gerenciador de filas dentro de um tempo de destino de um número de segundos que você especificar, com ou sem interromper tarefas de manutenção do gerenciador de filas não essenciais

Há duas maneiras de especificar um horário de destino quando você usa o comando **endmqm**. A opção **-t** permite que todas as tarefas de manutenção do gerenciador de filas sejam concluídas, o que pode prolongar a fase do término do gerenciador de filas. A opção **-tp** interrompe as tarefas de manutenção do gerenciador de filas não essenciais, se necessário, para obedecer ao tempo de destino especificado.

Tarefas de manutenção não essenciais incluem: compactação de arquivo de fila e mensagens NPMCLASS (HIGH) persistentes. No restante desta página, a palavra "limpeza" é usada.

Dependendo de seus padrões de uso do aplicativo, a compactação do arquivo de fila pode levar um longo tempo, portanto, se seu objetivo principal for terminar o gerenciador de filas rapidamente, use a opção **-tp**.

Quando você especifica um horário de destino, o tipo de encerramento de **-w**, **-i** ou **-p** indica o tipo de encerramento de início.

Nota: Um encerramento **immediate** ainda é ordenado, diferente de um encerramento **controlled**, principalmente pela maneira com que todos os aplicativos em execução são colocados em modo quiesce. Um encerramento **immediate** ainda executa a manutenção. Um encerramento limitado por tempo encerra essas ações quando elas interferem no cumprimento do horário de destino.

O gerenciador de filas intensifica o tipo de encerramento conforme necessário, em uma tentativa de atender o horário de destino. Por exemplo:

- Um destino **-t** de 10 segundos iniciando em **-w** pode ser de sete segundos sendo colocado em modo quiesce, dois segundos de encerramento imediato do gerenciador de filas, incluindo manutenção e, em seguida, encerramento imediato sem manutenção adicional:

```
endmqm -w -t 10 queue_manager
```

- Um destino **-tp** de 10 segundos pode ser de sete segundos sendo colocado em modo quiesce, dois segundos de encerramento imediato do gerenciador de filas, incluindo a manutenção, um segundo de encerramento imediato sem manutenção adicional e, em seguida, iniciar a finalização de processos do IBM MQ:

```
endmqm -c -tp 10 queue_manager
```

- Um destino **-tp** de dois segundos em **-i** pode ser um segundo de encerramento imediato do gerenciador de filas, incluindo a manutenção, um segundo de encerramento imediato sem manutenção adicional e, em seguida, iniciar a finalização de processos do IBM MQ:

```
endmqm -i -tp 2 queue_manager
```

- Um destino de um segundo em **-w** poderia ser de 0,1 segundos em `wait`, por exemplo, apenas o suficiente para enviar os códigos de retorno do IBM MQ para aplicativos conectados, 0,9 segundos de encerramento imediato do gerenciador de filas, incluindo a manutenção e, em seguida, encerramento imediato sem manutenção adicional; depois, iniciar a finalização de processos do IBM MQ.

Referências relacionadas

[endmqm \(terminar gerenciador de filas\)](#)



Parando um gerenciador de filas manualmente

Se os métodos padrão para parar e remover um gerenciador de filas falham, é possível tentar parar o gerenciador de filas manualmente.

Sobre esta tarefa

A maneira padrão de parar os gerenciadores de filas é usando o comando **endmqm**, conforme descrito em “Parando um Gerenciador de Filas” na página 131. Se não for possível parar um gerenciador de filas no modo padrão, será possível tentar parar um gerenciador de filas manualmente. O modo como isso é feito depende de qual plataforma você está usando.

Procedimento

-  Para parar um gerenciador de filas no Windows, veja “Parando um gerenciador de filas manualmente no Windows” na página 135.
-  Para parar um gerenciador de filas no AIX ou Linux, veja “Parando um gerenciador de filas manualmente no AIX and Linux” na página 136.

Tarefas relacionadas

[Criando e gerenciando gerenciadores de filas em Multiplataformas](#)

Referências relacionadas

[endmqm](#)

Parando um gerenciador de filas manualmente no Windows

Se não for possível parar um gerenciador de filas no Windows usando o comando **endmqm**, será possível tentar parar o gerenciador de filas manualmente terminando quaisquer processos que estão em execução e parando o serviço IBM MQ.

Sobre esta tarefa

Sugestão: O Windows Gerenciador de Tarefas e o comando **tasklist** fornecem informações limitadas sobre as tarefas. Para obter mais informações para ajudar a determinar quais processos se relacionam com um determinado gerenciador de filas, considere utilizar uma ferramenta como *Process Explorer* (`procexp.exe`), que está disponível para download no website do Microsoft em <http://www.microsoft.com>.

Para parar um gerenciador de filas no Windows, conclua as etapas a seguir.

Procedimento

1. Liste os nomes (IDs) dos processos que estão em execução, usando o Windows Gerenciador de Tarefas.
2. Encerre os processos utilizando o Windows Gerenciador de Tarefas ou o comando **taskkill**, na seguinte ordem (se eles estiverem em execução):

Tabela 7. Processos do Windows a serem interrompidos se em execução

Nome do processo	Descrição
AMQZMUC0	Gerenciador de processos crítico
AMQZXMA0	Controlador de execução
AMQZFUMA	Processo OAM
AMQZLAA0	Agentes LQM
AMQZLSA0	Agentes LQM
AMQZMUFO	Gerenciador de Utilitários
AMQZMGR0	Controlador de Processos
AMQZMUR0	Gerenciador de processos reinicializável
AMQFQPUB	Processo de publicação/assinatura
AMQFCXBA	Processo do trabalhador do broker
AMQRMPPA	Processo do conjunto de processos
AMQCRSTA	Processo de tarefa do respondente não encadeado
AMQCRS6B	Canal receptor da LU62 e conexão do cliente
AMQRRMFA	Processo do repositório (para clusters)
AMQPCSEA	O servidor de comandos
RUNMQTRM	Chamar um monitor acionador para um servidor
RUNMQDLQ	Chamar manipulador da fila de devoluções
RUNMQCHI	Processo do inicializador de canais
RUNMQLSR	Processo do listener de canal
AMQXSSVN	Servidores de memória compartilhada

3. Pare o serviço do IBM MQ em **Ferramentas de administração > Serviços** no Windows Painel de Controle.
4. Se você tiver tentado todos os métodos e o gerenciador de filas não foi interrompido, reinicialize o sistema.

Parando um gerenciador de filas manualmente no AIX and Linux

Se você não conseguir parar um gerenciador de filas no AIX ou no Linux usando o comando **endmqm**, será possível tentar parar o gerenciador de filas manualmente finalizando quaisquer processos que estejam em execução e parando o serviço IBM MQ.

Sobre esta tarefa

Para parar um gerenciador de filas no AIX ou no Linux, conclua as etapas a seguir.

Se você parar o gerenciador de filas manualmente, o FFST poderá ser obtido e os arquivos FDC colocados em `/var/mqm/errors`. Isso não deve ser considerado como um defeito no gerenciador de filas.

O gerenciador de filas será reiniciado normalmente, mesmo após você tê-lo interrompido usando esse método de parar manualmente.

Procedimento

1. Localize os identificadores de processo (PIDs) dos programas do Gerenciador de Filas que ainda estão em execução usando o comando **ps** .

Por exemplo, se o gerenciador de filas é chamado QMNAME, use o seguinte comando:

```
ps -ef | grep QMNAME
```

2. Encerre quaisquer processos do gerenciador de filas que ainda estejam em execução usando o comando **kill** , especificando os PIDs descobertos usando o comando **ps**

Para terminar um processo, use **kill -KILL <pid>** ou o comando **kill -9 <pid>** equivalente.

Você tem que trabalhar através dos PIDs que você quer matar, um por um, emitindo esse comando cada vez.

Importante: Se você usar qualquer sinal diferente de **9 (SIGKILL)** , o processo provavelmente não será interrompido e você obterá resultados imprevisíveis..

Encerre os processos na seguinte ordem:

Nome do processo	Descrição
amqzmuc0	Gerenciador de processos crítico
amqzma0	Controlador de execução
amqzfuma	Processo OAM
amqzlaa0	Agentes LQM
amqzlsa0	Agentes LQM
amqzmuf0	Gerenciador de Utilitários
amqzmur0	Gerenciador de processos reinicializável
amqzmgr0	Controlador de Processos
amqfpub	Processo de publicação/assinatura
amqfcxba	Processo do trabalhador do broker
amqrmppa	Processo do conjunto de processos
amqcrsta	Processo de tarefa do respondente não encadeado
amqcrs6b	Canal receptor da LU62 e conexão do cliente
amqrrmfa	Processo do repositório (para clusters)
amqpcsea	O servidor de comandos
runmqtrm	Chamar um monitor acionador para um servidor
runmqdlq	Chamar manipulador da fila de devoluções
runmqchi	Processo do inicializador de canais
runmqlsr	Processo do listener de canal

Tarefas relacionadas

“Parando um Gerenciador de Filas” na página 131

É possível usar o comando **endmqm** para parar um gerenciador de filas. Esse comando fornece quatro maneiras de parar um gerenciador de filas: um encerramento controlado ou em modo quiesce, um

encerramento imediato, um encerramento preventivo e um encerramento em espera. Como alternativa, no Windows e Linux, é possível parar um gerenciador de filas usando o IBM MQ Explorer.

Multi Reiniciando um Gerenciador de Filas

É possível usar o comando **strmqm** para reiniciar um gerenciador de filas ou, em sistemas Windows e Linux x86-64, é possível reiniciar um gerenciador de filas no IBM MQ Explorer.

Sobre esta tarefa

É possível reiniciar um gerenciador de filas usando o comando **strmqm**. Para uma descrição do comando **strmqm** e suas opções, veja [strmqm](#).

Windows **Linux** Em sistemas Windows e Linux x86-64, é possível reiniciar um gerenciador de filas usando o IBM MQ Explorer da mesma maneira que para iniciar um gerenciador de filas.

Procedimento

- Para reiniciar um gerenciador de filas usando o comando **strmqm**, insira o comando seguido pelo nome do gerenciador de filas que você deseja reiniciar.
Por exemplo, para iniciar um gerenciador de filas chamado `strmqm saturn.queue.manager`, insira o comando a seguir:

```
strmqm saturn.queue.manager
```

- Windows** **Linux**
Para iniciar um gerenciador de filas usando o IBM MQ Explorer, conclua as etapas a seguir:
 - Abra o IBM MQ Explorer.
 - Na visualização do Navegador, selecione o gerenciador de filas.
 - Clique em **Iniciar**.

Resultados

O gerenciador de filas é reiniciado.

Se a reinicialização do gerenciador de filas levar mais do que alguns segundos, o IBM MQ emitirá mensagens de informação intermitentes detalhando o progresso da inicialização.

Exibindo e alterando atributos do gerenciador de filas

Use o comando MQSC **DISPLAY QMGR** para exibir os parâmetros do Gerenciador de Filas para um gerenciador de fila. Use o comando MQSC **ALTER QMGR** para alterar os parâmetros do gerenciador de filas para um gerenciador de fila local..

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Procedimento

- Para exibir os atributos do gerenciador de filas especificado no comando **runmqsc**, use o comando do MQSC **DISPLAY QMGR**:

```
DISPLAY QMGR
```

O exemplo a seguir mostra a saída típica desse comando:

```
DISPLAY QMGR
  1 : DISPLAY QMGR
AMQ8408I: Display Queue Manager details.
QMNAME(QM1) ACCTCONO(DISABLED)
ACCTINT(1800) ACCTMQI(OFF)
ACCTQ(OFF) ACTIVREC(MSG)
ACTVCONO(DISABLED) ACTVTRC(OFF)
ADVCAP(DISABLED) ALTDATE(2022-05-05)
ALTTIME(14.24.34) AMQPCAP(NO)
AUTHOREV(DISABLED) CCSID(437)
CERTLABL(ibmwebsphereqm1) CERTVPOL(ANY)
CHAD(DISABLED) CHADEV(DISABLED)
CHADEXIT( ) CHLEV(DISABLED)
CHLAUTH(ENABLED) CLWLDATA( )
CLWLEXIT( ) CLWLLEN(100)
CLWLMRUC(999999999) CLWLUSEQ(LOCAL)
CMDEV(DISABLED) CMDLEVEL(930)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) CONFIGEV(DISABLED)
CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
CRDATE(2020-12-22) CRTIME(15.42.49)
CUSTOM( ) DEADQ( )
DEFCLXQ(SCTQ) DEFXMITQ( )
DESCR( ) DISTL(YES)
IMGINTVL(60) IMGLOGLN(OFF)
IMGRCOVO(YES) IMGRCOVQ(YES)
IMGSCHEM(MANUAL) INHIBTEV(DISABLED)
INITKEY( ) IPADDRV(IPV4)
LOCALEV(DISABLED) LOGGEREV(DISABLED)
MARKINT(5000) MAXHANDS(256)
MAXMSG(4194304) MAXPROPL(NOLIMIT)
MAXPRTY(9) MAXUMSGS(10000)
MONACLS(QMGR) MONCHL(OFF)
MONQ(OFF) PARENT( )
PERFMEV(DISABLED) PLATFORM(WINDOWS10)
PSMODE(ENABLED) PSCLUS(ENABLED)
PSNPMMSG(DISCARD) PSNPRES(NORMAL)
PSRTYCNT(5) PSSYNCP(1FPER)
QMID(QM1_2020-12-22_15.42.49) REMOTEEV(DISABLED)
REPOS( ) REPOSNL( )
REVDNS(ENABLED) ROUTEREC(MSG)
SCHINIT(QMGR) SCMDSERV(QMGR)
SPLCAP(DISABLED) SSLCRLNL( )
SSLCRYP( ) SSLEV(DISABLED)
SSLFIPS(NO) KEYRPPWD( )
SSLKEYR(C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key)
SSLRKEYC(32767) STATACLS(QMGR)
STATCHL(OFF) STATINT(1800)
STATMQI(OFF) STATQ(OFF)
STRSTPEV(ENABLED) SUITEB(NONE)
SYNCPT TREELIFE(1800)
TRIGINT(999999999) VERSION(09030000)
XRCAP(NO)
```

Nota: SYNCPT é um atributo somente leitura do gerenciador de filas.

O parâmetro **ALL** é o padrão no comando **DISPLAY QMGR**. Ele exibe todos os atributos do gerenciador de filas. Em particular, a saída informa o nome do gerenciador de filas padrão, o nome da fila de devoluções e o nome da fila de comandos.

É possível confirmar se essas filas existem, inserindo o comando:

```
DISPLAY QUEUE (SYSTEM.*)
```

Isso exibe uma lista de filas que correspondem à raiz **SYSTEM.***. Os parênteses são necessários.

- Para alterar os atributos do gerenciador de filas especificado no comando **runmqsc**, use o comando **ALTER QMGR** do MQSC, especificando os atributos e valores que você deseja alterar.

Por exemplo, use os seguintes comandos para alterar os atributos de **jupiter.queue.manager**:

```
runmqsc jupiter.queue.manager
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

O comando **ALTER QMGR** altera a fila de devoluções usada e permite eventos de inibição.

Parâmetros não especificados no comando **ALTER QMGR** resultam nos valores existentes para esses parâmetros sendo deixados inalterados.

Tarefas relacionadas

[Criando gerenciadores de filas em Multiplataformas](#)

Referências relacionadas

[Atributos do gerenciador de filas](#)

[runmqsc \(executar comandos MQSC\)](#)

[DISPLAY QMGR](#)

[ALTER QMGR](#)

Multi Excluindo um Gerenciador de Filas

É possível excluir um gerenciador de filas usando o comando de controle **dltmqm**. Como alternativa, em sistemas Windows e Linux, é possível usar o IBM MQ Explorer para excluir um gerenciador de filas.

Antes de começar



Atenção:

- A exclusão de um gerenciador de filas é uma etapa drástica, pois você exclui também todos os recursos associados ao gerenciador de filas, incluindo todas as filas e suas mensagens e todas as definições de objetos. Se você usar o comando de controle **dltmqm**, não haverá prompt exibido que permita mudar de ideia; ao pressionar a tecla Enter, todos os recursos associados serão perdidos.
- **Windows** No Windows, excluir um gerenciador de filas também remove o gerenciador de filas da lista de inicialização automática (descrita em “Iniciando um Gerenciador de Filas” na página 130). Quando o comando tiver sido concluído, uma mensagem IBM MQ `queue manager ending` será exibida; você não será informado de que o gerenciador de fila foi excluído.
- A exclusão de um gerenciador de filas do cluster não o remove do cluster. Para obter mais informações, veja as notas de uso em [dltmqm](#).

Sobre esta tarefa

É possível excluir um gerenciador de filas usando o comando **dltmqm control**. Para obter uma descrição do comando **dltmqm** e suas opções, consulte [dltmqm](#). Assegure-se de que apenas administradores confiáveis tenham autoridade para usar esse comando. (Para obter informações sobre segurança, consulte [Configurando a segurança no AIX, Linux, and Windows](#).)


Windows **Linux** Como alternativa, em sistemas Windows e Linux (plataformas x86 e x86-64), é possível excluir um gerenciador de filas usando o IBM MQ Explorer.

Procedimento

- Para excluir um gerenciador de filas usando o comando **dltmqm**, conclua as etapas a seguir:
 - a) Parar o gerenciador de fila.
 - b) Emita o seguinte comando:

```
dltmqm QMB
```

Nota: Deve-se usar o comando **dltmqm** da instalação associada ao gerenciador de filas com o qual está trabalhando. É possível descobrir com qual instalação um gerenciador de filas está associado usando o comando `dspmqr -o installation`.

-  

Para excluir um gerenciador de filas usando o IBM MQ Explorer, conclua as etapas a seguir:

- a) Abra o IBM MQ Explorer.
- b) Na visualização do Navegador, selecione o gerenciador de filas.
- c) Se o gerenciador de filas não estiver interrompido, pare-o.
Para parar o gerenciador de filas, clique com o botão direito nele e, em seguida, clique em **Parar**.
- d) Exclua o gerenciador de filas.
Para excluir o gerenciador de filas, clique com o botão direito nele e, em seguida, clique em **Excluir**.

Resultados

O gerenciador de filas é excluído.

Parando canais MQI

Ao emitir um comando STOP CHANNEL em um canal de conexão do servidor, é possível escolher qual método usar para parar o canal de conexão do cliente. Isto significa que um canal do cliente que emite uma chamada de espera MQGET pode ser controlado e é possível decidir como e quando parar o canal.

O comando STOP CHANNEL pode ser emitido com três modos, indicando como o canal deve ser parado:

Modo Quiesce

Para o canal após qualquer mensagem atual ter sido processada.

Se as conversações de compartilhamento estiverem ativadas, o IBM MQ MQI client se tornará ciente da solicitação de parada em tempo hábil; este tempo é dependente da velocidade da rede.

O aplicativo cliente se torna ciente da solicitação de parada como resultado da emissão de uma chamada subsequente para IBM MQ.

Forçar

Para o canal imediatamente.

Terminar

Para o canal imediatamente. Se o canal estiver em execução como um processo, ele pode finalizar o processo do canal ou, se o canal estiver em execução como um encadeamento, seu encadeamento.

Este é um processo de vários estágios. Se o modo terminate for usado, será feita uma tentativa de parar o canal de conexão do servidor, primeiro com o modo quiesce, em seguida, com o modo force e, se necessário, com o modo terminate. O cliente pode receber diferentes códigos de retorno durante os diferentes estágios da finalização. Se o processo ou encadeamento for finalizado, o cliente receberá um erro de comunicação.

Os códigos de retorno retornados ao aplicativo variam de acordo com a chamada MQI emitida e o comando STOP CHANNEL emitido. O cliente receberá um código de retorno MQRC_CONNECTION_QUIESCING ou MQRC_CONNECTION_BROKEN. Se um cliente detectar MQRC_CONNECTION_QUIESCING, ele deverá tentar concluir a transação atual e finalizar. Isto não é possível com MQRC_CONNECTION_BROKEN. Se o cliente não concluir a transação e finalizar rápido o suficiente, ele receberá CONNECTION_BROKEN após alguns segundos. Um comando STOP CHANNEL com MODE(FORCE) ou MODE(TERMINATE) é mais provável de resultar em um CONNECTION_BROKEN do que com MODE(QUIESCE).

Conceitos relacionados

[Canais](#)

Trabalhando com Filas Locais

Esta seção contém exemplos de alguns comandos MQSC que podem ser usados para gerenciar filas locais, de modelo e de alias.

Consulte [Comandos MQSC](#) para obter informações detalhadas sobre esses comandos.

Referências relacionadas

[Restrições de nomenclatura para filas](#)

[Restrições de nomenclatura para outros objetos](#)

Definindo uma fila local com DEFINE QLOCAL

Para um aplicativo, o gerenciador de filas locais é o gerenciador de filas ao qual o aplicativo está conectado. As filas gerenciadas pelo gerenciador de filas locais são consideradas como locais para esse gerenciador de filas. Use o comando MQSC **DEFINE QLOCAL** para criar uma fila local.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Sobre esta tarefa

Use o comando MQSC **DEFINE QLOCAL** para criar uma fila local. Também é possível utilizar o padrão definido na definição de fila local padrão ou é possível modificar as características da fila a partir daqueles da fila local padrão.

Nota: A fila local padrão é denominada SYSTEM.DEFAULT.LOCAL.QUEUE e é criada na instalação do sistema.

Procedimento

- Para criar uma fila local, insira o comando **DEFINE QLOCAL**, conforme mostrado no exemplo a seguir. Neste exemplo, o comando **DEFINE QLOCAL** define uma fila chamada ORANGE.LOCAL.QUEUE com estas características:
 - Ele é ativado para gets, ativada para entradas e opera em uma base ordem de prioridade.
 - É uma fila *normal*; ela não é uma fila de inicialização ou fila de transmissão e não gera mensagens do acionador.
 - A profundidade da fila máxima é 5000 mensagens; o comprimento máximo da mensagem é 4194304 bytes.

```
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT(ENABLED) +
  GET(ENABLED) +
  NOTRIGGER +
  MSGDLVSQ(PRIORITY) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  USAGE(NORMAL)
```

Notas:

1. Com exceção do valor para a descrição, todos os valores de atributos mostrados no exemplo são os valores padrão. Esses exemplos são incluídos para propósitos de ilustração. É possível omiti-los se você estiver certo de que os padrões são o que você deseja ou não foram mudados. Consulte também [“Exibindo atributos do objeto padrão com DISPLAY QUEUE”](#) na página 143.
2. **USAGE(NORMAL)** indica que esta fila não é uma fila de transmissão.
3. Se você já tiver uma fila local no mesmo gerenciador de filas com o nome ORANGE.LOCAL.QUEUE, esse comando falhará. Utilize o atributo **REPLACE** se você deseja sobrescrever a definição existente de uma fila, mas consulte também [“Mudando os atributos da fila local com ALTER QLOCAL ou DEFINE QLOCAL”](#) na página 144.

Referências relacionadas

[DEFINE QLOCAL](#)

Exibindo atributos do objeto padrão com DISPLAY QUEUE

É possível usar o comando MQSC **DISPLAY QUEUE** para exibir atributos que foram obtidos do objeto padrão quando um objeto IBM MQ foi definido..

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#) .

Sobre esta tarefa

Ao definir um objeto do IBM MQ, leva quaisquer atributos que você não especificar a partir do objeto padrão. Por exemplo, ao definir uma fila local, a fila herda quaisquer atributos que você omitir na definição da fila local padrão, que é denominada SYSTEM.DEFAULT.LOCAL.QUEUE. É possível usar o comando **DISPLAY QUEUE** para ver exatamente quais são esses atributos.

Procedimento

- Para exibir os atributos do objeto padrão de uma fila local, use o comando a seguir:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

A sintaxe do comando **DISPLAY** é diferente daquela do comando **DEFINE** correspondente. No comando **DISPLAY**, é possível fornecer apenas o nome da fila, enquanto, no comando **DEFINE**, você precisa especificar o tipo da fila, ou seja, QLOCAL, QALIAS, QMODEL ou QREMOTE.

É possível exibir seletivamente os atributos especificando-os individualmente. Por exemplo:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +  
MAXDEPTH +  
MAXMSGL +  
CURDEPTH;
```

Este comando exibe os três atributos especificados da seguinte forma:

```
AMQ8409: Display Queue details.  
QUEUE(ORANGE.LOCAL.QUEUE)      TYPE(QLOCAL)  
CURDEPTH(0)                     MAXDEPTH(5000)  
MAXMSGL(4194304)
```

CURDEPTH é a profundidade da fila atual, ou seja, o número de mensagens na fila. Este é um atributo útil para exibir, porque, monitorando a profundidade da fila, é possível assegurar que a fila não se torne cheia.

Referências relacionadas

[DISPLAY QUEUE](#)

[DEFINE para filas](#)

Copiando uma definição de fila local com DEFINE QLOCAL

É possível copiar uma definição de filas usando o atributo **LIKE** no comando do MQSC **DEFINE QLOCAL**

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#) .

Sobre esta tarefa

É possível usar o comando **DEFINE** com o atributo **LIKE** para criar uma fila com os mesmos atributos da fila especificada, em vez daqueles da fila local padrão do sistema. Também é possível usar este formulário do comando **DEFINE** para copiar uma definição de fila, mas substitua uma ou mais mudanças nos atributos do original.

Notas:

1. Ao usar o atributo **LIKE** em um comando **DEFINE**, você está copiando somente os atributos da fila. Você não está copiando as mensagens na fila.
2. Se você definir uma fila local sem especificar **LIKE**, será o mesmo que:

```
DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Procedimento

- Para criar uma fila com os mesmos atributos da fila especificada, em vez daqueles da fila local padrão do sistema, insira o comando **DEFINE** conforme mostrado no exemplo a seguir.

Insira o nome da fila a ser copiado exatamente como foi inserido quando você criou a fila. Se o nome contiver caracteres minúsculos, coloque o nome entre aspas simples.

Esse exemplo cria uma fila com os mesmos atributos da fila ORANGE.LOCAL.QUEUE, em vez daqueles da fila local padrão do sistema:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE)
```

- Para copiar uma definição de fila, mas substituir uma ou mais mudanças nos atributos do original, insira o comando **DEFINE**, conforme mostrado no exemplo a seguir.

Esse comando copia os atributos da fila ORANGE.LOCAL.QUEUE para a fila THIRD.QUEUE, mas especifica que o comprimento máximo da mensagem na nova fila deve ser de 1024 bytes, em vez de 4194304:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE) +  
MAXMSGL(1024);
```

Referências relacionadas

[DEFINE para filas](#)

Mudando os atributos da fila local com ALTER QLOCAL ou DEFINE QLOCAL

É possível alterar atributos de filas de duas maneiras, usando o comando MQSC **ALTER QLOCAL** ou **DEFINE QLOCAL** com o atributo **REPLACE**.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Sobre esta tarefa

É possível usar o atributo **REPLACE** dos comandos **ALTER** e **DEFINE** para substituir uma definição existente pela nova definição especificada. A diferença entre usar **ALTER** e **DEFINE** é que **ALTER** com **REPLACE** não muda parâmetros não especificados, mas **DEFINE** com **REPLACE** configura todos os parâmetros.

Procedimento

- Para mudar os atributos da fila, use o comando **ALTER** ou o comando **DEFINE**, conforme mostrado nos exemplos a seguir.

Nesses exemplos, o comprimento máximo da mensagem na fila ORANGE.LOCAL.QUEUE diminui para 10.000 bytes.

- Usando o comando **ALTER**:

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Esse comando muda um único atributo, aquele do comprimento máximo de mensagem; todos os outros atributos permanecem os mesmos.

- Usando o comando **DEFINE** com a opção **REPLACE**, por exemplo:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Este comando muda não somente o comprimento máximo da mensagem, mas também todos os outros atributos aos quais são fornecidos seus valores padrão. Portanto, por exemplo, se a fila era anteriormente colocada como inibida, isso mudou para colocá-la como ativada, pois esse é o padrão, conforme especificado pela fila SYSTEM.DEFAULT.LOCAL.QUEUE.

Se você diminuir o comprimento máximo da mensagem em uma fila existente, as mensagens existentes não serão afetadas. Todas as mensagens novas, no entanto, devem atender aos novos critérios.

Referências relacionadas

[ALTER de Filas](#)

[ALTER QLOCAL](#)

[DEFINE para filas](#)

[DEFINE QLOCAL](#)

Limpando uma fila local com CLEAR QLOCAL

É possível usar o comando MQSC do **CLEAR QLOCAL** para limpar uma fila local

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Não é possível limpar uma fila se:

- Há mensagens não confirmadas que foram colocadas na fila sob o ponto de sincronização.
- Um aplicativo possui atualmente a fila aberta.

Sobre esta tarefa

Se você quiser limpar uma fila local usando o comando **CLEAR QLOCAL**, o nome da fila deverá ser definido para o gerenciador de filas locais.

Nota: Não há nenhum prompt que permita mudar de ideia; uma vez pressionada a tecla Enter as mensagens serão perdidas.

Procedimento

Para limpar as mensagens de uma fila local, use o **CLEAR QLOCAL** conforme mostrado no exemplo a seguir.

Neste exemplo, todas as mensagens são excluídas de uma fila local chamada MAGENTA.QUEUE:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

Referências relacionadas

[CLEAR QLOCAL](#)

Excluindo uma fila local com DELETE QLOCAL

É possível usar o comando MQSC do **DELETE QLOCAL** para excluir uma fila local

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Não será possível excluir uma fila se ela contiver mensagens não confirmadas.

Se uma fila tiver uma ou mais mensagens confirmadas e nenhuma mensagem não confirmada, ela poderá ser excluída apenas se você especificar a opção **PURGE**. A exclusão então avançará mesmo se houver mensagens confirmadas na fila nomeada, que também serão limpas.

Especificar **NOPURGE** em vez de **PURGE** assegura que a fila não seja excluída se contiver quaisquer mensagens confirmadas.

Procedimento

- Para excluir uma fila local, use o comando **DELETE QLOCAL**, conforme mostrado no exemplo a seguir. Este exemplo excluirá a fila PINK.QUEUE se não houver mensagens confirmadas na fila:

```
DELETE QLOCAL (PINK.QUEUE) NOPURGE
```

Este exemplo excluirá a fila PINK.QUEUE mesmo se houver mensagens confirmadas na fila:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

Referências relacionadas

[DELETE QLOCAL](#)

Procurando filas com o programa de amostra

IBM MQ fornece um navegador de fila de amostra que é possível utilizar para examinar o conteúdo das mensagens em uma fila.

Sobre esta tarefa

O navegador é fornecido nos formatos de origem e executável nos locais a seguir, em que *MQ_INSTALLATION_PATH* representa o diretório de alto nível no qual o IBM MQ está instalado.

Windows No Windows, os nomes de arquivo e os caminhos para o navegador de fila de amostra são os seguintes:

Origem

```
MQ_INSTALLATION_PATH\tools\c\samples\
```

Executável

```
MQ_INSTALLATION_PATH\tools\c\samples\bin\amqsbcg.exe
```

Linux **AIX** No AIX and Linux, os nomes de arquivo e caminhos são os seguintes:

Origem

```
MQ_INSTALLATION_PATH/samp/amqsbcg0.c
```

Executable

`MQ_INSTALLATION_PATH/samp/bin/amqsbcg`

Procedimento

- Para executar o programa de amostra, insira um comando, conforme mostrado no exemplo a seguir. O programa de amostra requer dois parâmetros de entrada, o nome da fila na qual as mensagens vão ser procuradas e o gerenciador de filas que possui essa fila. Por exemplo:

```
amqsbcg SYSTEM.ADMIN.QMGREVENT.tpp01 saturn.queue.manager
```

Resultados

Os resultados típicos desse comando são mostrados no exemplo a seguir:

```
AMQSBCG0 - starts here
*****

MQOPEN - 'SYSTEM.ADMIN.QMGR.EVENT'

MQGET of message number 1
****Message descriptor****

  StrucId : 'MD '   Version : 2
  Report  : 0   MsgType : 8
  Expiry  : -1   Feedback : 0
  Encoding: 546   CodedCharSetId : 850
  Format   : 'MQEVENT'
  Priority : 0   Persistence : 0
  MsgId   : X'414D512073617475726E2E71756575650005D30033563DB8'
  CorrelId: X'00000000000000000000000000000000000000000000'
  BackoutCount : 0
  ReplyToQ      : '          '
  ReplyToQMgr   : 'saturn.queue.manager'
  ** Identity Context
  UserIdentifier : '          '
  AccountingToken :
  X'000000000000000000000000000000000000000000000000'
  ApplIdentityData : '          '
  ** Origin Context
  PutApplType    : '7'
  PutApplName    : 'saturn.queue.manager'
  PutDate        : '19970417'   PutTime : '15115208'
  ApplOriginData : '          '

  GroupId : X'000000000000000000000000000000000000000000'
  MsgSeqNumber : '1'
  Offset       : '0'
  MsgFlags     : '0'
  OriginalLength : '104'

**** Message ****

length - 104 bytes

00000000: 0700 0000 2400 0000 0100 0000 2C00 0000 | .....→.....|'
00000010: 0100 0000 0100 0000 0100 0000 AE08 0000 | .....|'
00000020: 0100 0000 0400 0000 4400 0000 DF07 0000 | .....D.....|'
00000030: 0000 0000 3000 0000 7361 7475 726E 2E71 | ...0...saturn.q|'
00000040: 7565 7565 2E6D 616E 6167 6572 2020 2020 | ueue.manager  |'
00000050: 2020 2020 2020 2020 2020 2020 2020 2020 |          |'
00000060: 2020 2020 2020 2020 |          |'

No more messages
MQCLOSE
MQDISC
```

Referências relacionadas

[O programa de amostra Browser](#)

Ativando Filas Grandes

O IBM MQ suporta filas maiores que 2 TB.

Windows Em sistemas Windows, suporte para arquivos grandes está disponível sem qualquer ativação adicionais.

Linux **AIX** Nos sistemas AIX and Linux , é necessário ativar explicitamente o suporte de arquivo grande antes de poder criar arquivos de filas de vários gigabytes ou terabytes Consulte a documentação do sistema operacional para obter informações sobre como fazer isso.

Alguns utilitários, como o tar, não podem lidar com arquivos de vários gigabytes ou terabytes. Antes de ativar o suporte a arquivos grandes, verifique a documentação do seu sistema operacional para obter informações sobre as restrições sobre os utilitários que você usa.

Para obter informações sobre como planejar a quantia de armazenamento necessário para filas, consulte [Documentos de desempenho do MQ](#) para relatórios de desempenho específicos da plataforma.

É possível controlar o tamanho dos arquivos de fila usando um novo atributo em filas locais e de modelo Consulte [“Modificando arquivos de fila do IBM MQ”](#) na página 148 para obter mais informações.

Multi Modificando arquivos de fila do IBM MQ

É possível controlar o tamanho dos arquivos de fila usando um atributo em filas locais e de modelo É possível exibir o tamanho atual de um arquivo de fila e o tamanho máximo para o qual ele é capaz de crescer atualmente (com base no tamanho de bloco atualmente em uso nesse arquivo), usando dois atributos de status da fila.

Atributo usado para modificar arquivos de fila

O atributo em filas locais e modelos é:

MAXFSIZE

Denota o tamanho máximo do arquivo de fila usado pela fila, em megabytes.

É possível configurar ou exibir o valor desse atributo usando comandos MQSC, IBM MQ Explorer ou administrative REST API. Também é possível exibir o valor desse atributo no IBM MQ Console

Consulte [MAXFSIZE](#) e [“Mudando o tamanho de um arquivo de fila do IBM MQ”](#) na página 149 para obter mais informações..

O equivalente de PCF desse atributo é **MQIA_MAX_Q_FILE_SIZE** Consulte [Fila de Mudança, Cópia e Criação](#).

Os dois atributos no status da fila são:

CURFSIZE

Exibe o tamanho atual do arquivo de fila em megabytes, arredondado para o megabyte mais próximo.

É possível configurar ou exibir o valor desse atributo usando comandos MQSC, IBM MQ Explorer ou administrative REST API.

Consulte [CURFSIZE](#) para obter mais informações.

O equivalente de PCF desse atributo é **MQIA_CUR_Q_FILE_SIZE** Consulte [Fila de Consulta](#) e [Fila de Consulta \(resposta\)](#).

CURMAXFS

Indica o tamanho máximo atual a que o arquivo de fila pode crescer, arredondado para o megabyte mais próximo, dado o tamanho atual do bloco em uso em uma fila.

É possível configurar ou exibir o valor desse atributo usando comandos MQSC, IBM MQ Explorer ou administrative REST API.

Consulte [CURMAXFS](#) para obter mais informações.

O equivalente de PCF desse atributo é **MQIA_CUR_MAX_FILE_SIZE** Consulte [Fila de Consulta](#) e [Fila de Consulta \(resposta\)](#).

Granularidade e tamanho de bloco

Os arquivos de fila são divididos em segmentos chamados blocos. Para aumentar o tamanho máximo de um arquivo de fila, o tamanho de bloco ou a granularidade da fila pode precisar ser mudado pelo gerenciador de filas.

Se uma fila recém-definida for criada com um grande valor de **MAXFSIZE**, a fila será criada com um tamanho de bloco adequado. No entanto, se uma fila existente tiver seu valor **MAXFSIZE** aumentado, por exemplo, usando o comando MQSC **ALTER QLOCAL**, pode ser necessário permitir que a fila seja esvaziada para que o gerenciador de filas reconfigure a fila

Consulte [“Calculando a quantidade de dados que um arquivo de fila do IBM MQ pode armazenar”](#) na página 150 para obter mais informações.



Atenção: Alguns sistemas de arquivos e sistemas operacionais têm limites no tamanho de todo o sistema de arquivos ou no tamanho de um arquivo individual. É necessário verificar os limites nos sistemas usados por sua empresa.

Referências relacionadas

[ALTER QUEUES](#)

[DISPLAY QUEUE](#)



[DISPLAY QSTATUS](#)

Mudando o tamanho de um arquivo de fila do IBM MQ

É possível aumentar ou diminuir o tamanho máximo de um arquivo de fila.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos MQSC:

-  **ALW** Sobre AIX, Linux, and Windows, você emite comandos MQSC de um **runmqsc** prompt de comando. Ver [Executando comandos MQSC interativamente em runmqsc](#) e [Executando comandos MQSC a partir de arquivos de texto em runmqsc](#).
-  **IBM i** Sobre IBM i, você cria uma lista de comandos em um arquivo de script e, em seguida, executa o arquivo usando o comando **STRMQMMQSC** comando. Ver [Administração usando comandos MQSC em IBM i](#).

Antes de configurar um novo tamanho para um arquivo de filas, use o comando MQSC [DISPLAY QLOCAL](#) para ver o tamanho do arquivo de filas que deseja alterar. Por exemplo, emita o seguinte comando:

```
DISPLAY QLOCAL (SYSTEM.DEFAULT.LOCAL.QUEUE) MAXFSIZE
```

Você recebe a seguinte saída:

```
AMQ8409I: Display queue details
  QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)      TYPE (QLOCAL)
  MAXFSIZE (DEFAULT)
```

que mostra que o tamanho máximo do arquivo de fila é o valor padrão de 2.088.960 MB.


Sobre esta tarefa

Os procedimentos a seguir mostram como:

- Reduzir o tamanho máximo ao qual um arquivo de fila pode crescer.
- Aumentar o tamanho máximo ao qual um arquivo de fila pode crescer.



Atenção: É necessário ser cauteloso ao aumentar o tamanho dos arquivos de fila sem considerar a maneira com que os aplicativos são gravados e o possível efeito no desempenho. Acessar as mensagens aleatoriamente em um arquivo de fila muito grande pode ser muito lento.

Se você estiver considerando aumentar o tamanho máximo de um arquivo de filas além do padrão, deverá ter cuidado ao usar seletores de mensagens, como IDs de correlação e seqüências do seletor IBM MQ classes for JMS  ou [IBM MQ classes for Jakarta Messaging](#) . Os arquivos de fila maiores são mais adequados para o acesso primeiro a entrar, primeiro a sair da fila.

Considere ter quantidades muito grandes de dados em arquivos de fila individuais apenas em gerenciadores de filas configurados para a criação de log circular ou quando a imagem de mídia não tiver sido ativada para a fila individual.

Não é necessário limitar o tamanho das filas do SISTEMA, pois isso pode afetar a operação do gerenciador de filas.

Procedimento

1. Reduzir o tamanho máximo do arquivo de fila

- a) Emita o seguinte comando MQSC para criar um arquivo local chamado SMALLQUEUE, com um tamanho de 500 gigabytes:

```
DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
2 : DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
AMQ8006I: IBM MQ queue created
```

e você receberá a mensagem: AMQ8006I:

Nota: Se você configurar uma fila com um valor menor que a quantia de dados que já está no arquivo, não será possível colocar novas mensagens na fila.

Se um aplicativo tentar colocar uma mensagem em um arquivo de fila que não tenha espaço suficiente, o aplicativo receberá o código de retorno MQRC_Q_SPACE_NOT_AVAILABLE. Quando mensagens suficientes são lidas destrutivamente na fila, os aplicativos podem começar a colocar novas mensagens na fila.

2. Aumentar o tamanho máximo do arquivo de fila.

- a) Emita o comando MQSC a seguir para criar um arquivo local chamado LARGEQUEUE, com um tamanho de 5 terabytes:



```
DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
3 : DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
AMQ8006I: IBM MQ queue created
```

Calculando a quantia de dados que um arquivo de fila do IBM MQ pode armazenar

A quantia de dados que podem ser armazenados em uma fila é limitada pelo tamanho dos blocos individuais nos quais a fila é dividida. Use comandos MQSC para confirmar o tamanho de bloco e a granularidade e verifique o tamanho de um arquivo de filas

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos MQSC:

-  Sobre AIX, Linux, and Windows , você emite comandos MQSC de um `runmqsc` prompt de comando. Ver [Executando comandos MQSC interativamente em runmqsc](#) e [Executando comandos MQSC a partir de arquivos de texto em runmqsc](#) .
-  Sobre IBM i , você cria uma lista de comandos em um arquivo de script e, em seguida, executa o arquivo usando o comando `STRMQMMQSC` comando. Ver [Administração usando comandos MQSC em IBM i](#) .

Procedimento

- Confirmar tamanho de bloco e granularidade.

O tamanho do bloco padrão é 512 bytes. Para suportar arquivos de fila maiores que dois terabytes, o gerenciador de filas precisará aumentar o tamanho do bloco.

O tamanho do bloco é calculado automaticamente quando você configura o `MAXFSIZE` de uma fila, mas o tamanho do bloco revisado não poderá ser aplicado à fila se ela já tiver mensagens nele. Quando uma fila está vazia, o gerenciador de filas modifica automaticamente o tamanho do bloco para suportar o `MAXFSIZE` configurado.

O comando **DISPLAY QSTATUS** possui um novo atributo, `CURMAXFS`, que permite confirmar que uma fila foi modificada para usar um novo tamanho do bloco

No exemplo a seguir, o valor de `CURMAXFS` de 4177920 confirma que o arquivo de fila está atualmente apto a crescer para aproximadamente quatro terabytes. Se o valor de `MAXFSIZE` configurado na fila for maior que o valor de `CURMAXFS`, o gerenciador de filas ainda estará esperando que a fila seja esvaziada antes de reconfigurar o tamanho do bloco do arquivo de fila..

```
DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
  2 : DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
AMQ8450I: Display queue status details
  QUEUE(LARGEQUEUE)                TYPE(QUEUE)
  CURMAXFS(4177920)                 CURDEPTH(100000)
```

- Verifique o tamanho de um arquivo de fila

É possível exibir o tamanho atual de um arquivo de fila no disco, em megabytes usando o atributo `CURFSIZE` no comando **DISPLAY QSTATUS**. Isso pode ser útil em plataformas como o IBM MQ Appliance, nas quais não é possível acessar o sistema de arquivos diretamente.

```
DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
  1 : DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
AMQ8450I: Display queue status details
  QUEUE(SMALLQUEUE)                TYPE(QUEUE)
  CURDEPTH(4024)                   CURFSIZE(10)
```

Nota: Quando uma fila tiver mensagens removidas dela, o atributo `CURFSIZE` não diminuirá imediatamente

Geralmente, o espaço em um arquivo de fila é liberado apenas quando nenhum aplicativo tiver a fila aberta e não houver mensagens em dúvida armazenadas na fila. Qualquer truncamento ou compactação necessária de um arquivo de filas carregado pelo gerenciador de filas ocorre durante o ponto de verificação, encerramento do gerenciador de filas ou durante a gravação de uma imagem de mídia da fila.

Referências relacionadas

[ALTER QUEUES](#)

[DISPLAY QSTATUS](#)

Trabalhando com Filas Remotas

Uma fila remota é uma definição em um gerenciador de fila local que se refere a uma fila em um gerenciador de filas remotas. Não é necessário definir uma fila remota a partir de uma posição local, mas se você fizer isso, os aplicativos poderão fazer referência à fila remota por seu nome definido localmente, em vez de ter que especificar um nome qualificado pelo ID do gerenciador de filas no qual a fila remota está localizada.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Sobre esta tarefa

Um aplicativo conecta-se a um gerenciador de filas locais e, em seguida, emite uma chamada MQOPEN. Na chamada aberta, o nome da fila especificado é aquele de uma definição de fila remota no gerenciador de filas locais. A definição de fila remota fornece os nomes da fila de destino, o gerenciador de filas de destino e, opcionalmente, uma fila de transmissão. Para colocar uma mensagem na fila remota, o aplicativo emite uma chamada MQPUT, especificando o identificador retornado da chamada MQOPEN. O gerenciador de filas usa o nome da fila remota e o nome do gerenciador de filas remotas em um cabeçalho de transmissão no início da mensagem. Estas informações são usadas para rotear a mensagem para seu destino correto na rede.

Como administrador, é possível controlar o destino da mensagem, mudando a definição de fila remota.

Procedimento

- Coloque uma mensagem numa fila pertencente a um gerenciador de filas remotas.

O aplicativo se conecta a um gerenciador de filas, por exemplo, saturn.queue.manager. A fila de destino pertence a outro gerenciador de filas.

Na chamada MQOPEN, o aplicativo especifica estes campos:

Valor do Campo	Descrição
<i>ObjectName</i> CYAN.REMOTE.QUEUE	Especifica o nome local do objeto de fila remota. Define a fila de destino e o gerenciador de filas de destino.
<i>ObjectType</i> (Queue)	Identifica esse objeto como uma fila.
<i>ObjectQmgrName</i> Em branco ou saturn.queue.manager	Este campo é opcional. Se estiver em branco, o nome do gerenciador de fila local será assumido. (Este é o gerenciador de filas no qual a definição de fila remota existe.)

Depois disso, o aplicativo emite uma chamada MQPUT para colocar uma mensagem nessa fila.

No gerenciador de filas locais, é possível criar uma definição local de uma fila remota utilizando os seguintes comandos do MQSC:

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
DESCR ('Queue for auto insurance requests from the branches') +
RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
RQMNAME (jupiter.queue.manager) +
XMITQ (INQUOTE.XMIT.QUEUE)
```

em que:

QREMOTE (CYAN.REMOTE.QUEUE)

Especifica o nome local do objeto de fila remota. Esse é o nome que os aplicativos conectados a este gerenciador de filas devem especificar na chamada MQOPEN para abrir a fila AUTOMOBILE.INSURANCE.QUOTE.QUEUE no gerenciador de filas remotas do jupiter.queue.manager.

DESCR ('Queue for auto insurance requests from the branches')

Fornecer texto adicional que descreve o uso da fila.

RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)

Especifica o nome da fila de destino no gerenciador de filas remotas. Esta é a fila de destino real para mensagens enviadas por aplicativos que especificam o nome da fila CYAN.REMOTE.QUEUE. A fila AUTOMOBILE.INSURANCE.QUOTE.QUEUE deve ser definida como uma fila local no gerenciador de filas remotas.

RQMNAME (jupiter.queue.manager)

Especifica o nome do gerenciador de filas remotas que tem a fila de destino AUTOMOBILE.INSURANCE.QUOTE.QUEUE.

XMITQ (INQUOTE.XMIT.QUEUE)

Especifica o nome da fila de transmissão. Isso é opcional; se o nome de uma fila de transmissão não for especificado, uma fila com o mesmo nome que o gerenciador de filas remoto é utilizada.

Em qualquer um dos casos, a fila de transmissão apropriada deve ser definida como uma fila local com um atributo **Usage** que especifica que é uma fila de transmissão (USAGE(XMITQ) em comandos MQSC).

- Coloque as mensagens em uma fila remota (método alternativo).

Utilizar uma definição local de uma fila remota não é a única maneira de colocar mensagens em uma fila remota. Os aplicativos podem especificar o nome da fila completo, incluindo o nome do gerenciador de filas remotas, como parte da chamada MQOPEN. Neste caso, você não precisa de uma definição local de uma fila remota. No entanto, isto significa que os aplicativos devem conhecer ou ter acesso ao nome do gerenciador de filas remotas no tempo de execução.

- Use outros comandos com filas remotas..

É possível utilizar os comandos do MQSC para exibir ou alterar os atributos de um objeto de fila remoto ou é possível excluir o objeto de fila remota. Por exemplo:

- Para exibir os atributos da fila remota:

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- Para mudar a fila remota para ativar puts. Isso não afeta a fila de destino, somente os aplicativos que especificam esta fila remota:

```
ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)
```

- Para excluir esta fila remota. Isso não afeta a fila de destino, somente sua definição de local:

```
DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

Nota: Quando você exclui uma fila remota, você exclui somente a representação local da fila remota. Você não exclui a fila remota em si ou quaisquer mensagens nela.

As definições de fila remota podem ser usadas como aliases

Além de localizar uma fila em outro gerenciador de filas, também é possível usar uma definição local de uma fila remota para aliases do gerenciador de filas e aliases da fila de resposta. Ambos os tipos de alias são resolvidas através da definição local de uma fila remota. Deve-se configurar os canais apropriados para a mensagem chegar em seu destino.

Aliases do gerenciador de filas

Um alias é o processo pelo qual o nome do gerenciador de filas de destino, conforme especificado em uma mensagem, é mudado por um gerenciador de filas na rota da mensagem. aliases do gerenciador de filas são importantes porque é possível utilizá-las para controlar o destino de mensagens em uma rede de gerenciadores de filas.

Você faz isso mudando a definição de fila remota no gerenciador de filas no ponto de controle. O aplicativo de envio não está ciente de que o nome do gerenciador de filas especificado é um alias.

Para obter mais informações sobre aliases do gerenciador de filas, consulte [O que são aliases?](#)

Aliases da Fila de Resposta

Opcionalmente, um aplicativo pode especificar o nome de uma fila de resposta quando ela coloca uma *mensagem de solicitação* em uma fila.

Se o aplicativo que processa a mensagem extrai o nome da fila de resposta, ele saiba onde enviar as *mensagem de resposta*, se necessário.

Um alias da fila de resposta é o processo pelo qual uma fila de resposta, conforme especificado em uma mensagem de solicitação, for alterado por um gerenciador de filas na rota da mensagem. O aplicativo de envio não está ciente de que o nome da fila de resposta especificado é um alias.

Um alias da fila de resposta permite que você altere o nome da fila de resposta e, opcionalmente, seu gerenciador de filas. Isso, por sua vez, permite que você controle qual rota é utilizada para mensagens de resposta.

Para obter mais informações sobre mensagem de solicitação, mensagens de resposta e filas de resposta, consulte [Tipos de mensagem](#) e [Fila de resposta e gerenciador de filas](#).

Para obter mais informações sobre aliases da fila de resposta, consulte [Aliases de fila de resposta e clusters](#).

Trabalhando com Filas de Alias

É possível definir uma fila de alias para fazer referência indiretamente a outra fila ou tópico.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).



Atenção: As listas de distribuição não suportam o uso de filas de alias que apontam para objetos do tópico. Se uma fila de alias apontar para um objeto de tópico em uma lista de distribuição, o IBM MQ retornará MQRC_ALIAS_BASE_Q_TYPE_ERROR.

Sobre esta tarefa

A fila à qual uma fila de alias se refere pode ser qualquer uma das seguintes:

- Uma fila local (consulte [“Definindo uma fila local com DEFINE QLOCAL”](#) na página 142).
- Uma definição local de uma fila remota (consulte [“Trabalhando com Filas Remotas”](#) na página 151).
- Um tópico.

Uma fila de alias não é uma fila real, mas uma definição que resolve uma fila real (ou destino) no tempo de execução. A definição da fila de alias especifica a fila de destino. Quando um aplicativo faz uma chamada MQOPEN para uma fila de alias, o gerenciador de filas resolve o alias para o nome de fila de destino.

Uma fila de alias não pode resolver para outro alias da fila definida localmente. No entanto, uma fila de alias podem resolver para as filas de alias que são definidos em outro lugar no clusters do qual o gerenciador de filas locais é um membro. Consulte [Resolução de Nome](#) para obter informações adicionais.

As filas de alias são úteis para:

- Fornecer aos diferentes aplicativos diferentes níveis de autoridades de acesso à fila de destino.
- Permitir que os diferentes aplicativos trabalhem com a mesma fila de diferentes maneiras. (Talvez você queira designar diferentes prioridades padrão ou diferentes valores de persistência padrão.)
- Simplificar a manutenção, a migração e o balanceamento de carga de trabalho. (Talvez você queira alterar o nome da fila de destino sem ter que alterar o seu aplicativo, que continua usando o alias.)

Por exemplo, assuma que um aplicativo tenha sido desenvolvido para colocar as mensagens em uma fila denominada MY.ALIAS.QUEUE. Isso especifica o nome desta fila quando ela fizer uma solicitação MQOPEN

e, indiretamente, se colocar uma mensagem nesta fila. O aplicativo não reconhece que a fila é uma fila de alias. Para cada chamada MQI que usa este alias, o gerenciador de filas resolve o nome de fila real, que poderia ser uma fila local ou uma fila remota definida neste gerenciador de filas.

Alterando o valor do atributo TARGET, é possível redirecionar chamadas MQI para outra fila, possivelmente em outro gerenciador de filas. Isso é útil para manutenção, migração e balanceamento de carga.

Procedimento

- Defina uma fila de alias

O comando MQSC a seguir cria uma fila de alias:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGET (YELLOW.QUEUE)
```

Este comando redireciona as chamadas MQI que especificam o MY.ALIAS.QUEUE para a fila YELLOW.QUEUE. O comando não cria a fila de destino; as chamadas MQI falharão se a fila YELLOW.QUEUE não existir o tempo de execução.

Se você alterar a definição de alias, poderá redirecionar as chamadas MQI para outra fila. Por exemplo:

```
ALTER QALIAS (MY.ALIAS.QUEUE) TARGET (MAGENTA.QUEUE)
```

Este comando redireciona as chamadas MQI para outra fila, MAGENTA.QUEUE.

Também é possível usar as filas de alias para fazer com que uma única fila (fila de destino) pareça ter diferentes atributos para diferentes aplicativos. Isso é feito definindo dois alias, um para cada aplicativo. Imagine que existem dois aplicativos:

- O aplicativo ALPHA pode colocar as mensagens no YELLOW.QUEUE, mas não tem permissão para obter as mensagens dele.
- O aplicativo BETA pode obter as mensagens do YELLOW.QUEUE, mas não tem permissão para colocar as mensagens nele.

O comando MQSC a seguir define um alias que é colocado ativado e desativado para o aplicativo ALPHA:

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (ENABLED) +  
GET (DISABLED)
```

O seguinte comando define um alias que é desativado e ativado para o aplicativo BETA:

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (DISABLED) +  
GET (ENABLED)
```

O ALPHA usa o nome de fila ALPHAS.ALIAS.QUEUE em suas chamadas MQI; BETA usa o nome de fila BETAS.ALIAS.QUEUE. Ambos acessam a mesma fila, mas de diferentes maneiras.

É possível usar os atributos LIKE e REPLACE ao definir os alias de fila, na mesma maneira que você usa esses atributos com as filas locais.

- Use outros comandos com as filas de alias

É possível usar os comandos MQSC apropriados para exibir ou alterar os atributos de filas de alias ou excluir o objeto da fila de alias. Por exemplo:

Use o comando **DISPLAY QALIAS** para exibir os atributos da fila de alias:

```
DISPLAY QALIAS (ALPHAS.ALIAS.QUEUE)
```

Use o comando **ALTER QALIAS** para alterar o nome da fila de base, para o qual o alias é resolvido, em que a opção **force** força a mudança mesmo se a fila está aberta:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGET(ORANGE.LOCAL.QUEUE) FORCE
```

Use o comando **DELETE QALIAS** para excluir este alias da fila:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Não é possível excluir uma fila de alias se um aplicativo tiver atualmente a fila aberta.

Conceitos relacionados

[Listas de distribuição](#)

Referências relacionadas

[ALTER QALIAS](#)

[DEFINE QALIAS](#)

[DELETE QALIAS](#)

Trabalhando com filas modelo

Filas modelo fornece um método conveniente para aplicativos para criar filas conforme necessário.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Sobre esta tarefa

Um gerenciador de filas cria uma *fila dinâmica* se receber uma chamada MQI de um aplicativo especificando um nome de fila que foi definida como uma fila modelo. O nome da fila dinâmica nova é gerada pelo gerenciador de filas quando a fila é criada. Uma *fila modelo* é um modelo que especifica os atributos de quaisquer filas dinâmicas criadas por meio dele.

Procedimento

- Defina uma fila de modelo

Você usa o comando MQSC **DEFINE QMODEL** para definir uma fila modelo com um conjunto de atributos da mesma forma que você define uma fila local. Filas modelos e filas locais possuem o mesmo conjunto de atributos, exceto que em filas modelos é possível especificar se as filas dinâmicas permanentes ou temporárias são criadas. (As filas permanentes são mantidas nos reinícios do gerenciador de filas, os temporários não são.) Por exemplo:

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +
DESCR('Queue for messages from application X') +
PUT (DISABLED) +
GET (ENABLED) +
NOTRIGGER +
MSGDLVSQ (FIFO) +
MAXDEPTH (1000) +
MAXMSGL (2000) +
USAGE (NORMAL) +
DEFTYPE (PERMDYN)
```

Este comando cria uma definição de fila modelo. No atributo **DEFTYPE**, é possível ver que as filas reais criadas por meio desse modelo são filas dinâmicas permanentes. Quaisquer atributos não especificados são automaticamente copiados da fila padrão `SYSTEM.DEFAULT.MODEL.QUEUE`.

É possível usar os atributos **LIKE** e **REPLACE** ao definir filas modelo, da mesma maneira que você os usa com as filas locais.

- Use outros comandos com filas de modelo

É possível usar os comandos MQSC apropriados para exibir ou alterar os atributos de uma fila modelo ou excluir o objeto da fila de modelo. Por exemplo:

Use o comando **DISPLAY QUEUE** para exibir os atributos da fila modelo:

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

Use o comando **ALTER QMODEL** para alterar o modelo para ativar colocações em qualquer fila dinâmica criada por meio desse modelo:

```
ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)
```

Use o comando **DELETE QMODEL** para excluir esta fila modelo:

```
DELETE QMODEL (RED.MODEL.QUEUE)
```

Referências relacionadas

[ALTER QMODEL](#)

[DEFINE QMODEL](#)

[DELETE QMODEL](#)

[DISPLAY QUEUE](#)

Trabalhando com filas de mensagens não entregues

Cada gerenciador de filas geralmente possui uma fila local para utilizar como uma fila de mensagens não entregues, de forma que mensagens que não puderem ser entregues ao seu destino correto possam ser armazenadas para recuperação posterior. Você informa ao gerenciador de filas sobre a fila de mensagens não entregues e especifica como as mensagens localizadas em uma fila de mensagens não entregues devem ser processadas. Utilizar as filas de mensagens não entregues pode afetar a sequência na qual as mensagens são entregues, assim é possível optar por não as usar.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Sobre esta tarefa

Uma fila de mensagens não entregues de amostra chamado `SYSTEM.DEAD.LETTER.QUEUE` está disponível com o produto. Esta fila é criada automaticamente quando você cria o gerenciador de filas. É possível modificar essa definição, se necessário e renomeá-la.

Uma fila de mensagens não entregues não possui requisitos especiais, exceto que:

- Ele deve ser uma fila local
- Seu atributo `MAXMSGL` (comprimento máximo da mensagem) deve ativar a fila para acomodar as maiores mensagens que o gerenciador de filas possui para manipular e o tamanho do cabeçalho de mensagens não entregues (`MQDLH`)

Utilizar as filas de mensagens não entregues pode afetar a sequência na qual as mensagens são entregues, assim é possível optar por não as usar.

Procedimento

- Informe o gerenciador de filas sobre a fila de devoluções

Para fazer isso, especifique um nome da fila de devoluções no comando **crtmqm** (`crtmqm -u DEAD.LETTER.QUEUE`, por exemplo) ou usando o atributo **DEADQ** no comando **ALTER QMGR** para especificar um posteriormente. Deve-se definir a fila de mensagens não entregues antes de usá-lo.

- Especifique como as mensagens localizadas em uma fila de mensagens não entregues devem ser processada

Você define o atributo do canal **USEDLQ** para determinar se a fila de mensagens não entregues é usada quando as mensagens não podem ser entregues. Este atributo pode ser configurado para que algumas funções do gerenciador de filas usem a fila de devoluções, enquanto outras funções não. Para obter mais informações sobre o uso do atributo do canal **USEDLQ** em diferentes comandos **MQSC**, consulte [DEFINE CHANNEL](#), [DISPLAY CHANNEL](#), [ALTER CHANNEL](#) e [DISPLAY CLUSQMGR](#).

Use o manipulador de fila de mensagens não entregues **IBM MQ** para especificar como as mensagens localizadas em uma fila de mensagens não entregues devem ser processadas ou removidas. Consulte [“Processando mensagens em uma fila de mensagens não entregues do IBM MQ”](#) na página 158.

Conceitos relacionados

[Filas de Devoluções](#)

Tarefas relacionadas

[Resolução de problemas de mensagens não entregues](#)

Referências relacionadas

[ALTER QMGR](#)

[crtmqm \(criar gerenciador de filas\)](#)

Processando mensagens em uma fila de mensagens não entregues do IBM MQ

Para processar mensagens em uma fila de mensagens não entregues (DLQ), use o manipulador de DLQ padrão fornecido pelo IBM MQ. O manipulador corresponde mensagens na fila de mensagens não entregues com relação às entradas em uma tabela de regras que você define.

Sobre esta tarefa

As mensagens podem ser colocadas em uma fila de mensagens não entregues pelos gerenciadores de fila, agentes do canal de mensagem (MCAs) e aplicativos. Todas as mensagens no DLQ devem ser prefixadas com uma estrutura de *cabeçalho de devoluções*, **MQDLH**. As mensagens colocadas na fila de mensagens não entregues por um gerenciador de filas ou um agente do canal de mensagem sempre possuem esse cabeçalho; aplicativos que colocam mensagens na fila de mensagens não entregues devem fornecer esse cabeçalho. O campo *Razão* da estrutura **MQDLH** contém um código de razão que identifica o motivo pelo qual a mensagem está no DLQ.

Todos os ambientes do IBM MQ precisam de uma rotina para processar mensagens na fila de mensagens não entregues regularmente. O IBM MQ fornece uma rotina padrão, chamada *manipulador de filas de mensagens não entregues* (o manipulador de fila de mensagens não entregues), que você chama usando o comando **MQSC** **runmqdlq**.

As instruções para processar as mensagens no DLQ são fornecidas para o manipulador DLQ por meio de uma *tabela de regras* gravada pelo usuário. Ou seja, o manipulador DLQ corresponde às mensagens no DLQ nas entradas na tabela de regras; quando uma mensagem DLQ corresponder a uma entrada na tabela de regras, o manipulador DLQ executará a ação associada com essa entrada.

Tarefas relacionadas

[Resolução de problemas de mensagens não entregues](#)

Referências relacionadas

[Filas de Devoluções](#)

Chamando o manipulador da fila de mensagens não entregues

Chame o manipulador de fila de mensagens não entregues (DLQ) usando o comando de controle **q runmqdl**. É possível nomear a fila de mensagens não entregues que você deseja processar e o gerenciador de filas que você deseja usar de duas maneiras.

Antes de começar

Para executar o manipulador de fila de mensagens não entregues deve-se estar autorizado a acessar a própria fila de mensagens não entregues e qualquer fila de mensagem para as quais as mensagens na fila de mensagens não entregues são encaminhadas. Para o manipulador de fila de mensagens não entregues colocar as mensagens nas filas com a autoridade do ID do usuário no contexto da mensagem, também deve-se estar autorizado a assumir a identidade de outros usuários.

Sobre esta tarefa

Os exemplos a seguir se aplicam ao DLQ chamado `ABC1.DEAD.LETTER.QUEUE`, de propriedade do gerenciador de fila `ABC1.QUEUE.MANAGER`

Se você não especificar o DLQ ou o gerenciador de filas conforme mostrado, o gerenciador de filas padrão para a instalação será usado juntamente com o DLQ pertencente a esse gerenciador de filas.

O comando **runmqdlq** obtém sua entrada de `stdin`. Você associa a tabela de regras ao **runmqdlq** redirecionando `stdin` da tabela de regras.

Para obter mais informações sobre o comando **runmqdlq**, consulte [runmqdlq](#).

Procedimento

- É possível nomear o DLQ e o gerenciador de filas como parâmetros para o comando **runmqdlq**

Por exemplo, no prompt de comandos:

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrule.rul
```

- É possível nomear o DLQ e o gerenciador de fila na tabela de regras

Por exemplo:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

Conceitos relacionados

[Filas de Devoluções](#)

Tarefas relacionadas

[Resolução de problemas de mensagens não entregues](#)

O manipulador de DLQ de amostra amqsdlq

Além do manipulador da fila de devoluções chamado usando o comando **runmqdlq**, o IBM MQ fornece a origem de um manipulador da fila de devoluções de amostra **amqsdlq** com uma função que é semelhante àquela fornecida pelo **runmqdlq**.

É possível customizar **amqsdlq** para fornecer um manipulador de DLQ que atenda aos seus requisitos. Por exemplo, é possível decidir que você deseja um manipulador de fila de mensagens não entregues que podem processar mensagens sem cabeçalho de mensagens não entregues. (Tanto o manipulador de DLQ padrão e a amostra, **amqsdlq**, processam apenas essas mensagens no DLQ que começam com um cabeçalho de devoluções, MQDLH. As mensagens que não começam com um MQDLH são identificadas como estando em erro e permanecem na fila de mensagens não entregues indefinidamente.)

O `MQ_INSTALLATION_PATH` representa o diretório de alto nível no qual o IBM MQ está instalado.

No IBM MQ for Windows, a origem de `amqsdlq` é fornecido no diretório:

```
MQ_INSTALLATION_PATH\tools\c\samples\d1q
```

e a versão compilada é fornecida no diretório:

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

Em sistemas IBM MQ for UNIX e Linux, a origem de **amqsdlq** é fornecida no diretório:



```
MQ_INSTALLATION_PATH/samp/d1q
```

e a versão compilada é fornecida no diretório:

```
MQ_INSTALLATION_PATH/samp/bin
```

Uma versão construída do programa de amostra, denominada **amqsdlqc**, está incluída. É possível usar isso para se conectar a um gerenciador de filas remotas no modo cliente. Para usar o **amqsdlqc**, deve-se configurar uma das variáveis de ambiente `MQSERVER`, `MQCHLLIB` ou `MQCHLTAB` para identificar como se conectar ao gerenciador de filas. Por exemplo:

```
export MQSERVER="SYSTEM.DEF.SVRCONN/TCP/myappliance.co.uk(1414) "
```

  *A tabela de regras do manipulador DLQ*

A tabela de regras do manipulador da fila de devoluções define como o manipulador de DLQ processa as mensagens que chegam na DLQ.

Há dois tipos de entrada em uma tabela de regras:

- A primeira entrada na tabela, que é opcional, contém *dados de controle*.
- Todas as outras entradas na tabela são *regras* para o manipulador de fila de mensagens não entregues seguir. Cada regra consiste em um *padrão* (um conjunto de características da mensagem) em relação ao qual uma mensagem é correspondida e uma *ação* a ser tomada quando uma mensagem na fila de mensagens não entregues corresponder ao padrão especificado. Deve haver pelo menos uma regra em uma tabela de regras.

Cada entrada na tabela de regras é composta por uma ou mais palavras-chave.

Conceitos relacionados

[Filas de Devoluções](#)

Tarefas relacionadas

[Resolução de problemas de mensagens não entregues](#)

  *Dados de controle de DLQ*

É possível incluir palavras-chave em uma entrada de dados de controle em uma tabela de regras do manipulador da fila de devoluções.

Nota:

- A linha vertical (|) separa alternativas, somente um dos quais pode ser especificado.
- Todas as palavras-chave são opcionais.

INPUTQ (*QueueName* | " (padrão)

O nome da fila de mensagens não entregues que você deseja processar:

1. Qualquer valor que você fornecer INPUTQ como um parâmetro para o comando `runmqdlq` substitui qualquer valor INPUTQ na tabela de regras.
2. Se você não especificar um valor INPUTQ como um parâmetro para o comando `runmqdlq`, mas **especificar** um valor na tabela de regras, o valor INPUTQ na tabela de regras será utilizado.

3. Se nenhuma fila de mensagens não entregues for especificada ou você especificar INPUTQ(' ') na tabela de regras, o nome da fila de mensagens não entregues pertencente ao gerenciador de filas com o nome que é fornecido como um parâmetro para o comando runmqdlq é utilizado.
4. Se você não especificar um valor INPUTQ como um parâmetro para o comando runmqdlq ou como um valor na tabela de regras, a fila de mensagens não entregues pertencente ao gerenciador de filas nomeado na palavra-chave INPUTQM na tabela de regras será usada.

INPUTQM (*QueueManagerNome* | " (padrão)

O nome do gerenciador de filas que possui a fila de mensagens não entregues nomeado na palavra-chave INPUTQ:

1. Qualquer fonte de valor que você INPUTQM como um parâmetro para o comando runmqdlq substitui qualquer valor INPUTQM na tabela de regras.
2. Se você não especificar um valor INPUTQM como um parâmetro para o comando runmqdlq, o valor INPUTQM na tabela de regras será utilizado.
3. Se nenhum gerenciador de filas for especificado ou você especificar INPUTQM(' ') na tabela de regras, o gerenciador de filas padrão para a instalação será usado.

RETRYINT (*Intervalo* | 60 (padrão)

O intervalo, em segundos, no qual o manipulador de fila de mensagens não entregues deve reprocessar mensagens na fila de mensagens não entregues que não pôde ser processado na primeira tentativa e para o qual as tentativas repetidas foram solicitadas. Por padrão, o intervalo de novas tentativas é 60 segundos.

WAIT (YES (padrão) |NO|*nnn*)

Se o manipulador de fila de mensagens não entregues deve aguardar a chegada de novas mensagens na fila de mensagens não entregues quando ele detecta que não há mensagens adicionais que possa processar.

SIM

O manipulador de fila de mensagens não entregues aguardará indefinidamente.

NÃO

O manipulador de fila de mensagens não entregues termina quando ele detecta que a fila de mensagens não entregues está vazia ou não contém mensagens que ele pode processar.

nnn

O manipulador de fila de mensagens não entregues aguardará até *nnn* segundos pela chegada do novo trabalho antes de terminar após detectar que a fila está vazia ou que não contém nenhuma mensagem que possa processar.

Especifique WAIT (YES) para DLQs ocupadas e WAIT (NO) ou WAIT (*nnn*) para DLQs que possuem um baixo nível de atividade. Se o manipulador de fila de mensagens não entregues tem permissão para terminar, chame-o novamente utilizando o acionamento. Para obter mais informações sobre acionamento, consulte [Iniciando aplicativos IBM MQ usando acionadores](#).

Uma alternativa para incluir dados de controle na tabela de regras é fornecer os nomes da fila de mensagens não entregues e seu gerenciador de filas como parâmetros de entrada para o comando runmqdlq. Se você especificar um valor tanto na tabela de regras e como entrada para o comando runmqdlq, o valor especificado no comando runmqdlq terá precedência.

Se você incluir uma entrada de dados de controle na tabela de regras, ela deverá ser a **primeira** entrada na tabela.

Regras de DLQ (padrões e ações)

Uma descrição das palavras-chave de correspondência de padrões (aquelas com relação às quais as mensagens na fila de mensagens não entregues são correspondidas) e as palavras-chave de ação (aquelas que determinam como o manipulador de DLQ deve processar uma mensagem correspondente). Uma regra de exemplo também é fornecida.

Palavras-chave de correspondência de padrões

As palavras-chave de correspondência de padrão, que você utiliza para especificar os valores em relação aos quais as mensagens na fila de mensagens não entregues são correspondidas, são as seguintes. (Todas as palavras-chave de correspondência de padrões são opcionais):

APPLIDAT (*ApplIdentityDados* | * (padrão))

O valor *ApplIdentityData* especificado no descritor de mensagens, MQMD, da mensagem na fila de mensagens não entregues.

APPLNAME (*PutApplNome* | * (padrão))

O nome do aplicativo que emitiu a chamada MQPUT ou MQPUT1, conforme especificado no campo *PutApplName* do descritor de mensagens, MQMD, da mensagem no DLQ.

APPLTYPE (*PutApplTipo* | * (padrão))

O valor *PutApplType*, especificado no descritor de mensagens, MQMD, da mensagem na fila de mensagens não entregues.

DESTQ (*QueueName* | * (padrão))

O nome da fila de mensagens para a qual a mensagem é destinada.

DESTQM (*QueueManagerNome* | * (padrão))

O nome do gerenciador de filas da fila de mensagens para o qual a mensagem é destinada.

FEEDBACK (*Feedback* | * (padrão))

Quando o valor *MsgType* é MQFB_REPORT, *Feedback* descreve a natureza do relatório.

É possível usar nomes simbólicos. Por exemplo, é possível utilizar o nome simbólico MQFB_COA para identificar as mensagens na fila de mensagens não entregues que precisam de confirmação de chegada em suas filas de destino.

FORMAT (*Formato* | * (padrão))

O nome que o emissor da mensagem utiliza para descrever o formato dos dados da mensagem.

MSGTYPE (*MsgType* | * (padrão))

O tipo de mensagem da mensagem na fila de mensagens não entregues.

É possível usar nomes simbólicos. Por exemplo, é possível utilizar o nome simbólico MQMT_REQUEST para identificar as mensagens na fila de mensagens não entregues que precisam de respostas.

PERSIST (*Persistência* | * (padrão))

O valor da persistência da mensagem. (A persistência de uma mensagem determina se ela sobrevive a reinícios do gerenciador de filas.)

É possível usar nomes simbólicos. Por exemplo, é possível utilizar o nome simbólico MQPER_PERSISTENT para identificar mensagens na fila de mensagens não entregues que são persistentes.

REASON (*ReasonCode* | * (padrão))

O código de razão que descreve por que a mensagem foi colocada na fila de mensagens não entregues.

É possível usar nomes simbólicos. Por exemplo, é possível utilizar o nome simbólico MQRC_Q_FULL para identificar as mensagens colocadas na fila de mensagens não entregues porque suas filas de destino estavam cheias.

REPLYQ (*QueueName* | * (padrão))

O nome da fila de resposta especificada no descritor de mensagens, MQMD, da mensagem na fila de mensagens não entregues.

REPLYQM (*QueueManagerNome* | * (padrão))

O nome do gerenciador de filas da fila de resposta, conforme especificado no descritor de mensagens, MQMD, da mensagem na fila de mensagens não entregues.

USERID (*UserIdentifier* | * (padrão))

O ID do usuário do usuário que originou a mensagem na fila de mensagens não entregues, conforme especificado no descritor de mensagens, MQMD, da mensagem na fila de mensagens não entregues.

As palavras-chave de ação

As palavras-chave de ação utilizadas para descrever como uma mensagem correspondente deve ser processada são as seguintes:

ACTION (DISCARD|IGNORE|RETRY|FWD)

A ação a ser executada para qualquer mensagem na fila de mensagens não entregues que corresponde ao padrão definido nesta regra.

DESCARTAR

Exclua a mensagem da fila de mensagens não entregues.

IGNORAR

Deixe a mensagem na fila de mensagens não entregues.

TENTAR NOVAMENTE

Se a primeira tentativa de colocar a mensagem em sua fila de destino falhar, tente novamente. A palavra-chave RETRY define o número de tentativas feitas para implementar uma ação. A palavra-chave RETRYINT dos dados de controle controla o intervalo entre as tentativas.

AVANÇAR

Encaminhar a mensagem para a fila nomeada na palavra-chave FWDQ.

Deve-se especificar a palavra-chave ACTION.

FWDQ (QueueName|&DESTQ|&REPLYQ)

O nome da fila de mensagens para a qual redirecionar a mensagem quando ACTION (FWD) é solicitada.

QueueName

O nome de uma fila de mensagens. FWDQ(' ') não é válido.

&DESTQ

Utilize o nome da fila do campo *DestQName* na estrutura MQDLH.

&REPLYQ

Execute o nome da fila do campo *ReplyToQ* no descritor de mensagens, MQMD.

Para evitar mensagens de erro quando uma regra que especifica FWDQ (&REPLYQ) corresponde a uma mensagem com um campo *ReplyToQ* em branco, especifique REPLYQ (?*) no padrão de mensagem.

FWDQM (QueueManagerNome| & DESTQM | & REPLYQM | " (padrão)

O gerenciador de filas da fila para a qual encaminhar uma mensagem.

QueueManagerName

O nome do gerenciador de filas da fila para o qual encaminhar uma mensagem quando ACTION (FWD) é solicitada.

&DESTQM

Tome o nome do gerenciador de filas do campo *DestQMgrName* na estrutura MQDLH.

&REPLYQM

Tome o nome do gerenciador de filas do campo *ReplyToQMgr* no descritor de mensagens, MQMD.

..

FWDQM(' '), que é o valor padrão, identifica o gerenciador de filas locais.

HEADER (YES (padrão) |NO)

Se o MQDLH deve permanecer em uma mensagem para a qual ACTION (FWD) é solicitada. Por padrão, o MQDLH permanece na mensagem. A palavra-chave HEADER não é válida para ações diferentes de FWD.

PUTAUT (DEF (padrão) | CTX)

A autoridade com a qual as mensagens devem ser colocadas pelo manipulador de fila de mensagens não entregues:

DEF

Coloque mensagens com a autoridade do próprio manipulador de fila de mensagens não entregues.

CTX

Colocar as mensagens com a autoridade do ID do usuário no contexto da mensagem. Se você especificar PUTAUT (CTX), deve-se estar autorizado a assumir a identidade de outros usuários.

RETRY (*RetryCount*|1 (padrão))

O número de vezes, no intervalo 1 – 999.999.999, para tentar uma ação (no intervalo especificado na palavra-chave RETRYINT dos dados de controle). A contagem de tentativas feitas pelo manipulador de fila de mensagens não entregues para implementar qualquer regra particular é específica para a instância atual do manipulador de fila de mensagens não entregues; a contagem não persiste nas reinicializações. Se o manipulador de fila de mensagens não entregues for reiniciado, a contagem de tentativas feitas para aplicar uma regra é reconfigurada para zero.

Exemplo de regra

Aqui está uma regra de exemplo a partir de uma tabela de regras do manipulador de fila de mensagens não entregues:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Essa regra instrui o manipulador de fila de mensagens não entregues a fazer três tentativas para entregar à sua fila de destino persistente qualquer mensagem que foi colocada na fila de mensagens não entregues porque o MQPUT e MQPUT1 foram inibidos.

Todas as palavras-chave que podem ser utilizadas em uma regra são descritas no restante desta seção. Observe o seguinte :

- O valor padrão para uma palavra-chave, se houver, é sublinhado. Para a maioria das palavras-chave, o valor padrão é * (asterisco), que corresponde a qualquer valor.
- A linha vertical (|) separa alternativas, somente um dos quais pode ser especificado.
- Todas as palavras-chave, exceto ACTION, são opcionais.

Convenções de tabela de regras de DLQ

A sintaxe, a estrutura e o conteúdo da tabela de regras do manipulador de fila de devoluções devem seguir estas convenções.

A tabela de regras deve aderir às seguintes convenções:

- Uma tabela de regras deve conter pelo menos uma regra.
- As palavras-chave podem ocorrer em qualquer ordem.
- Uma palavra-chave pode ser incluída somente uma vez em qualquer regra.
- As palavras-chave não fazem distinção entre maiúsculas e minúsculas.
- Uma palavra-chave e seu valor de parâmetro devem ser separados de outras palavras-chave por pelo menos um espaço em branco ou uma vírgula.
- Pode haver qualquer número de espaços em branco no início ou final de uma regra e entre as palavras-chave, pontuação e valores.
- Cada regra deve iniciar em uma nova linha.
- Em sistemas Windows, a última regra na tabela deve terminar com um caractere de retorno de linha/alimentação de linha. É possível fazer isso assegurando que pressionar a tecla Enter no final da regra, para que a última linha da tabela é uma linha em branco.
- Por razões de portabilidade, o comprimento significativo de uma linha não deve ser maior que 72 caracteres.

- Utilize o sinal de mais (+) como o último caractere não em branco em uma linha para indicar que a regra continuará a partir do primeiro caractere não em branco na próxima linha. Utilize o sinal de menos (-) como o último caractere não em branco em uma linha para indicar que a regra continuará a partir do início da próxima linha. Caracteres de continuação podem ocorrer dentro de palavras-chave e parâmetros.

Por exemplo:

```
APPLNAME(' ABC+
D')
```

resulta em 'ABCD' e

```
APPLNAME(' ABC-
D')
```

resulta em 'ABC D'.

- Linhas de comentário, que começam com um asterisco (*), podem ocorrer em qualquer lugar na tabela de regras.
- Linhas em branco são ignoradas.
- Cada entrada na tabela de regras do manipulador de fila de mensagens não entregues é composta de uma ou mais palavras-chave e seus parâmetros associados. Os parâmetros devem seguir estas regras de sintaxe:
 - Cada valor de parâmetro deve incluir pelo menos um caractere significativo. O delimitador entre aspas simples em valores que são colocados entre aspas não são considerados significativos. Por exemplo, esses parâmetros são válidos:

FORMAT(' ABC ')	3 caracteres significativos
FORMAT(ABC)	3 caracteres significativos
FORMAT(' A ')	1 caractere significante
FORMAT(A)	1 caractere significante
FORMAT(' ')	1 caractere significante

Esses parâmetros são inválidos porque eles não contêm caracteres significativos:

```
FORMAT(' ' )
FORMAT( )
FORMAT( )
FORMAT
```

- Os caracteres curinga são suportados. É possível usar o ponto de interrogação (?) em vez de qualquer caractere único, exceto um espaço em branco; é possível usar o asterisco (*) em vez de zero ou mais caracteres adjacentes. O asterisco (*) e o ponto de interrogação (?) são **sempre** interpretados como caracteres curinga em valores de parâmetro.
- Caracteres curinga não podem ser incluídos nos parâmetros destas palavras-chave: ACTION, HEADER, RETRY, FWDQ, FWDQM e PUTAUT.
- Espaços em branco à direita em valores de parâmetros e nos campos correspondentes na mensagem na fila de mensagens não entregues não são significativos ao executar correspondências de curinga. No entanto, iniciais e espaços em branco incorporados dentro de sequências que são colocadas entre aspas simples são significativos para correspondência de curinga.
- Os parâmetros numéricos não podem incluir o caractere curinga ponto de interrogação (?). É possível utilizar o asterisco (*) em vez de um parâmetro numérico inteiro, mas não como parte de um parâmetro numérico. Por exemplo, esses são parâmetros numéricos válidos:

MSGTYPE(2)	Somente mensagens de resposta são elegíveis
MSGTYPE(*)	Qualquer tipo de mensagem é elegível
MSGTYPE(' * ')	Qualquer tipo de mensagem é elegível

No entanto, MSGTYPE(' 2* ') não é válido, pois ele inclui um asterisco (*) como parte de um parâmetro numérico.

- Parâmetros numéricos devem estar no intervalo 0-999 999 999. Se o valor do parâmetro for nesse intervalo, ele é aceito, mesmo se ele não for atualmente válido no campo para o qual a palavra-chave está relacionada. É possível utilizar nomes simbólicos para parâmetros numéricos.
- Se um valor de sequência é menor que o campo no MQDLH ou MQMD para a qual a palavra-chave se relaciona, o valor será preenchido com espaços em branco para o comprimento do campo. Se o valor, exceto asteriscos, é mais longo do que o campo, um erro foi diagnosticado. Por exemplo, esses são todos os valores de sequência válidos para um campo de 8 caracteres:

'ABCDEFGH'	8 caracteres
'A*C*E*G*I'	5 caracteres excluindo asteriscos
'*A*C*E*G*I*K*M*O* *'	8 caracteres excluindo asteriscos

- Coloque as sequências de caracteres que contêm espaços em branco, caracteres minúsculos ou caracteres especiais que não sejam ponto (.), barra (?), sublinhado (_) e sinal de porcentagem (%) entre aspas simples. Caracteres minúsculos não colocados entre uma única aspa simples são convertidos em letras maiúsculas. Se a sequência incluir uma citação, utilize duas aspas simples para indicar o início e o fim da citação. Quando o comprimento da sequência é calculado, cada ocorrência de aspas duplas é contada como um único caractere.

Como a tabela de regras de DLQ é processada

O manipulador da fila de devoluções procura na tabela de regras uma regra em que o padrão corresponda a uma mensagem na DLQ.

A procura começa com a primeira regra na tabela e continua sequencialmente através da tabela. Quando o manipulador de fila de mensagens não entregues localiza uma regra com um padrão correspondente, ele executa a ação dessa regra. O manipulador de fila de mensagens não entregues incrementa a contagem de novas tentativas para uma regra, 1 sempre que ela se aplica essa regra. Se a primeira tentativa falhar, o manipulador de fila de mensagens não entregues tenta novamente até que o número de tentativas corresponda ao número especificado na palavra-chave RETRY. Se todas as tentativas falharem, o manipulador de fila de mensagens não entregues procurará a próxima regra de correspondência na tabela.

Este processo é repetido para regras de correspondência subsequentes até que uma ação seja bem-sucedida. Quando cada regra de correspondência foi tentada o número de vezes especificado em sua palavra-chave RETRY e todas as tentativas falharam, ACTION (IGNORE) é assumido. ACTION (IGNORE) também é assumido se nenhuma regra correspondente for localizada.

Nota:

1. Padrões de regra de correspondência são procurados somente para mensagens na fila de mensagens não entregues que começam com um MQDLH. As mensagens que não começam com um MQDLH são relatadas periodicamente como estando em erro e permanecem na fila de mensagens não entregues indefinidamente.
2. Todas as palavras-chave padrão podem ter permissão para assumirem o padrão, de modo que uma regra pode consistir em uma única ação. Observe, no entanto, que as regras de ação somente são aplicadas a todas as mensagens na fila que possuem MQDLHs e que ainda não foram transformadas em conformidade com outras regras na tabela.
3. A tabela de regras é validada quando o manipulador de fila de mensagens não entregues é iniciado e os erros são sinalizados nesse momento. É possível fazer mudanças na tabela de regras a qualquer

momento, mas essas mudanças não entram em vigor até que o manipulador de fila de mensagens não entregues seja reiniciado.

4. O manipulador de fila de mensagens não entregues não altera o conteúdo de mensagens, o MQDLH ou do descritor de mensagens. O manipulador de fila de mensagens não entregues sempre coloca mensagens para outras filas com a opção de mensagem MQPMO_PASS_ALL_CONTEXT.
5. Erros de sintaxe consecutivos na tabela de regras não podem ser reconhecidos porque a tabela de regras é projetada para eliminar a geração de erros repetitiva durante a validação.
6. O manipulador de fila de mensagens não entregues abre a fila de mensagens não entregues com a opção MQOO_INPUT_AS_Q_DEF.
7. Várias instâncias do manipulador de fila de mensagens não entregues podem ser executadas simultaneamente em relação à mesma fila utilizando a mesma tabela de regras. No entanto, é mais comum haver um relacionamento de um para um entre uma fila de mensagens não entregues e um manipulador de fila de mensagens não entregues.

Conceitos relacionados

[Filas de Devoluções](#)

Tarefas relacionadas

[Resolução de problemas de mensagens não entregues](#)

  *Um exemplo de tabela de regras do manipulador DLQ*

Uma tabela de regras de fila de mensagens não entregues de exemplo para o comando **runmqdlq**, contendo uma única entrada de dados de controle e várias regras.

```
*****
*   An example rules table for the runmqdlq command   *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.
* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never do things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)
```

```
* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.
```

```
REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

```
* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it should be able
* to cope with the message being lost, so we can afford to
* discard the message. PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We don't have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.
```

```
REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)
```

```
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

Conceitos relacionados

[Filas de Devoluções](#)

Tarefas relacionadas

[Resolução de problemas de mensagens não entregues](#)

Referências relacionadas

[runmqdlq \(executar manipulador da fila de devoluções\)](#)

Chamando o manipulador de fila de devoluções no IBM i ..

No IBM MQ for IBM i, você chama o manipulador de DLQ configurando o comando **STRMQMDLQ**.

Antes de começar

Deve-se estar autorizado a acessar a DLQ sozinho e qualquer fila de mensagem para as quais as mensagens na DLQ são encaminhadas, para executar o manipulador DLQ. Deve-se também estar autorizado a assumir a identidade de outros usuários, para que a DLQ coloque as mensagens nas filas com a autoridade do ID do usuário no contexto da mensagem.

Nota: Geralmente é preferível evitar colocar as mensagens em uma DLQ. Para obter informações sobre como usar e evitar DLQs, consulte [“Trabalhando com filas de mensagens não entregues”](#) na página 157.

Sobre esta tarefa

Uma *fila de devoluções* (DLQ), às vezes conhecida como uma *fila de mensagens não entregues*, é uma fila de participação para mensagens que não podem ser entregues para suas filas de destino. Cada gerenciador de filas em uma rede deve ter uma DLQ associada.

Os gerenciadores de fila, agentes do canal de mensagens e aplicativos podem colocar as mensagens na DLQ. Todas as mensagens no DLQ devem ser prefixadas com uma estrutura de *cabeçalho de devoluções*, MQDLH. As mensagens colocadas na DLQ por um gerenciador de filas ou por um agente do canal de mensagem sempre possuem um MQDLH. Para aplicativos que colocam mensagens no DLQ, você deve fornecer um MQDLH.

O campo *Razão* da estrutura MQDLH contém um código de razão que identifica o motivo pelo qual a mensagem está no DLQ.

Em todos os ambientes do IBM MQ, deve haver uma rotina que é executada regularmente para processar mensagens na fila de mensagens não entregues. O IBM MQ fornece uma rotina padrão, chamada *manipulador de filas de devoluções* (o manipulador de DLQ), que você chama usando o comando **STRMQMDLQ**. Uma *tabela de regras* gravada pelo usuário fornece instruções para o manipulador DLQ, para as mensagens de processamento na DLQ. Ou seja, o manipulador DLQ corresponde as mensagens na DLQ às entradas na tabela de regras. Quando uma mensagem DLQ corresponder uma entrada na tabela de regras, o manipulador DLQ executará a ação associada com essa entrada.

Procedimento

- Chamar o manipulador de DLQ

Use o comando **STRMQMDLQ** para chamar o manipulador de DLQ. É possível nomear a DLQ que você deseja processar e o gerenciador de filas que você deseja usar de duas maneiras:

- Como parâmetros para **STRMQMDLQ** no prompt de comandos. Por exemplo:

```
STRMQMDLQ UDLMSGQ(ABC1.DEAD.LETTER.QUEUE) SRCMBR(QRULE) SRCFILE(library/QXTSRC)
MQMNAME(MY.QUEUE.MANAGER)
```

- Na tabela de regras. Por exemplo:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE)
```

Nota: A tabela de regras é um membro em um arquivo físico de origem que pode assumir qualquer nome.

Os exemplos se aplicam ao DLQ chamado ABC1.DEAD.LETTER.QUEUE, pertencente ao gerenciador de filas padrão.

Se você não especificar o DLQ ou o gerenciador de filas conforme mostrado, o gerenciador de filas padrão para a instalação será usado juntamente com o DLQ pertencente a esse gerenciador de filas.

Conceitos relacionados

[Filas de Devoluções](#)

Tarefas relacionadas

[Resolução de problemas de mensagens não entregues](#)

 *A tabela de regras do manipulador de DLQ no IBM i*

A tabela de regras do manipulador da fila de devoluções define como o manipulador de DLQ processa as mensagens que chegam na DLQ do IBM i.

A tabela de regras do manipulador de fila de mensagens não entregues define como o manipulador de fila de mensagens não entregues irá processar mensagens que chegam à fila de mensagens não entregues. Há dois tipos de entrada em uma tabela de regras:

- A primeira entrada na tabela, que é opcional, contém *dados de controle*.
- Todas as outras entradas na tabela são *regras* para o manipulador de fila de mensagens não entregues seguir. Cada regra consiste em um *padrão* (um conjunto de características da mensagem) em relação ao qual uma mensagem é correspondida e uma *ação* a ser tomada quando uma mensagem na fila de mensagens não entregues corresponder ao padrão especificado. Deve haver pelo menos uma regra em uma tabela de regras.

Cada entrada na tabela de regras é composta por uma ou mais palavras-chave.

Dados de controle

Essa seção descreve as palavras-chave que é possível incluir em uma entrada de dados de controle em uma tabela de regras do manipulador DLQ. Observe o seguinte :

- O valor padrão para uma palavra-chave, se houver, é sublinhado.
- A linha vertical (|) separa alternativas. É possível especificar somente um deles.
- Todas as palavras-chave são opcionais.

INPUTQ (*QueueName* | " (padrão)

O nome da fila de mensagens não entregues que você deseja processar:

1. Qualquer valor UDLMMSGQ (ou *DFT) que você especificar como um parâmetro para o comando **STRMQMDLQ** substitui qualquer valor INPUTQ na tabela de regras.
2. Se você especificar um valor UDLMMSGQ em branco como um parâmetro para o comando **STRMQMDLQ**, o valor INPUTQ na tabela de regras é utilizado.
3. Se você especificar um valor UDLMMSGQ em branco como um parâmetro para o comando **STRMQMDLQ** e um valor INPUTQ em branco na tabela de regras, a fila de mensagens não entregues padrão do sistema será utilizada.

INPUTQM (*QueueManagerNome* | " (padrão)

O nome do gerenciador de filas que possui a fila de mensagens não entregues nomeada na palavra-chave INPUTQ.

Se você não especificar um gerenciador de filas ou se você especificar INPUTQM(' ') na tabela de regras, o sistema utiliza o gerenciador de filas padrão para a instalação.

RETRYINT (*Intervalo* | 60 (padrão)

O intervalo, em segundos, no qual o manipulador da fila de mensagens não entregues deve tentar reprocessar as mensagens na fila de mensagens não entregues que não puderam ser processadas na primeira tentativa e para o qual as tentativas repetidas foram solicitadas. Por padrão, o intervalo de novas tentativas é 60 segundos.

WAIT (YES (padrão) |NO|*nnn*)

Se o manipulador de fila de mensagens não entregues deve aguardar a chegada de novas mensagens na fila de mensagens não entregues quando ele detecta que não há mensagens adicionais que possa processar.

SIM

Faz com que o manipulador da fila de mensagens não entregues aguarde indefinidamente.

NÃO

Faz com que o manipulador da fila de mensagens não entregues finalize quando ele detectar que a fila de mensagens não entregues está vazia ou não contém mensagens que ele pode processar.

nnn

Faz com que o manipulador da fila de mensagens não entregues aguarde *nnn* segundos pela chegada do novo trabalho antes de finalizar após detectar que a fila está vazia ou que não contém mensagens que possa processar.

Especifique WAIT (YES) para DLQs ocupadas e WAIT (NO) ou WAIT (*nnn*) para DLQs que possuem um baixo nível de atividade. Se o manipulador da fila de mensagens não entregues tem permissão para encerrar, chame-o novamente usando o acionamento.

É possível fornecer o nome da fila de mensagens não entregues como um parâmetro de entrada para o comando **STRMQMDLQ**, como uma alternativa para incluir dados de controle na tabela de regras. Se algum valor for especificado na tabela de regras e na entrada para o comando **STRMQMDLQ**, o valor especificado no comando **STRMQMDLQ** terá precedência.

Nota: Se uma entrada de dados de controle estiver incluída na tabela de regras, ela deverá ser a primeira entrada na tabela.

Uma descrição dos padrões e ações para cada uma das regras de fila de mensagens não entregues do IBM i.

Aqui está uma regra de exemplo a partir de uma tabela de regras do manipulador de fila de mensagens não entregues:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +
ACTION (RETRY) RETRY (3)
```

Esta regra instrui o manipulador de fila de mensagens não entregues para fazer 3 tentativas para entregar a sua fila de destino persistente qualquer mensagem que foi colocada na fila de mensagens não entregues porque MQPUT e MQPUT1 foram inibidas.

Esta seção descreve as palavras-chave que podem ser incluídas em uma regra. Observe o seguinte :

- O valor padrão para uma palavra-chave, se houver, é sublinhado. Para a maioria das palavras-chave, o valor padrão é * (asterisco), que corresponde a qualquer valor.
- A linha vertical (|) separa alternativas. É possível especificar somente um deles.
- Todas as palavras-chave, exceto ACTION, são opcionais.

Esta seção começa com uma descrição das palavras-chave de correspondência de padrões (aquelas para as quais as mensagens na fila de mensagens não entregues são correspondidas). Em seguida, descreve as palavras-chave de ação (aquelas que determinam como o manipulador de fila de mensagens não entregues deve processar uma mensagem correspondente).

As palavras-chave de correspondência de padrões são descritas em um exemplo. Use essas palavras-chave para especificar os valores com relação aos quais as mensagens na fila de mensagens não entregues do IBM i são correspondidas. Todas as palavras-chave de correspondência de padrões são opcionais.

APPLIDAT (*ApplIdentityDados*|* (padrão))

O valor *ApplIdentityData* da mensagem na fila de mensagens não entregues, especificados no descritor de mensagens, MQMD.

APPLNAME (*PutApplNome*|* (padrão))

O nome do aplicativo que emitiu a chamada MQPUT ou MQPUT1, conforme especificado no campo *PutApplName* do descritor de mensagens, MQMD, da mensagem na fila de mensagens não entregues.

APPLTYPE (*PutApplTipo*|* (padrão))

O valor *PutApplType* especificado no descritor de mensagens, MQMD, da mensagem na fila de mensagens não entregues.

DESTQ (*QueueName*|* (padrão))

O nome da fila de mensagens para a qual a mensagem é destinada.

DESTQM (*QueueManagerNome*|* (padrão))

O nome do gerenciador de filas para a fila de mensagens para o qual a mensagem é destinada.

FEEDBACK (*Feedback*|* (padrão))

Quando o valor *MsgType* é MQMT_REPORT, *Feedback* descreve a natureza do relatório.

É possível usar nomes simbólicos. Por exemplo, é possível utilizar o nome simbólico MQFB_COA para identificar as mensagens na fila de mensagens não entregues que requerem confirmação de chegada em suas filas de destino.

FORMAT (*Formato*|* (padrão))

O nome que o emissor da mensagem utiliza para descrever o formato dos dados da mensagem.

MSGTYPE (*MsgType*|* (padrão))

O tipo de mensagem da mensagem na fila de mensagens não entregues.

É possível usar nomes simbólicos. Por exemplo, é possível utilizar o nome simbólico MQMT_REQUEST para identificar as mensagens na fila de mensagens não entregues que requerem respostas.

PERSIST (*Persistência* | * (padrão))

O valor da persistência da mensagem. (A persistência de uma mensagem determina se ela sobrevive a reinícios do gerenciador de filas.)

É possível usar nomes simbólicos. Por exemplo, é possível utilizar o nome simbólico MQPER_PERSISTENT para identificar as mensagens na fila de mensagens não entregues que são persistentes.

REASON (*ReasonCode* | * (padrão))

O código de razão que descreve por que a mensagem foi colocada na fila de mensagens não entregues.

É possível usar nomes simbólicos. Por exemplo, é possível utilizar o nome simbólico MQRC_Q_FULL para identificar as mensagens colocadas na fila de mensagens não entregues porque suas filas de destino estavam cheias.

REPLYQ (*QueueName* | * (padrão))

O nome da fila de resposta especificada no descritor de mensagens, MQMD, da mensagem na fila de mensagens não entregues.

REPLYQM (*QueueManagerName* | * (padrão))

O nome do gerenciador de filas da fila de resposta especificado na palavra-chave REPLYQ.

USERID (*UserIdentifier* | * (padrão))

O ID do usuário do usuário que originou a mensagem na fila de mensagens não entregues, conforme especificado no descritor de mensagens, MQMD.

IBM i *Palavras-chave de ação de DLQ no IBM i*

Use estas palavras-chave de ação da fila de mensagens não entregues para determinar como uma mensagem correspondente na fila de mensagens não entregues do IBM i é processada.

ACTION (DISCARD|IGNORE|RETRY|FWD)

A ação tomada para qualquer mensagem na fila de mensagens não entregues que corresponde ao padrão definido nesta regra.

DESCARTAR

Faz com que a mensagem seja excluída da fila de mensagens não entregues.

IGNORAR

Faz com que a mensagem seja mantida no DLQ.

TENTAR NOVAMENTE

Faz com que o manipulador de fila de mensagens não entregues tente novamente colocar a mensagem em sua fila de destino.

AVANÇAR

Faz com que o manipulador de fila de mensagens não entregues redirecione a mensagem para a fila denominada na palavra-chave FWDQ.

Deve-se especificar a palavra-chave ACTION. O número de tentativas feitas para implementar uma ação é controlado pela palavra-chave RETRY. A palavra-chave RETRYINT dos dados de controle controla o intervalo entre as tentativas.

FWDQ (*QueueName* | &DESTQ | &REPLYQ)

O nome da fila de mensagens para a qual a mensagem é encaminhada quando você seleciona a palavra-chave ACTION.

QueueName

O nome de uma fila de mensagens. FWDQ(' ') não é válido.

&DESTQ

Utilize o nome da fila do campo *DestQName* na estrutura MQDLH.

&REPLYQ

Execute o nome da fila do campo *ReplyToQ* no descritor de mensagens, MQMD.

É possível especificar REPLYQ (?*) no padrão de mensagem para evitar mensagens de erro, quando uma regra especificando FWDQ (&REPLYQ) corresponde a uma mensagem com um campo *ReplyToQ* com espaço em branco.

FWDQM (*QueueManagerName* | & DESTQM | & REPLYQM | " (padrão)

O gerenciador de filas da fila para o qual uma mensagem será redirecionada.

QueueManagerName

O nome do gerenciador de filas para a fila para a qual a mensagem é encaminhada quando você seleciona a palavra-chave ACTION (FWD).

&DESTQM

Tome o nome do gerenciador de filas do campo *DestQMGrName* na estrutura MQDLH.

&REPLYQM

Tome o nome do gerenciador de filas do campo *ReplyToQMGr* no descritor de mensagens, MQMD.

..

FWDQM(' '), que é o valor padrão, identifica o gerenciador de filas locais.

HEADER (YES (padrão) |NO)

Se o MQDLH deve permanecer em uma mensagem para a qual ACTION (FWD) é solicitada. Por padrão, o MQDLH permanece na mensagem. A palavra-chave HEADER não é válida para ações diferentes de FWD.

PUTAUT (DEF (padrão) | CTX)

A autoridade com a qual as mensagens devem ser colocadas pelo manipulador de fila de mensagens não entregues:

DEF

Coloca as mensagens com a autoridade do próprio manipulador de fila de mensagens não entregues.

CTX

Faz com que as mensagens sejam colocadas com a autoridade do ID do usuário no contexto da mensagem. Deve-se estar autorizado a assumir a identidade de outros usuários, se você especificar PUTAUT (CTX).

RETRY (*RetryCount*|1 (padrão)

O número de vezes, no intervalo 1 – 999.999.999, para tentar uma ação (no intervalo especificado na palavra-chave RETRYINT dos dados de controle).

Nota: A contagem de tentativas feitas pelo manipulador de fila de mensagens não entregues para implementar qualquer regra particular é específica para a instância atual do manipulador de fila de mensagens não entregues; a contagem não persiste nas reinicializações. Se você reiniciar o manipulador de fila de mensagens não entregues, a contagem de tentativas feitas para aplicar uma regra é reconfigurada para zero.



Convenções de tabela de regras de DLQ no IBM i

A tabela de regras de fila de mensagens não entregues do IBM i deve aderir às convenções específicas sobre sua sintaxe, estrutura e conteúdo.

- Uma tabela de regras deve conter pelo menos uma regra.
- As palavras-chave podem ocorrer em qualquer ordem.
- Uma palavra-chave pode ser incluída somente uma vez em qualquer regra.
- As palavras-chave não são sensíveis a maiúsculas e minúsculas.
- Uma palavra-chave e seu valor de parâmetro devem ser separados de outras palavras-chave por pelo menos um espaço em branco ou uma vírgula.

- Pode ocorrer qualquer número de espaços em branco no início ou término de uma regra e entre palavras-chave, pontuação e valores.
- Cada regra deve iniciar em uma nova linha.
- Para portabilidade, o comprimento significativo de uma linha não deve ser maior que 72 caracteres.
- Utilize o sinal de mais (+) como o último caractere não em branco em uma linha para indicar que a regra continuará a partir do primeiro caractere não em branco na próxima linha. Utilize o sinal de menos (-) como o último caractere não em branco em uma linha para indicar que a regra continuará a partir do início da próxima linha. Caracteres de continuação podem ocorrer dentro de palavras-chave e parâmetros.

Por exemplo:

```
APPLNAME (' ABC+
D')
```

results in 'ABCD'.

```
APPLNAME (' ABC-
D')
```

resulta em 'ABC D'.

- Linhas de comentário, que começam com um asterisco (*), podem ocorrer em qualquer lugar na tabela de regras.
- Linhas em branco são ignoradas.
- Cada entrada na tabela de regras do manipulador de fila de mensagens não entregues é composta de uma ou mais palavras-chave e seus parâmetros associados. Os parâmetros devem seguir estas regras de sintaxe:
 - Cada valor de parâmetro deve incluir pelo menos um caractere significativo. As aspas delimitadoras em valores colocados entre aspas não são consideradas significativas. Por exemplo, esses parâmetros são válidos:

FORMAT (' ABC ')	3 caracteres significativos
FORMAT (ABC)	3 caracteres significativos
FORMAT (' A ')	1 caractere significante
FORMAT (A)	1 caractere significante
FORMAT (' ')	1 caractere significante

Esses parâmetros são inválidos porque eles não contêm caracteres significativos:

```
FORMAT ( ' ' )
FORMAT ( )
FORMAT ( )
FORMAT
```

- Os caracteres curinga são suportados. É possível usar o ponto de interrogação (?) no lugar de qualquer caractere único, exceto um espaço em branco à direita. É possível utilizar o asterisco (*) no lugar de zero ou mais caracteres adjacentes. O asterisco (*) e o ponto de interrogação (?) são **sempre** interpretados como caracteres curinga em valores de parâmetro.
- Não é possível incluir caracteres curinga nos parâmetros dessas palavras-chave: ACTION, HEADER, RETRY, FWDQ, FWDQM e PUTAUT.
- Espaços em branco à direita em valores de parâmetros e nos campos correspondentes na mensagem na fila de mensagens não entregues não são significativos ao executar correspondências de curinga.

No entanto, iniciais e espaços em branco incorporados dentro de sequências entre aspas são significativos para correspondências de curinga.

- Os parâmetros numéricos não podem incluir o caractere curinga ponto de interrogação (?). É possível incluir o asterisco (*) no lugar de um parâmetro numérico inteiro, mas o asterisco não pode ser incluído como parte de um parâmetro numérico. Por exemplo, esses são parâmetros numéricos válidos:

MSGTYPE (2)	Somente mensagens de resposta são elegíveis
MSGTYPE (*)	Qualquer tipo de mensagem é elegível
MSGTYPE ('*')	Qualquer tipo de mensagem é elegível

No entanto, MSGTYPE ('2*') não é válido, pois ele inclui um asterisco (*) como parte de um parâmetro numérico.

- Parâmetros numéricos devem estar no intervalo 0-999 999 999. Se o valor do parâmetro for nesse intervalo, ele é aceito, mesmo se ele não for atualmente válido no campo para o qual a palavra-chave está relacionada. É possível utilizar nomes simbólicos para parâmetros numéricos.
- Se um valor de sequência é menor que o campo no MQDLH ou MQMD para a qual a palavra-chave se relaciona, o valor será preenchido com espaços em branco para o comprimento do campo. Se o valor, exceto asteriscos, é mais longo do que o campo, um erro foi diagnosticado. Por exemplo, estas são todos os valores de sequência válidos para um campo de 8 caracteres:

'ABCDEFGH'	8 caracteres
'A*C*E*G*I'	5 caracteres excluindo asteriscos
'*A*C*E*G*I*K*M*O*'	8 caracteres excluindo asteriscos

- Sequências que contêm espaços em branco, caracteres minúsculos ou caracteres especiais que não sejam ponto (.), barra (?), sublinhado (_) e sinal de porcentagem (%) devem ser colocadas entre aspas simples. Caracteres minúsculos não colocados entre aspas simples são convertidos em letras maiúsculas. Se a sequência incluir uma aspa, duas aspas simples devem ser utilizadas para indicar o início e o fim da citação. Quando o comprimento da sequência é calculado, cada ocorrência de aspas duplas é contada como um único caractere.



Como a tabela de regras DLQ é processada no IBM i

O manipulador da fila de devoluções procura na tabela de regras uma regra com um padrão que corresponda a uma mensagem na fila de mensagens não entregues do IBM i.

A procura começa com a primeira regra na tabela e continua sequencialmente através da tabela. Quando uma regra com um padrão correspondente for localizada, a tabela de regras tenta a ação a essa regra. O manipulador de fila de mensagens não entregues incrementa a contagem de novas tentativas para uma regra pelo 1 sempre que ela tenta aplicar essa regra. Se a primeira tentativa falhar, a tentativa será repetida até que a contagem de tentativas feitas corresponda ao número especificado na palavra-chave RETRY. Se todas as tentativas falharem, o manipulador de fila de mensagens não entregues procurará a próxima regra de correspondência na tabela.

Este processo é repetido para regras de correspondência subsequentes até que uma ação seja bem-sucedida. Quando cada regra de correspondência foi tentada o número de vezes especificado em sua palavra-chave RETRY e todas as tentativas falharam, ACTION (IGNORE) é assumido. ACTION (IGNORE) também é assumido se nenhuma regra correspondente for localizada.

Nota:

1. Padrões de regra de correspondência são procurados somente para mensagens na fila de mensagens não entregues que começam com um MQDLH. As mensagens que não começam com um MQDLH são relatadas periodicamente como estando em erro e permanecem na fila de mensagens não entregues indefinidamente.
2. Todas as palavras-chave padrão podem assumir o padrão, de modo que uma regra pode consistir em uma única ação. Observe, no entanto, que as regras de ação somente são aplicadas a todas as

mensagens na fila que possuem MQDLHs e que ainda não foram transformadas em conformidade com outras regras na tabela.

3. A tabela de regras é validada quando o manipulador de fila de mensagens não entregues é iniciado e os erros sinalizados nesse momento. (Mensagens de erro emitidas pelo manipulador de fila de mensagens não entregues são descritas em [Mensagens e códigos de razão](#).) É possível fazer mudanças na tabela de regras a qualquer momento, mas essas mudanças não entram em vigor até que o manipulador de fila de mensagens não entregues seja reiniciado.
4. O manipulador de fila de mensagens não entregues não altera o conteúdo de mensagens, do MQDLH ou do descritor de mensagens. O manipulador DLQ sempre coloca mensagens em outras filas com a opção de mensagem **MQPMO_PASS_ALL_CONTEXT**.
5. Erros de sintaxe consecutivos na tabela de regras não podem ser reconhecidos porque a validação da tabela de regras elimina a geração de erros repetitivos.
6. O manipulador DLQ abre o DLQ com a opção **MQOO_INPUT_AS_Q_DEF**.
7. Várias instâncias do manipulador de fila de mensagens não entregues podem ser executadas simultaneamente em relação à mesma fila utilizando a mesma tabela de regras. No entanto, é mais comum haver um relacionamento de um para um entre uma fila de mensagens não entregues e um manipulador de fila de mensagens não entregues.

IBM i

Um exemplo de tabela de regras do manipulador de DLQ no IBM i

Código de exemplo para uma tabela de regras do manipulador da fila de devoluções no IBM i. Esse exemplo de tabela de regras contém uma única entrada de dados de controle e diversas regras.

```
*****
* An example rules table for the STRMQMDLQ command *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* STRMQMDLQ, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to STRMQMDLQ,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation is always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never does things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
```



```

action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.

REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)

* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it must be able
* to cope with the message being lost, so we can afford to
* discard the message.

PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)

* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We do not have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.

REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)

ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)

```

Assegurado que todas as mensagens DLQ sejam processadas

O manipulador da fila de devoluções mantém um registro de todas as mensagens na DLQ que foram vistas, mas não removidas. Assegure-se de que o DLQ contenha o mínimo de mensagens possível.

Sobre esta tarefa

Se você utilizar o manipulador de fila de mensagens não entregues como um filtro para extrair um pequeno subconjunto das mensagens da fila de mensagens não entregues, o manipulador de fila de mensagens não entregues ainda manterá um registro dessas mensagens na fila de mensagens não entregues que ele não processou. Além disso, o manipulador de fila de mensagens não entregues não pode garantir que as novas mensagens que chegam à fila de mensagens não entregues sejam vistas, mesmo se a fila de mensagens não entregues estiver definida como primeiro a entrar, primeiro a sair (FIFO). Se a fila não estiver vazia, a fila de mensagens não entregues periodicamente será novamente varrida para verificar todas as mensagens.

Por essas razões, você deve assegurar que o DLQ contenha o mínimo de mensagens possível. Se as mensagens que não podem ser descartadas ou redirecionadas para outras filas (por qualquer razão) têm permissão para acumular na fila, a carga aumenta o manipulador de fila de mensagens não entregues e a própria fila de mensagens não entregues corre o risco de preenchimento.

Para ativar o manipulador de DLQ para esvaziar o DLQ, execute as seguintes medidas:

Procedimento

- Para mensagens que você ignoraria de outra forma, use uma ação que mova as mensagens para outra fila.

Tente não usar o comando **ACTION (IGNORE)**, que deixa mensagens no DLQ-e lembre-se de que **ACTION (IGNORE)** é assumido para mensagens que não são explicitamente endereçadas por outras regras na tabela. Em vez disso, utilize uma ação que mova as mensagens para outra fila. Por exemplo:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

- Torne a regra final na tabela "catch-all" para processar mensagens que não foram tratadas por regras anteriores na tabela.

Por exemplo, a regra final na tabela pode ser algo como o seguinte:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Isso encaminha as mensagens que passam pela regra final na tabela para a fila `REALLY.DEAD.QUEUE`, na qual podem ser processadas manualmente. Se você não tiver essa regra, as mensagens devem permanecer na fila de mensagens não entregues indefinidamente.

Trabalhando com tópicos administrativos

Use os comandos MQSC para gerenciar tópicos administrativos.

Consulte [Comandos MQSC](#) para obter informações detalhadas sobre esses comandos.

Conceitos relacionados

[Objetos de Tópico Administrativo](#)

Tarefas relacionadas

[“Definindo um tópico administrativo” na página 178](#)

Use o comando do MQSC **DEFINE TOPIC** para criar um tópico administrativo. Ao definir um tópico administrativo, é possível, opcionalmente, configurar cada atributo do tópico.

[“Exibindo atributos do objeto do tópico administrativo” na página 179](#)

Use o comando do MQSC **DISPLAY TOPIC** para exibir um objeto do tópico administrativo.

[“Mudando atributos de tópico administrativo” na página 180](#)

É possível alterar os atributos do tópico de duas maneiras, usando o comando **ALTER TOPIC** ou o comando **DEFINE TOPIC** com o atributo **REPLACE**

[“Copiando uma definição de tópico administrativo” na página 180](#)

É possível copiar uma definição de tópico utilizando o atributo **LIKE** no comando **DEFINE**.

[“Excluindo uma definição de tópico administrativo” na página 181](#)

É possível utilizar o comando MQSC **DELETE TOPIC** para excluir um tópico administrativo.

Definindo um tópico administrativo

Use o comando do MQSC **DEFINE TOPIC** para criar um tópico administrativo. Ao definir um tópico administrativo, é possível, opcionalmente, configurar cada atributo do tópico.

Antes de começar

Nota: O exemplo nesta tarefa requer que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Sobre esta tarefa

Qualquer atributo do tópico que não é explicitamente configurado é herdado do tópico administrativo padrão, `SYSTEM.DEFAULT.TOPIC`, que foi criado quando a instalação do sistema foi instalado.

Exemplo

Por exemplo, o comando **DEFINE TOPIC** que segue, define um tópico chamado `ORANGE.TOPIC` com estas características:

- Resolve para o `ORANGE` sequência de tópicos. Para obter informações sobre como as sequências de tópicos podem ser usadas, consulte [Combinando sequências de tópicos](#).

- Qualquer atributo que é configurado para ASPARENT utiliza o atributo conforme definido pelo tópico pai deste tópico. Essa ação é repetida a árvore de tópicos tanto quanto o tópico raiz, SYSTEM.BASE.TOPIC está localizado. Para obter mais informações, consulte [Árvores de tópico](#).

```
DEFINE TOPIC (ORANGE.TOPIC) +
TOPICSTR (ORANGE) +
DEFPRTY(ASPARENT) +
NPMSGDLV(ASPARENT)
```

Nota:

- Exceto para o valor da sequência de tópico, todos os valores de atributos mostrados são os valores padrão. Eles são mostrados aqui somente como uma ilustração. É possível omiti-los se você estiver certo de que os padrões são o que você deseja ou não foram mudados. Consulte também [“Exibindo atributos do objeto do tópico administrativo”](#) na página 179.
- Se você já tiver um tópico administrativo no mesmo gerenciador de filas com o nome ORANGE.TOPIC, este comando falhará. Use o atributo REPLACE se você deseja sobrescrever a definição existente de um tópico, mas veja também [“Mudando atributos de tópico administrativo”](#) na página 180.

Referências relacionadas

[DEFINE TOPIC](#)

Exibindo atributos do objeto do tópico administrativo

Use o comando do MQSC **DISPLAY TOPIC** para exibir um objeto do tópico administrativo.

Antes de começar

Nota: Os exemplos nesta tarefa requerem que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Exemplo

Este comando exibe todos os tópicos:

```
DISPLAY TOPIC(ORANGE.TOPIC)
```

É possível exibir atributos seletivamente especificando-os individualmente com o comando **DISPLAY TOPIC**. Por exemplo:

```
DISPLAY TOPIC(ORANGE.TOPIC) +
TOPICSTR +
DEFPRTY +
NPMSGDLV
```

Esse comando exibe os três atributos especificados:

```
AMQ8633: Display topic details.
TOPIC(ORANGE.TOPIC)                TYPE(LOCAL)
TOPICSTR(ORANGE)                    DEFPRTY(ASPARENT)
NPMSGDLV(ASPARENT)
```

Para exibir os valores ASPARENT do tópico conforme eles são usados no Tempo de Execução, use o comando **DISPLAY TPSTATUS**. Por exemplo, use:

```
DISPLAY TPSTATUS(ORANGE) DEFPRTY NPMSGDLV
```

O comando exibe os seguintes detalhes:

```
AMQ8754: Display topic status details.  
TOPICSTR(ORANGE)          DEFPRTY(0)  
NPMSGDLV(ALLAVAIL)
```

Ao definir um tópico administrativo, ele assume quaisquer atributos que você não especificar explicitamente do tópico administrativo padrão, que é chamado SYSTEM.DEFAULT.TOPIC. Para ver quais esses atributos padrão são, utilize o seguinte comando:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

Referências relacionadas

[DISPLAY TOPIC](#)

[DISPLAY TPSTATUS](#)

Mudando atributos de tópico administrativo

É possível alterar os atributos do tópico de duas maneiras, usando o comando **ALTER TOPIC** ou o comando **DEFINE TOPIC** com o atributo **REPLACE**

Antes de começar

Nota: Os exemplos nesta tarefa requerem que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Exemplo

Se, por exemplo, você deseja alterar a prioridade padrão de mensagens entregues para um tópico chamado ORANGE.TOPIC, para ser 5, use um dos comandos a seguir:

- Usando o comando **ALTER**:

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

Esse comando muda um único atributo, que a prioridade padrão de mensagens entregues a este tópico para 5; todos os outros atributos permanecem os mesmos.

- Usando o comando **DEFINE**:

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Este comando muda a prioridade padrão das mensagens entregues para este tópico. Todos os outros atributos recebem valores padrão.

Se você alterar a prioridade de mensagens enviadas para este tópico, as mensagens existentes não serão afetadas. No entanto, qualquer nova mensagem usa a prioridade especificada se não for fornecida pelo aplicativo de publicação.

Referências relacionadas

[ALTER TOPIC](#)

[DISPLAY TOPIC](#)

Copiando uma definição de tópico administrativo

É possível copiar uma definição de tópico utilizando o atributo LIKE no comando **DEFINE**.

Antes de começar

Nota: Os exemplos nesta tarefa requerem que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Exemplo

O comando a seguir cria um tópico, MAGENTA.TOPIC, com os mesmos atributos do tópico original, ORANGE.TOPIC, em vez dos tópicos administrativos padrão do sistema. Digite o nome do tópico a ser copiado exatamente como foi digitado quando você criou o tópico. Se o nome contiver caracteres minúsculos, coloque o nome entre aspas simples.

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

Também é possível utilizar este formulário do comando **DEFINE** para copiar uma definição de tópico, mas faça mudanças aos atributos do original. Por exemplo:

```
DEFINE TOPIC (BLUE.TOPIC) +  
TOPICSTR (BLUE) +  
LIKE (ORANGE.TOPIC)
```

Também é possível copiar os atributos do tópico BLUE.TOPIC para o tópico GREEN.TOPIC e especificar que quando as publicações não podem ser entregues para sua fila de assinantes correta que não são colocados na fila de mensagens não entregues. Por exemplo:

```
DEFINE TOPIC (GREEN.TOPIC) +  
TOPICSTR (GREEN) +  
LIKE (BLUE.TOPIC) +  
USEDLQ (NO)
```

Referências relacionadas

[DEFINE TOPIC](#)

Excluindo uma definição de tópico administrativo

É possível utilizar o comando MQSC **DELETE TOPIC** para excluir um tópico administrativo.

Antes de começar

Nota: O exemplo nesta tarefa requer que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Exemplo

```
DELETE TOPIC (ORANGE.TOPIC)
```

Os aplicativos não estarão mais aptos a abrir o tópico para publicação ou criar novas assinaturas usando o nome do objeto, ORANGE.TOPIC. Aplicativos de publicação que possuem a abertura do tópico estão aptos a continuar publicando a sequência de tópicos resolvidos. Todas as assinaturas já feitas neste tópico continuam recebendo publicações após a exclusão do tópico.

Os aplicativos que não estão referenciando esse objeto do tópico, mas estão utilizando a sequência de tópico resolvida que este objeto do tópico representado, 'ORANGE' neste exemplo, continuam a funcionar. Neste caso eles herdam as propriedades de um objeto do tópico mais alto na árvore de tópicos. Para obter mais informações, consulte [Árvores de tópico](#).

Referências relacionadas

[DELETE TOPIC](#)

Trabalhando com assinaturas

Use os comandos MQSC para gerenciar assinaturas.

Sobre esta tarefa

As assinaturas podem ser um dos três tipos, definido no atributo **SUBTYPE**:

ADMIN

Administrativamente definido por um usuário.

PROXY

Uma assinatura criada internamente para roteamento de publicações entre gerenciadores de filas.

API

Criado programaticamente, por exemplo, usando a chamada MQI MQSUB.

Consulte [Comandos MQSC](#) para obter informações detalhadas sobre esses comandos.

Definindo uma assinatura administrativa

Use o comando do MQSC **DEFINE SUB** para criar uma assinatura administrativa. Também é possível utilizar o padrão definido na definição de assinatura local padrão. Ou, é possível modificar as características de assinatura a partir desses da assinatura local padrão, SYSTEM.DEFAULT.SUB que foi criado quando o sistema foi instalado.

Antes de começar

Nota: Os exemplos nesta tarefa requerem que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Exemplo

O comando **DEFINE SUB** a seguir define uma assinatura chamada ORANGE com estas características:

- Assinatura durável, o que significa que ele persistir durante a reinicialização do gerenciador de filas, com expiração ilimitada.
- Receber as publicações feitas na sequência de tópico ORANGE, com as prioridades de mensagens conforme definido pelos aplicativos de publicação.
- Publicações entregues para essa assinatura são enviadas para a fila local SUBQ, essa fila deve ser definida antes da definição da assinatura.

```
DEFINE SUB (ORANGE) +  
TOPICSTR (ORANGE) +  
DESTCLAS (PROVIDED) +  
DEST (SUBQ) +  
EXPIRY (UNLIMITED) +  
PUBPRTY (AS PUB)
```

Nota:

- O nome de sequência de assinatura e tópico não precisam corresponder.
- Exceto para os valores da sequência de destino e tópico, todos os valores de atributos mostrados são os valores padrão. Eles são mostrados aqui somente como uma ilustração. É possível omiti-los se você estiver certo de que os padrões são o que você deseja ou não foram mudados. Consulte também [“Exibindo atributos de assinaturas” na página 183](#).
- Se você já tiver uma assinatura local no mesmo gerenciador de filas com o nome ORANGE, esse comando falhará. Use o atributo **REPLACE** se você desejar sobrescrever a definição existente de uma fila, mas consulte também [“Mudando atributos de assinatura local” na página 184](#)
- Se o SUBQ de filas não existir, este comando falhará.

Referências relacionadas

[DEFINE SUB](#)

Exibindo atributos de assinaturas

É possível utilizar o comando **DISPLAY SUB** para exibir os atributos configurados de qualquer assinatura conhecidos para o gerenciador de filas.

Antes de começar

Nota: Os exemplos nesta tarefa requerem que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Exemplo

```
DISPLAY SUB(ORANGE)
```

É possível exibir seletivamente os atributos especificando-os individualmente. Por exemplo:

```
DISPLAY SUB(ORANGE) +  
SUBID +  
TOPICSTR +  
DURABLE
```

Este comando exibe os três atributos especificados da seguinte forma:

```
AMQ8096: IBM MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03) TOPICSTR(ORANGE)  
SUB(ORANGE)  
DURABLE(YES)
```

TOPICSTR é a sequência de tópicos resolvidos no qual esse assinante está operando. Quando uma assinatura é definida para usar um objeto do tópico, a sequência de tópicos a partir desse objeto é usada como um prefixo para a sequência de tópicos fornecida ao fazer a assinatura. SUBID é um identificador exclusivo designado pelo gerenciador de filas quando uma assinatura for criada. Este é um atributo útil para exibição porque alguns nomes de assinatura podem ser longos ou estar em um conjunto de caracteres diferentes para os quais pode se tornar inviável.

Um método alternativo para exibir assinaturas é utilizar o SUBID:

```
DISPLAY SUB +  
SUBID(414D51204141412020202020202020EE921E4E20002A03) +  
TOPICSTR +  
DURABLE
```

Esse comando fornece a mesma saída de antes:

```
AMQ8096: IBM MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03) TOPICSTR(ORANGE)  
SUB(ORANGE)  
DURABLE(YES)
```

As assinaturas de proxy em um gerenciador de filas não são exibidas por padrão. Para exibi-los especificar um **SUBTYPE** de PROXY ou ALL.

É possível utilizar o comando [DISPLAY SBSTATUS](#) para exibir os atributos de Execução. Por exemplo, utilize o comando:

```
DISPLAY SBSTATUS(ORANGE) NUMMSGS
```

A seguinte saída é exibida:

```
AMQ8099: IBM MQ subscription status inquired.
```

```
SUB(ORANGE)
SUBID(414D51204141412020202020202020EE921E4E20002A03)
NUMMSG(0)
```

Quando você definir uma assinatura administrativa, ele toma quaisquer atributos que você não especificar explicitamente a partir da assinatura padrão, que é chamado SYSTEM.DEFAULT.SUB. Para ver quais esses atributos padrão são, utilize o seguinte comando:

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

Referências relacionadas

[DISPLAY SUB](#)

Mudando atributos de assinatura local

É possível alterar os atributos de assinatura de duas maneiras, usando o comando **ALTER SUB** ou o comando **DEFINE SUB** com o atributo **REPLACE**

Antes de começar

Nota: Os exemplos nesta tarefa requerem que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#) .

Exemplo

Se desejar mudar a prioridade de mensagens entregues para uma assinatura chamada ORANGE para 5, use um dos comandos a seguir:

- Usando o comando **ALTER**:

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

Este comando muda um único atributo, aquele da prioridade de mensagens entregues para esta assinatura para 5; todos os outros atributos permanecem os mesmos.

- Usando o comando **DEFINE**:

```
DEFINE SUB(ORANGE) PUBPRTY(5) REPLACE
```

Esse comando muda não só a prioridade de mensagens entregues para esta assinatura, mas todos os outros atributos que recebem valores padrão.

Se você alterar a prioridade de mensagens enviadas para esta assinatura, as mensagens existentes não são afetadas. Todas as mensagens novas, no entanto, têm a prioridade especificada.

Referências relacionadas

[ALTER SUB](#)

[DEFINE SUB](#)

Copiando uma definição de assinatura local

É possível copiar uma definição de assinatura usando o atributo **LIKE** no comando **DEFINE**.

Antes de começar

Nota: Os exemplos nesta tarefa requerem que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#) .

Exemplo

```
DEFINE SUB(BLUE) +  
LIKE(ORANGE)
```

Também é possível copiar os atributos do sub REAL para o sub THIRD.SUB e especificar que o **correlID** de publicações entregues é THIRD, em vez de os publicadores **correlID**. Por exemplo:

```
DEFINE SUB(THIRD.SUB) +  
LIKE(BLUE) +  
DESTCURL(ORANGE)
```

Referências relacionadas

[DEFINE SUB](#)

Excluindo uma assinatura local

É possível utilizar o comando MQSC **DELETE SUB** para excluir uma assinatura local.

Antes de começar

Nota: Os exemplos nesta tarefa requerem que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Exemplo

```
DELETE SUB(ORANGE)
```

Também é possível excluir uma assinatura utilizando o SUBID:

```
DELETE SUB SUBID(414D5120414141202020202020202020EE921E4E20002A03)
```

Referências relacionadas

[DELETE SUB](#)

Verificando mensagens em uma assinatura

Quando uma assinatura é definida, ela é associada a uma fila. As mensagens correspondentes a esta assinatura serão colocadas nesta fila. Use os comandos MQSC para verificar mensagens atualmente enfileiradas para uma assinatura.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Sobre esta tarefa

Observe que os comandos do MQSC a seguir mostram apenas as assinaturas que receberam mensagens. Para verificar se há mensagens atualmente enfileiradas para uma assinatura execute as seguintes etapas:

Procedimento

1. Para verificar as mensagens enfileiradas para um tipo de assinatura **DISPLAY SBSTATUS**(*sub_name*) **NUMMSGs**, consulte [“Exibindo atributos de assinaturas”](#) na página 183.
2. Se o valor for maior que zero **NUMMSGs** identificar a fila associada à assinatura digitando **DISPLAY SUB**(*sub_name*) **DEST**.

3. Utilizando o nome de fila retornados, é possível visualizar as mensagens, a técnica descrita no [“Procurando filas com o programa de amostra”](#) na página 146.

Referências relacionadas

[DISPLAY SBSTATUS](#)

Trabalhando com Serviços

Os objetos de serviço são um meio pelo qual processos adicionais podem ser gerenciados como parte de um gerenciador de filas. Com os serviços, é possível definir programas que são iniciados e interrompidos quando o gerenciador de filas inicia e termina. Os serviços do IBM MQ são sempre iniciados sob o ID do usuário que iniciou o gerenciador de filas.

Sobre esta tarefa

Os objetos de serviços podem ser um dos tipos a seguir:

Servidor

Um servidor é um objeto de serviço que possui o parâmetro **SERVTYPE** especificado como **SERVER**. Um objeto de serviço do servidor é a definição de um programa que é executado quando um gerenciador de filas especificado é iniciado. Os objetos de serviço do servidor definem programas que geralmente são executados por um longo tempo. Por exemplo, um objeto de serviço do servidor pode ser utilizado para executar um processo monitor do acionador, como **runmqtm**.

Somente uma instância de um objeto de serviço do servidor pode ser executada simultaneamente. O status de objetos de serviço do servidor em execução pode ser monitorado usando o comando do MQSC, **DISPLAY SVSTATUS**.

Comando:

Um comando é um objeto de serviço que possui o parâmetro **SERVTYPE** especificado como **COMMAND**. Os objetos de serviço de comando são semelhantes aos objetos de serviço do servidor; no entanto, diversas instâncias de um objeto de serviço de comando podem ser executadas simultaneamente e seus status não podem ser monitorados usando o comando do MQSC **DISPLAY SVSTATUS**.

Se o comando MQSC, **STOP SERVICE**, for executado, não será feita nenhuma verificação para determinar se o programa iniciado pelo comando MQSC, **START SERVICE**, ainda está ativo antes de parar o programa

Referências relacionadas

[DEFINE SERVICE](#)

[DISPLAY SVSTATUS](#)

[START SERVICE](#)

[STOP SERVICE](#)

Definindo um objeto de serviço

Você define um objeto de serviço com o comando do MQSC **DEFINE SERVICE**.

Antes de começar

Nota: Esta tarefa requer que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#) .

Procedimento

- Defina um objeto de serviço com o comando do MQSC **DEFINE SERVICE**

Os atributos que você precisa definir são os seguintes:

SERVTYPE

Define o tipo do objeto de serviço. Os valores possíveis são os seguintes:

SERVER

Um objeto de serviço do servidor.

Somente uma instância de um objeto de serviço do servidor pode ser executada por vez. O status de objetos de serviço do servidor pode ser monitorado usando o comando MQSC

DISPLAY SVSTATUS.

COMANDO

Um objeto de serviço de comando.

Várias instâncias de um objeto de serviço de comando podem ser executadas simultaneamente. O status de um objeto de serviço de comando não pode ser monitorado.

STARTCMD

O programa que é executado para iniciar o serviço. Um caminho completo para o programa deve ser especificado.

STARTARG

Argumentos transmitidos para o programa de início.

STDERR

Especifica o caminho para um arquivo para o qual o erro padrão (stderr) do programa de serviço deve ser redirecionado.

STDOUT

Especifica o caminho para um arquivo para o qual a saída padrão (stdout) do programa de serviço deve ser redirecionado.

STOPCMD

O programa que é executado para parar o serviço. Um caminho completo para o programa deve ser especificado.

STOPARG

Argumentos passados para o programa de parada.

CONTROL

Especifica como o serviço deve ser iniciado e parado:

MANUAL

O serviço não deve ser iniciado automaticamente ou parado automaticamente. Ele é controlado pelo uso dos comandos **START SERVICE** e **STOP SERVICE**. Esse é o valor-padrão.

QMGR

O serviço que está sendo definido deve ser iniciado e parado ao mesmo tempo que o gerenciador de filas é iniciado e parado.

STARTONLY

O serviço deve ser iniciado ao mesmo tempo que o gerenciador de filas é iniciado, mas não é solicitado a parar quando o gerenciador de filas é parado.

Tarefas relacionadas

[“Serviço gerenciado” na página 187](#)

Uma instância de um objeto de serviço pode ser iniciada e parada automaticamente pelo gerenciador de filas ou iniciada e parada usando os comandos MQSC **START SERVICE** e **STOP SERVICE**

Referências relacionadas

[DEFINE SERVICE](#)

[DISPLAY SVSTATUS](#)

[START SERVICE](#)

[STOP SERVICE](#)

Serviço gerenciado

Uma instância de um objeto de serviço pode ser iniciada e parada automaticamente pelo gerenciador de filas ou iniciada e parada usando os comandos MQSC **START SERVICE** e **STOP SERVICE**

Antes de começar

Nota: Esta tarefa requer que você execute os comandos MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Procedimento

- Configure o parâmetro **CONTROL** no gerenciador de filas para iniciar ou parar uma instância de um objeto de serviço automaticamente ou use os comandos **START SERVICE** e **STOP SERVICE** do MQSC para fazer isso manualmente

Quando uma instância de um objeto de serviço é iniciado, uma mensagem será gravada no log de erros do gerenciador de filas que contém o nome do objeto de serviço e o ID de processo do processo iniciado. Aqui está um exemplo de entrada de log para um objeto de serviço do servidor iniciando:

```
02/15/2005 11:54:24 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5028: The Server 'S1' has started. ProcessId(13031).

EXPLANATION:
The Server process has started.
ACTION:
None.
```

Aqui está um exemplo de entrada de log para um objeto de serviço de comando iniciando:

```
02/15/2005 11:53:55 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5030: The Command 'C1' has started. ProcessId(13030).

EXPLANATION:
The Command has started.
ACTION:
None.
```

Quando um serviço de instância do servidor para, uma mensagem será gravada nos logs de erros do gerenciador de filas que contém o nome do serviço e o ID do processo final. Aqui está um exemplo de entrada de log para uma parada de objeto de serviço do servidor:

```
02/15/2005 11:54:54 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5029: The Server 'S1' has ended. ProcessId(13031).

EXPLANATION:
The Server process has ended.
ACTION:
None.
```

Tarefas relacionadas

[“Definindo variáveis de ambiente adicionais no arquivo service.env” na página 189](#)

Quando um serviço é iniciado, o ambiente no qual o processo de serviço é iniciado é herdado do ambiente do gerenciador de filas. É possível definir variáveis de ambiente adicionais a serem configuradas no ambiente do processo de serviço incluindo as variáveis que você deseja definir em um dos arquivos de substituição de ambiente `service.env`.

Referências relacionadas

[STOP SERVICE \(parar um serviço\) em Multiplataformas](#)

[START SERVICE \(iniciar um serviço\) em Multiplataformas](#)

Definindo variáveis de ambiente adicionais no arquivo service.env

Quando um serviço é iniciado, o ambiente no qual o processo de serviço é iniciado é herdado do ambiente do gerenciador de filas. É possível definir variáveis de ambiente adicionais a serem configuradas no ambiente do processo de serviço incluindo as variáveis que você deseja definir em um dos arquivos de substituição de ambiente `service.env`.

Sobre esta tarefa

Há dois arquivos possíveis nos quais você pode incluir variáveis de ambiente:

- O arquivo `service.env` de escopo da máquina
- O arquivo de escopo do gerenciador de filas `service.env`

Ambos os arquivos são processados, se disponível, com definições no arquivo do escopo do gerenciador de filas que têm precedência sobre as definições no arquivo do escopo de máquina.

É possível especificar qualquer variável de ambiente no arquivo `service.env`. Por exemplo, se o serviço IBM MQ executar vários comandos, poderá ser útil configurar a variável de usuário **PATH** no arquivo `service.env`.

Nota: Os valores para os quais você configura a variável não podem ser variáveis de ambiente; por exemplo, `CLASSPATH= %CLASSPATH%` está incorreto. Da mesma forma, no Linux `PATH= $PATH :/opt/mqm/bin` forneceria resultados inesperados



CLASSPATH deve ser capitalizado, e a instrução de caminho de classe pode conter apenas literais. Alguns serviços (de telemetria, por exemplo) configuram seu próprio caminho da classe. O **CLASSPATH** definido em `service.env` é incluído nele..

O formato das variáveis definidas no arquivo `service.env` é uma lista de pares de variáveis de nome e de valor. Cada variável deve ser definida em uma nova linha e cada variável é obtida como é explicitamente definida, incluindo espaço em branco.

Procedimento

- Inclua variáveis de ambiente no arquivo `service.env` do escopo da Máquina

Esse arquivo está localizado em:


–   `/var/mqm` em sistemas AIX and Linux.

–  O diretório de dados selecionado durante a instalação em sistemas Windows.

- Inclua variáveis de ambiente no arquivo `service.env` do escopo do gerenciador de filas

Esse arquivo está localizado no diretório de dados do gerenciador de filas. Por exemplo, a localização do arquivo de substituição do ambiente para um gerenciador de filas denominado QMNAME é:

–   Em sistemas AIX and Linux, `/var/mqm/qmgrs/QMNAME/service.env`

–  Em sistemas Windows, `C:\ProgramData \IBM \MQ\qmgrs\QMNAME\service.env`

Exemplo de um arquivo service.env

```
#####  
##*                                                                    *##  
##* <N_OCO_COPYRIGHT>                                                *##  
##* Licensed Materials - Property of IBM                               *##  
##*                                                                    *##  
##* 63H9336                                                            *##  
##* (C) Copyright IBM Corporation 2005, 2024.                        *##  
##*                                                                    *##  
##* <NOC_COPYRIGHT>                                                  *##  
##*                                                                    *##  
#####
```

```

#*****#
#* Module Name: service.env                               *#
#* Type       : IBM MQ service environment file         *#
#* Function   : Define additional environment variables to be set *#
#*           : for SERVICE programs.                  *#
#* Usage     : <VARIABLE>=<VALUE>                     *#
#*           :                                         *#
#*****#
MYLOC=/opt/myloc/bin
MYTMP=/tmp
TRACEDIR=/tmp/trace
MYINITQ=ACCOUNTS.INITIATION.QUEUE

```

Tarefas relacionadas

“Usando inserções substituíveis em definições de serviço” na página 190

É possível substituir tokens na definição de um objeto de serviço.. Tokens que são substituídos são automaticamente substituídos por seu texto expandido quando o programa de serviço é executado.

Referências relacionadas

[Descrições de variáveis de ambiente](#)

Usando inserções substituíveis em definições de serviço

É possível substituir tokens na definição de um objeto de serviço.. Tokens que são substituídos são automaticamente substituídos por seu texto expandido quando o programa de serviço é executado.

Sobre esta tarefa

Os tokens substitutos podem ser obtidos da lista de tokens comuns a seguir ou de quaisquer variáveis definidas no arquivo, `service.env`.

Procedimento

- Para usar inserções substituíveis, insira o token dentro de caracteres + em qualquer uma das sequências **STARTCMD**, **STARTARG**, **STOPCMD**, **STOPARG**, **STDOUT** ou **STDERR**.

Para obter exemplos disso, consulte [“Usando um objeto de serviço do servidor”](#) na página 191 e [“Usando um objeto de serviço de comando”](#) na página 193.



Os seguintes são tokens comuns que podem ser utilizados para substituir tokens na definição de um objeto de serviço:

MQ_INSTALL_PATH

O local onde o IBM MQ está instalado.

MQ_DATA_PATH

O local do diretório de dados do IBM MQ:

-  Em sistemas AIX and Linux, o local do diretório de dados do IBM MQ é `/var/mqm/`
-  Em sistemas Windows, o local do diretório de dados do IBM MQ é o diretório de dados selecionado durante a instalação do IBM MQ

QMNAME

O nome do gerenciador de filas atual.

MQ_SERVICE_NAME

O nome do serviço.

MQ_SERVER_PID

Esse token só pode ser usado pelos argumentos **STOPARG** e **STOPCMD**.

Para objetos de serviço do servidor, esse token é substituído pelo ID do processo iniciado pelos argumentos **STARTCMD** e **STARTARG**. Caso contrário, esse token é substituído por 0.

MQ_Q_MGR_DATA_PATH

O local do diretório de dados do gerenciador de filas.

MQ_Q_MGR_DATA_NAME

O nome transformado do gerenciador de filas. Para obter mais informações sobre transformação de nome, consulte [Entendendo IBM MQ nomes de arquivos](#).

Usando um objeto de serviço do servidor

Estes exemplos mostram como definir, usar e alterar um objeto de serviço do servidor para iniciar um monitor acionador ou outro programa..

Antes de começar

Nota: Esses exemplos requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Esses exemplos são gravados com caracteres separadores de caminho de estilo UNIX , exceto quando indicado de outra forma..

Procedimento

1. Defina um objeto de serviço do servidor usando o comando **DEFINE SERVICE MQSC:**

```
DEFINE SERVICE(S1) +  
CONTROL(QMGR) +  
SERVTYPE(SERVER) +  
STARTCMD('+MQ_INSTALL_PATH+bin/runmqtrm') +  
STARTARG('-m +QMNAME+ -q ACCOUNTS.INITIATION.QUEUE') +  
STOPCMD('+MQ_INSTALL_PATH+bin/amqsstop') +  
STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+')
```

em que:

+MQ_INSTALL_PATH+ é um token que representa o diretório de instalação.

+QMNAME+ é um token que representa o nome do gerenciador de filas.

ACCOUNTS.INITIATION.QUEUE é a fila de inicialização.

amqsstop é um programa de amostra fornecido com o IBM MQ que solicita ao gerenciador de filas para dividir todas as conexões para o ID do processo. amqsstop gera comandos PCF, portanto, o servidor de comandos deve estar em execução.

+MQ_SERVER_PID+ é um token que representa o ID do processo passado para o programa de parada.

Consulte [“Usando inserções substituíveis em definições de serviço”](#) na página 190 para uma lista dos tokens comuns.

2. Uma instância de objeto de serviço do servidor é executada quando o gerenciador de filas é iniciado pela próxima vez.. No entanto, é possível iniciar uma instância do objeto de serviço do servidor imediatamente com o comando **START SERVICE MQSC:**

```
START SERVICE(S1)
```

3. Exibir o status do processo de serviço do servidor, usando o comando **DISPLAY SVSTATUS MQSC:**

```
DISPLAY SVSTATUS(S1)
```

4. Mude o objeto de serviço do servidor e tenha as atualizações selecionadas reiniciando manualmente o processo de serviço do servidor, usando o comando **ALTER SERVICE MQSC.**

O objeto de serviço do servidor é alterado para que a fila de inicialização é especificado como JUPITER.INITIATION.QUEUE.

```
ALTER SERVICE(S1) +  
STARTARG('-m +QMNAME+ -q JUPITER.INITIATION.QUEUE')
```

Nota: Um serviço em execução não seleciona nenhuma atualização para sua definição de serviço até que ele seja reiniciado

5. **Reinicie o processo de serviço do servidor para que a alteração seja selecionada usando os comandos STOP SERVICE e START SERVICE MQSC:**

```
STOP SERVICE(S1)
```

Seguido por:

```
START SERVICE(S1)
```

O processo de serviço do servidor é reiniciado e selecionará as alterações feitas no “4” na página 191.

Nota: O comando do MQSC, **STOP SERVICE**, só pode ser usado se um argumento **STOPCMD** for especificado na definição de serviço.

Mais exemplos de argumentos de transmissão

- **Defina um objeto de serviço do servidor para iniciar um programa chamado runserv quando um gerenciador de filas for iniciado..**

Faça isso usando o comando MQSC **DEFINE SERVICE** .

Esse exemplo é escrito com caracteres de separador de caminho de estilo do Windows.

Um dos argumentos que é passado para o programa inicial é uma cadeia que contém um espaço. Esse argumento precisa ser transmitido como uma sequência única. Para isso, aspas duplas são usadas conforme mostrado no comando a seguir para definir o objeto de serviço de comando

```
DEFINE SERVICE(S1) SERVTYPE(SERVER) CONTROL(QMGR) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')

DEFINE SERVICE(S4) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

em que:

+QMNAME+ é um token que representa o nome do gerenciador de filas.

"C:\Program Files\Tools\'" é uma sequência que contém um espaço, que será transmitido como uma sequência única.

- **Defina um objeto de serviço do servidor que possa ser usado para iniciar automaticamente o Trigger Monitor quando o gerenciador de filas for iniciado**

Faça isso usando o comando MQSC **DEFINE SERVICE** .

```
DEFINE SERVICE(TRIG_MON_START) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('runmqtrm') +
STARTARG('-m +QMNAME+ -q +IQNAME+')
```

em que:

+QMNAME+ é um token que representa o nome do gerenciador de filas.

+IQNAME+ é uma variável de ambiente definida pelo usuário em um dos arquivos service.env que representa o nome da fila de inicialização.

Referências relacionadas

[ALTER SERVICE](#)

[DEFINE SERVICE](#)

[DISPLAY SVSTATUS](#)

[START SERVICE](#)

[STOP SERVICE](#)

Usando um objeto de serviço de comando

Estes exemplos mostram como definir um objeto de serviço de comando para iniciar um programa que grava entradas no log do sistema operacional quando um gerenciador de filas é iniciado ou interrompido.

Antes de começar

Nota: Estes exemplos requerem a execução do comando MQSC **DEFINE SERVICE**. A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Estes exemplos são gravados com caracteres separadores de caminho de estilo UNIX.

Sobre esta tarefa

Nos seguintes exemplos:

`logger` é um programa de amostra fornecido com o IBM MQ que pode gravar entradas no log do sistema do sistema operacional.

`+QMNAME+` é um token que representa o nome do gerenciador de filas.

Procedimento

- Defina um objeto de serviço de comando para iniciar um programa que grava entradas no log do sistema do sistema operacional quando um gerenciador de filas é iniciado ou interrompido:

```
DEFINE SERVICE(S2) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STARTCMD('/usr/bin/logger') +
STARTARG('Queue manager +QMNAME+ starting') +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

- Defina um objeto de serviço de comandos para iniciar um programa que grava entradas no log do sistema do sistema operacional somente quando um gerenciador de filas é interrompido:

```
DEFINE SERVICE(S3) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

Referências relacionadas

[DEFINE SERVICE](#)

Gerenciando os Objetos para Acionamento

Estes exemplos mostram como iniciar um aplicativo automaticamente, quando determinadas condições em uma fila são atendidas. Por exemplo, talvez você queira iniciar um aplicativo quando o número de mensagens em uma fila atingir um número especificado. Este recurso é denominado *acionamento*. Você precisa definir os objetos que suportam o acionamento.

Antes de começar

Nota: Esses exemplos requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver [Administrando IBM MQ usando comandos MQSC](#).

Estes exemplos são gravados com caracteres separadores de caminho de estilo UNIX.

Sobre esta tarefa

Para obter uma descrição detalhada do acionador, consulte [Iniciando IBM MQ aplicativos usando acionadores](#).

Procedimento

- Defina uma fila do aplicativo para acionamento

Uma fila do aplicativo é uma fila local que é usada pelos aplicativos para sistema de mensagens, por meio do MQI. O acionamento requer que inúmeros atributos de fila sejam definidos na fila do aplicativo.

O próprio acionamento é ativado pelo atributo **Trigger** (TRIGGER em comandos MQSC). Neste exemplo, um evento acionador deve ser gerado quando houver 100 mensagens de prioridade 5 ou mais na fila local MOTOR.INSURANCE.QUEUE, da seguinte maneira:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +  
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +  
MAXMSGL (2000) +  
DEFPSIST (YES) +  
INITQ (MOTOR.INS.INIT.QUEUE) +  
TRIGGER +  
TRIGTYPE (DEPTH) +  
TRIGDPTH (100)+  
TRIGMPRI (5)
```

em que:

QLOCAL (MOTOR.INSURANCE.QUEUE)

É o nome da fila do aplicativo sendo definida.

PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

É o nome da definição de processo que define o aplicativo a ser iniciado por um programa do monitor acionador.

MAXMSGL (2000)

É o comprimento máximo das mensagens na fila.

DEFPSIST (YES)

Especifica que as mensagens nesta fila são persistentes por padrão.

INITQ (MOTOR.INS.INIT.QUEUE)

É o nome da fila de inicialização na qual o gerenciador de filas deve colocar a mensagem do acionador.

TRIGGER

É o valor de atributo do acionado.

TRIGTYPE (DEPTH)

Especifica que um evento acionador é gerado quando o número de mensagens da propriedade necessária (TRIGMPRI) atinge o número especificado em TRIGDPTH.

TRIGDPTH (100)

É o número de mensagens necessárias para gerar um evento acionador.

TRIGMPRI (5)

É a prioridade de mensagens que devem ser contadas pelo gerenciador de filas ao decidir se deve gerar um evento acionador. Apenas as mensagens com prioridade 5 ou mais são contadas.

- Definir uma fila de inicialização

Quando ocorrer um evento do acionador, o gerenciador de filas colocará uma mensagem do acionador na fila de inicialização especificada na definição da fila do aplicativo. As filas de inicialização não possuem configurações especiais, mas é possível usar a seguinte definição na fila local MOTOR.INS.INIT.QUEUE para orientação:

```
DEFINE QLOCAL (MOTOR.INS.INIT.QUEUE) +
GET (ENABLED) +
NOSHARE +
NOTRIGGER +
MAXMSGL (2000) +
MAXDEPTH (1000)
```

- Definir um processo

Use o comando DEFINE PROCESS para criar uma definição de processo. Uma definição de processo define o aplicativo a ser usado para processar as mensagens da fila do aplicativo. A definição da fila do aplicativo nomeia o processo a ser usado e portanto associa a fila de aplicativos ao aplicativo a ser usado para processar suas mensagens. Isso é feito por meio do atributo PROCESS na fila do aplicativo MOTOR.INSURANCE.QUEUE. O seguinte comando MQSC define o processo necessário, MOTOR.INSURANCE.QUOTE.PROCESS, identificado neste exemplo:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
DESCR ('Insurance request message processing') +
APPLTYPE (UNIX) +
APPLICID ('/u/admin/test/IRMP01') +
USERDATA ('open, close, 235')
```

em que:

MOTOR.INSURANCE.QUOTE.PROCESS

É o nome da definição de processo.

DESCR ('Insurance request message processing')

Descreve o programa de aplicativo ao qual esta definição se relaciona. Este texto é exibido ao usar o comando DISPLAY PROCESS. Isso pode ajudá-lo a identificar o qual o processo faz. Se você usar os espaços na sequência, deve colocar a sequência entre aspas simples.

APPLTYPE (UNIX)

É o tipo de aplicativo a ser iniciado.

APPLICID ('/u/admin/test/IRMP01')

É o nome do arquivo executável do aplicativo, especificado como um nome completo do arquivo. Em sistemas Windows, um típico valor APPLICID seria c:\appl\test\irmp01.exe.

USERDATA ('open, close, 235')

São dados definidos pelo usuário, que podem ser usados pelo aplicativo.

- Exibir atributos de uma definição de processo

Use o comando DISPLAY PROCESS para examinar os resultados de sua definição. Por exemplo:

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing')
APPLICID ('/u/admin/test/IRMP01')
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (UNIX)
```

Também é possível usar o comando ALTER PROCESS do MQSC para alterar uma definição de processo existente e o comando DELETE PROCESS para excluir uma definição de processo.

Usando o utilitário **dmpmqmsg** entre dois sistemas

O utilitário **dmpmqmsg** (anteriormente *qload*) permite copiar ou mover o conteúdo de uma fila, ou suas mensagens, para um arquivo.

Visão Geral

O arquivo que você cria com **dmpmqmsg** pode ser salvo fora conforme necessário e usado em algum ponto posterior para recarregar as mensagens novamente na fila.

Importante:

1. O arquivo tem um formato específico compreendido pelo utilitário. No entanto, o arquivo é legível para que você possa atualizá-lo em um editor antes de recarregá-lo. Se você editar o arquivo, não deverá mudar seu formato.
2. O utilitário **dmpmqmsg** é enviado com o conjunto de arquivos de tempo de execução para AIX, Linux, and Windows, portanto, está disponível no servidor e cliente IBM MQ.

Os usos possíveis são:

- Salvar as mensagens que estão em uma fila para um arquivo. Possivelmente para fins de arquivamento e recarregar de volta para uma fila mais tarde.
- Recarregar uma fila com mensagens que você salvou anteriormente em um arquivo.
- Remover antigas mensagens de uma fila.
- 'Reproduzir' as mensagens de teste a partir de um local armazenado, mesmo mantendo o horário correto entre as mensagens, se necessário.



Atenção: SupportPac MO03 usou o parâmetro **-l** para especificar uma ligação local ou de cliente. **-l** foi substituído pelo parâmetro **-c**.

-P agora é usado para informações de página de códigos em vez de **-c**.

Consulte [dmpmqmsg](#) para obter informações adicionais sobre o comando e os parâmetros disponíveis.

Exemplo de uso do utilitário **dmpmqmsg** no Linux usando uma máquina Windows

Você possui um gerenciador de filas em uma máquina Linux que tem mensagens em uma fila (*Q1*) que você deseja mover para outra fila (*Q2*) no mesmo gerenciador de filas. Você deseja iniciar o utilitário **dmpmqmsg** a partir de uma máquina Windows.

A fila (*Q1*) tem quatro mensagens que foram incluídas usando o aplicativo de amostra **amqspu**t (gerenciador de filas locais) ou **amqspu**tc (gerenciador de filas remotas).

Na máquina Linux, você vê:

```
display q1(Q1) CURDEPTH
      2 : display q1(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

Configure a variável de ambiente **MQSERVER** para apontar para o gerenciador de filas no Linux. Por exemplo:

```
set MQSERVER=SYSTEM.DEF.SVRCONN/TCP/veracruz.x.com(1414)
```

em que *veracruz* é o nome da máquina.

Execute o utilitário **dmpmqmsg** para ler a partir da fila, *Q1*, e armazene a saída em `c:\temp\mqqload.txt`.

Conecte-se como um cliente remoto ao gerenciador de filas, *QM_VER*, em execução no host e na porta do Linux estabelecidos por MQSERVER. Você obtém a conexão como um cliente remoto usando o atributo: -c.

```
dmpmqmsg -m QM_VER -i Q1 -f c:\temp\mqqlload.txt -c
Read      - Files:    0  Messages:    4  Bytes:          22
Written - Files:    1  Messages:    4  Bytes:          22
```

O arquivo de saída `c:\temp\mqqlload.txt` contém texto usando um formato que o utilitário **dmpmqmsg** entende.

Na máquina Windows, emita o comando **dmpmqmsg** (usando a opção -o em vez da opção -i) para carregar fila (Q2) na máquina Linux a partir de um arquivo na máquina Windows:

```
dmpmqmsg -m QM_VER -o Q2 -f c:\temp\mqqlload.txt -c
Read      - Files:    1  Messages:    4  Bytes:          22
Written - Files:    0  Messages:    4  Bytes:          22
```

Na máquina Linux, observe que agora há quatro mensagens na fila que foram restauradas a partir do arquivo.

```
display ql(Q2) CURDEPTH
      6 : display ql(Q2) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q2)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

Na máquina Linux,

Exclua as mensagens da fila original.

```
clear qllocal(Q1)
      4 : clear qllocal(Q1)
AMQ8022: IBM MQ queue cleared.
```

Confirme se não há mais mensagens na fila original:

```
display ql(Q1) CURDEPTH
      5 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(0)
```

Consulte [dmpmqmsg](#) para ver uma descrição do comando e de seus parâmetros.

Conceitos relacionados

[“Exemplos de como usar o utilitário dmpmqmsg” na página 197](#)

Maneiras simples nas quais é possível usar o utilitário **dmpmqmsg** (anteriormente **qload**)

Exemplos de como usar o utilitário dmpmqmsg

Maneiras simples nas quais é possível usar o utilitário **dmpmqmsg** (anteriormente **qload**)

Descarregue uma fila para um arquivo

Use as opções a seguir na linha de comandos para salvar as mensagens que estão em uma fila, em um arquivo:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile
```

Este comando utiliza uma cópia das mensagens da fila e as salva no arquivo especificado.

Descarregue uma fila para uma série de arquivos

É possível descarregar uma fila para uma série de arquivos utilizando um caractere `insert` no nome do arquivo. Nesse modo cada mensagem é gravada em um novo arquivo:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile%n
```

Este comando descarrega a fila para arquivos, `myfile1`, `myfile2`, `myfile3` e assim por diante.

Carregar uma fila a partir de um arquivo

Para recarregar uma fila com as mensagens que você salvou em [“Descarregue uma fila para um arquivo”](#) na página 197, use as opções a seguir na linha de comandos:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Este comando descarrega a fila para arquivos, `myfile1`, `myfile2`, `myfile3` e assim por diante.

Carregar uma fila a partir de uma série de arquivos

É possível carregar uma fila a partir de uma série de arquivos utilizando um caractere `insert` no nome do arquivo. Nesse modo cada mensagem é gravada em um novo arquivo:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Este comando carrega a fila para arquivos, `myfile1`, `myfile2`, `myfile3` e assim por diante.

Copie as mensagens de uma fila para outra fila

Substitua o parâmetro do arquivo em [“Descarregue uma fila para um arquivo”](#) na página 197 por outro nome da fila e utilize as opções a seguir:

```
dmpmqmsg -m QM1 -i Q1 -o Q2
```

Esse comando permite que as mensagens de uma fila sejam copiadas para outra fila.

Copie as primeiras 100 mensagens de uma fila para outra fila

Use o comando no exemplo anterior e inclua a opção `-r#100`:

```
dmpmqmsg -m QM1 -i Q1 -o Q2 -r#100
```

Mova as mensagens de uma fila para outra

Uma variação do [“Carregar uma fila a partir de um arquivo”](#) na página 198. Observe a distinção entre usar `-i` (minúscula) que apenas procura em uma fila e `-I` (maiúscula) que destrutivamente obtém de uma fila:

```
dmpmqmsg -m QM1 -I Q1 -o Q2
```

Mova mensagens mais antigas que um dia de uma fila para outra fila

Este exemplo mostra o uso de seleção de idade. As mensagens que podem ser selecionadas são as mais antigas que, mais jovens que ou as que estão dentro de um intervalo de idades.

```
dmpmqmsg -m QM1 -I Q1 -o Q2 -T1440
```

Exiba as idades de mensagens atualmente em uma fila

Utilize as seguintes opções na linha de comandos:

```
dmpmqmsg -m QM1 -i Q1 -f stdout -dT
```

Trabalhe com o arquivo de mensagens

Após ter descarregado a mensagem a partir de sua fila, como em [“Descarregue uma fila para um arquivo”](#) na página 197, é possível desejar editar o arquivo.

Também é possível querer mudar o formato do arquivo para utilizar uma das opções de exibição que você não especificou no momento em que descarregou a fila.

É possível usar o utilitário **dmpmqmsg** para processar novamente o arquivo no formato necessário mesmo depois que o descarregamento da fila ocorreu. Utilize as opções a seguir na linha de comandos.

```
dmpmqmsg -f c:\oldfile -f c:\newfile -dA
```

Consulte [dmpmqmsg](#) para ver uma descrição do comando e de seus parâmetros.

Trabalhando com objetos remotos do IBM MQ

É possível administrar objetos do IBM MQ em gerenciadores de filas remotas usando comandos MQSC, comandos PCF ou a administrativa REST API. Antes de poder usar qualquer um desses métodos, deve-se definir filas de transmissão e canais entre o gerenciador de filas locais e o gerenciador de filas remotas para que os comandos possam ser enviados para o gerenciador de filas remotas e respostas recebidas pelo gerenciador de filas locais. Como alternativa, é possível configurar um cluster de gerenciador de filas e, em seguida, usar os mesmos métodos de administração remota.

Sobre esta tarefa

Para preparar gerenciadores de filas para administração remota, deve-se configurar os objetos a seguir no gerenciador de filas locais:

- Um ouvinte.
- Uma fila de transmissão que tem o nome do gerenciador de filas remotas.
- Um canal emissor que tem os detalhes de conexão para o gerenciador de filas remotas.
- Um canal receptor que tem o mesmo nome que o canal emissor no gerenciador de filas remotas.

Deve-se também configurar os objetos a seguir no gerenciador de filas remotas:

- Um ouvinte.
- Uma fila de transmissão que tem o nome do gerenciador de filas locais.
- Um canal emissor que tem os detalhes de conexão para o gerenciador de filas locais.
- Um canal receptor que tem o mesmo nome que o canal emissor no gerenciador de filas locais.

Para obter mais informações sobre como configurar esses objetos, veja [“Configurando Gerenciadores de Filas para Administração Remota”](#) na página 200.

Como alternativa, é possível configurar um cluster de gerenciador de filas. Um *cluster* é um grupo de gerenciadores de filas configurados de tal modo que os gerenciadores de filas podem se comunicar diretamente entre si através de uma rede única sem fila de transmissão complexa, canal e definições de fila. Os clusters podem ser configurados facilmente e geralmente contêm gerenciadores de filas que

estão logicamente relacionados de alguma maneira e precisam compartilhar dados ou aplicativos. Mesmo o menor cluster reduz os custos de administração do sistema.

Estabelecer uma rede de gerenciadores de filas em um cluster envolve menos definições do que estabelecer um ambiente de enfileiramento distribuído tradicional. Com menos definições a fazer, é possível configurar ou alterar sua rede mais rápida e facilmente e reduzir o risco de cometer um erro em suas definições.

Para configurar um cluster, você precisa de um emissor de cluster (CLUSDR) e a definição de um receptor de cluster (CLUSRCVR) para cada gerenciador de filas. Você não precisa de nenhuma definição de fila de transmissão ou definições de fila remota. Os princípios da administração remota são os mesmos quando utilizados dentro de um cluster, mas as próprias definições são bem mais simplificadas.

Para obter mais informações sobre como configurar um cluster, veja [Configurando um cluster de gerenciador de filas](#).

Procedimento

- Para obter informações sobre como administrar objetos remotos do IBM MQ, veja os subtópicos a seguir:
 - [“Configurando Gerenciadores de Filas para Administração Remota”](#) na página 200
 - [“Gerenciando o servidor de comandos para administração remota”](#) na página 204
 - [“Emitindo comandos MQSC em um gerenciador de filas remotas”](#) na página 205
 - [“Conversão de dados entre conjuntos de caracteres codificados”](#) na página 207

Configurando Gerenciadores de Filas para Administração Remota

É possível administrar um gerenciador de filas remotas por meio de um gerenciador de filas locais usando os comandos de administrative REST API, MQSC ou PCF. O gerenciador de filas remotas pode estar no mesmo sistema, em uma instalação diferente ou em um sistema diferente com o mesmo ambiente ou em um ambiente diferente do IBM MQ. Antes de poder administrar remotamente um gerenciador de filas por meio de um gerenciador de filas locais, deve-se criar um canal emissor e receptor, um listener e uma fila de transmissão em cada gerenciador de filas. Esses canais e filas permitem que os comandos sejam enviados para o gerenciador de filas remotas e as respostas sejam recebidas no gerenciador de filas locais. O procedimento para criar essas filas e canais é o mesmo se você deseja usar os comandos de administrative REST API, MQSC ou PCF.

Antes de começar

- O procedimento a seguir usa os gerenciadores de filas de exemplo `source.queue.manager` e `target.queue.manager`. Deve-se criar e iniciar esses gerenciadores de filas em seu sistema para seguir estas etapas ou substituir seus próprios nomes de gerenciadores de filas nas etapas relevantes.
- O procedimento a seguir usa TCP/IP como o tipo de transporte. Deve-se saber o endereço IP de ambos os sistemas para concluir esta tarefa.
- O procedimento a seguir cria listeners que usam as portas de rede 1818 no sistema local e 1819 no sistema remoto. É possível usar outras portas, mas deve-se substituir seus valores de porta nas etapas apropriadas.
- Deve-se executar os comandos no procedimento localmente ou em um recurso de rede como o Telnet.

Sobre esta tarefa

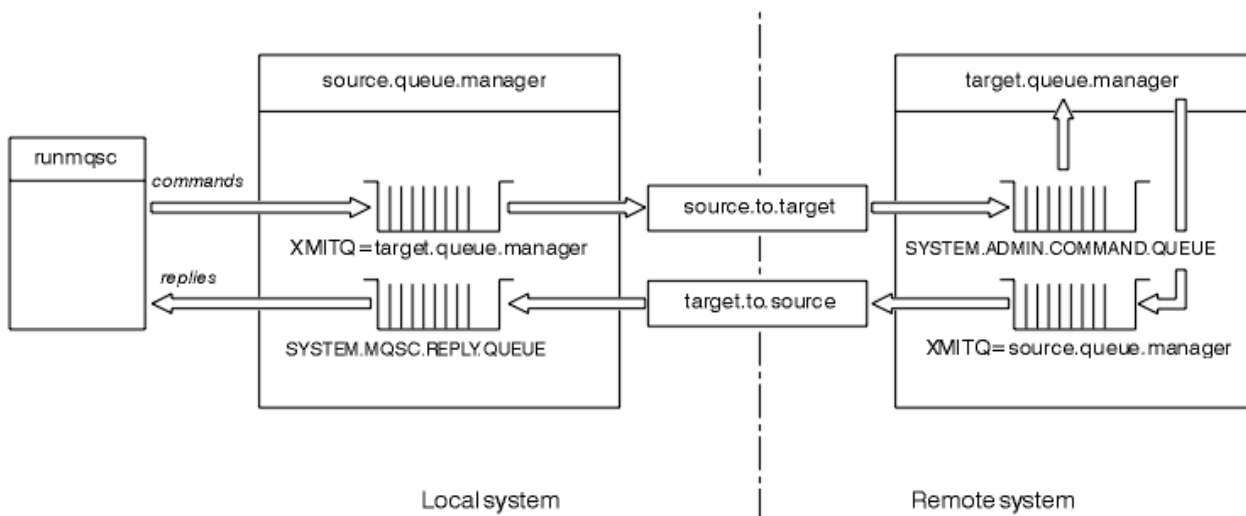


Figura 15. Configurando canais e filas para administração remota

A Figura 15 na página 201 mostra a configuração de gerenciadores de filas, filas e canais necessários para a administração remota:

- O objeto `source.queue.manager` é o gerenciador de filas de origem do qual é possível emitir comandos de administrative REST API, MQSC ou PCF e para o qual os resultados desses comandos são retornados.
- O objeto `target.queue.manager` é o nome do gerenciador de filas de destino, que processa os comandos e gera as mensagens do operador.
- Os comandos são colocados na fila de transmissão que tem o mesmo nome que o gerenciador de filas remotas. Nesse caso, `target.queue.manager`. Uma fila de transmissão é uma fila local especializada que mantém temporariamente as mensagens antes que o MCA as selecione e as envie para o gerenciador de filas remotas.
- Os comandos são enviados pelo canal `source.to.target` para o `SYSTEM.ADMIN.COMMAND.QUEUE` no gerenciador de filas remotas. Cada extremidade do canal possui uma definição separada. Uma extremidade é um emissor e a outra extremidade é um receptor. As duas definições devem ter o mesmo nome e juntas constituir um canal de mensagens único.
- A saída de comando é colocada na fila de transmissão remota que tem o mesmo nome que o gerenciador de filas locais do qual o comando foi enviado. Nesse caso, `source.queue.manager`.
- A saída é enviada pelo canal `target.to.source` para uma fila de resposta apropriada, na qual ela é tomada e enviada pelo comando original.

Procedimento

1. No gerenciador de filas do sistema remoto, assegure-se de que a fila de comandos `SYSTEM.ADMIN.COMMAND.QUEUE` esteja presente. Esta fila é criada por padrão quando um gerenciador de filas é criado.
2. No sistema remoto, verifique se o servidor de comandos está em execução no gerenciador de filas. Se o servidor de comandos não estiver em execução, a administração remota não será possível:
 - a) Inicie `runmqsc` para o gerenciador de filas. Por exemplo, para o gerenciador de filas `target.queue.manager`, insira o comando a seguir:

```
runmqsc target.queue.manager
```

- b) Exiba o status do servidor de comandos, inserindo o comando a seguir:

```
DISPLAY QMSTATUS CMDSERV
```

- c) Saia do prompt de comandos do **runmqsc** inserindo o comando a seguir:

```
end
```

- d) Se o servidor de comandos não tiver sido iniciado, inicie-o. Por exemplo, para o gerenciador de filas `target.queue.manager`, insira o comando a seguir:

```
strmqcsv target.queue.manager
```

3. Defina os canais, o listener e a fila de transmissão no gerenciador de filas locais:

- a) Inicie **runmqsc** para o gerenciador de filas. Por exemplo, para o gerenciador de filas `source.queue.manager`, insira o comando a seguir:

```
runmqsc source.queue.manager
```

- b) Defina o canal emissor. Esse canal emissor deve ter o mesmo nome que o canal receptor no gerenciador de filas remotas. Por exemplo, insira o comando MQSC a seguir, substituindo o valor para **CONNNAME** pelo endereço IP para o gerenciador de filas remotas e o número da porta do listener:

```
DEFINE CHANNEL ('source.to.target') +  
CHLTYPE(SDR) +  
CONNNAME (localhost:1819) +  
XMITQ ('target.queue.manager') +  
TRPTYPE(TCP)
```

- c) Defina o canal receptor. Esse canal receptor deve ter o mesmo nome que o canal emissor no gerenciador de filas remotas. Por exemplo, insira o comando a seguir:

```
DEFINE CHANNEL ('target.to.source') +  
CHLTYPE(RCVR) +  
TRPTYPE(TCP)
```

- d) Defina o listener no gerenciador de filas locais. Por exemplo, insira o comando a seguir:

```
DEFINE LISTENER ('source.queue.manager') +  
TRPTYPE (TCP) +  
PORT (1818)
```

- e) Defina a fila de transmissão no gerenciador de filas locais. Essa fila de transmissão deve ter o mesmo nome que o gerenciador de filas remotas. Por exemplo, insira o comando a seguir:

```
DEFINE QLOCAL ('target.queue.manager') +  
USAGE (XMITQ)
```

- f) Inicie o listener. Por exemplo, insira o comando a seguir:

```
START LISTENER ('source.queue.manager')
```

- g) Saia do prompt de comandos do **runmqsc** inserindo o comando a seguir:

```
end
```

4. Defina os canais, o listener e a fila de transmissão no gerenciador de filas remotas:

- a) Inicie **runmqsc** para o gerenciador de filas. Por exemplo, para o gerenciador de filas `target.queue.manager`, insira o comando a seguir:

```
runmqsc target.queue.manager
```

- b) Defina o canal emissor. Esse canal emissor deve ter o mesmo nome que o canal receptor no gerenciador de filas locais. Por exemplo, insira o comando MQSC a seguir, substituindo o valor para **CONNNAME** pelo endereço IP para o gerenciador de filas locais e o número da porta para o listener:

```
DEFINE CHANNEL ('target.to.source') +
CHLTYPE(SDR) +
CONNNAME (localhost:1818) +
XMITQ ('source.queue.manager') +
TRPTYPE(TCP)
```

- c) Defina o canal receptor. Esse canal receptor deve ter o mesmo nome que o canal emissor no gerenciador de filas locais. Por exemplo, insira o comando a seguir:

```
DEFINE CHANNEL ('source.to.target') +
CHLTYPE(RCVR) +
TRPTYPE(TCP)
```

- d) Defina o listener. Por exemplo, insira o comando a seguir:

```
DEFINE LISTENER ('target.queue.manager') +
TRPTYPE (TCP) +
PORT (1819)
```

- e) Defina a fila de transmissão. Essa fila de transmissão deve ter o mesmo nome que o gerenciador de filas locais. Por exemplo, insira o comando a seguir:

```
DEFINE QLOCAL ('source.queue.manager') +
USAGE (XMITQ)
```

- f) Inicie o listener. Por exemplo, insira o comando a seguir:

```
START LISTENER ('target.queue.manager')
```

- g) Saia do **runmqsc** inserindo o comando a seguir:

```
end
```

5. Inicie o canal emissor no sistema local:

- a) Inicie **runmqsc** para o gerenciador de filas. Por exemplo, para o gerenciador de filas `source.queue.manager`, insira o comando a seguir:

```
runmqsc source.queue.manager
```

- b) Inicie o canal emissor. Por exemplo, insira o comando a seguir:

```
START CHANNEL ('source.to.target')
```

- c) Saia do **runmqsc** inserindo o comando a seguir:

```
end
```

6. Inicie o canal emissor no sistema remoto:

- a) Inicie **runmqsc** para o gerenciador de filas. Por exemplo, para o gerenciador de filas `target.queue.manager`, insira o comando a seguir:

```
runmqsc target.queue.manager
```

- b) Inicie o canal emissor. Por exemplo, insira o comando a seguir:

```
START CHANNEL ('target.to.source')
```

c) Saia do **runmqsc** inserindo o comando a seguir:

```
end
```

7. Teste se a configuração foi concluída com êxito enviando um comando MQSC do sistema local para o gerenciador de filas remotas:

a) Inicie o prompt de comandos **runmqsc** para o gerenciador de filas remotas a partir do sistema local. Por exemplo, insira o comando a seguir:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

b) Exiba as filas no gerenciador de filas remotas inserindo o comando a seguir:

```
DISPLAY QUEUE (*)
```

No sucesso, uma lista de filas do gerenciador de filas remotas é exibida.

c) Se essas etapas não funcionarem, verifique se os canais em ambos os sistemas estão em um estado de execução. Se os canais não estiverem em execução e não forem iniciados, verifique se os canais e as filas de transmissão estão configurados corretamente e se o servidor de comandos está em execução. Por exemplo, verifique se o CONNAME correto está especificado para os canais emissores e se as filas de transmissão têm os nomes corretos. Além disso, verifique os logs do gerenciador de filas para obter as exceções de segurança que podem ajudar a resolver o problema.

Resultados

Seus gerenciadores de filas estão configurados para administrar remotamente o gerenciador de filas remotas por meio do sistema local.

Como proceder a seguir

- Saiba mais sobre administração remota usando comandos MQSC: [“Emitindo comandos MQSC em um gerenciador de filas remotas”](#) na página 205
- Saiba mais sobre gravação de programas de administração usando comandos PCF: [“Usando os formatos de comando programável do IBM MQ”](#) na página 27.
- Saiba mais sobre o uso da administrative REST API para administração remota: [“Administração remota usando a REST API”](#) na página 80.

Gerenciando o servidor de comandos para administração remota

Cada gerenciador de filas tem um servidor de comandos associado a ele. Um servidor de comandos processa qualquer comando de entrada a partir de gerenciadores de filas remotas ou comandos PCF a partir de aplicativos. Ele apresenta os comandos para o gerenciador de filas para processamento e retorna um código de conclusão ou uma mensagem do operador. É possível iniciar, parar e exibir o status do servidor de comandos. Um servidor de comandos é obrigatório para toda a administração que envolve os comandos PCF, a MQAI e também para administração remota.

Antes de começar

Dependendo do valor do atributo do gerenciador de filas, **SCMDSERV**, o servidor de comandos é iniciado automaticamente quando o gerenciador de filas é iniciado ou deve ser iniciado manualmente. Se o servidor de comandos for iniciado automaticamente, não será possível usar os comandos **strmqcsv** ou **endmqcsv** para iniciar e parar o servidor de comandos. É possível mudar o valor do atributo **SCMDSERV** usando o comando MQSC **ALTER QMGR**. Por padrão, o servidor de comandos é iniciado automaticamente.

Parar um gerenciador de filas também encerra o servidor de comandos associado a ele.

Procedimento

- Exiba o status do servidor de comandos:

- a) Inicie o prompt de comando **runmqsc** para o gerenciador de filas apropriado inserindo o comando a seguir:

```
runmqsc target.queue.manager
```

em que `target.queue.manager` é o gerenciador de filas para o qual o servidor de comandos está sendo exibido.

- b) Exiba o status do servidor de comandos inserindo o comando MQSC a seguir:

```
DISPLAY QMSTATUS CMDSERV
```

- c) Saia do prompt de comandos do **runmqsc** inserindo o comando a seguir:

```
end
```

- Se o servidor de comandos não estiver configurado para ser iniciado automaticamente, inicie o servidor de comandos inserindo o comando a seguir:

```
strmqcsv target.queue.manager
```

em que `target.queue.manager` é o gerenciador de filas para o qual o servidor de comandos está sendo iniciado.

- Se o servidor de comandos não estiver configurado para ser iniciado automaticamente, pare o servidor de comandos inserindo o comando a seguir:

```
endmqcsv target.queue.manager
```

em que `target.queue.manager` é o gerenciador de filas para o qual o servidor de comandos está sendo interrompido.

Por padrão, o servidor de comandos é interrompido de uma maneira controlada. É possível parar o servidor de comandos imediatamente incluindo a sinalização `-i` no comando.

Emitindo comandos MQSC em um gerenciador de filas remotas

Depois de configurar os gerenciadores de filas para administração remota, é possível usar uma forma específica do comando **runmqsc** em um sistema local para executar comandos MQSC em um gerenciador de filas remotas. Cada comando é enviado como um PCF de escape para a fila de comandos, `SYSTEM.ADMIN.COMMAND.QUEUE`, do gerenciador de filas remotas. As respostas são recebidas na fila `SYSTEM.MQSC.REPLY.QUEUE`.

Antes de começar

Deve-se concluir as etapas em [“Configurando Gerenciadores de Filas para Administração Remota”](#) na página 200 para configurar canais, filas de transmissão, listeners e o servidor de comandos antes de poder administrar remotamente um gerenciador de filas usando comandos MQSC.

Procedimento

1. Assegure-se de que o servidor de comandos esteja em execução no gerenciador de filas remotas. Para obter informações sobre como iniciar o servidor de comandos em um gerenciador de filas, consulte [“Gerenciando o servidor de comandos para administração remota”](#) na página 204.
2. No gerenciador de filas de origem, é possível, então, executar comandos MQSC de uma de duas maneiras:
 - Interativamente, iniciando **runmqsc** com os comandos a seguir:

- Se o gerenciador de filas remotas estiver no z/OS, insira o comando a seguir:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager
```

- Se o gerenciador de filas remotas estiver em Multiplataformas, insira o seguinte comando:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- A partir de um arquivo de comando:
 - a. Coloque os comandos MQSC a serem executados no sistema remoto em um arquivo de texto, um comando por linha.
 - b. Verifique os comandos do MQSC no gerenciador de filas locais usando o sinalizador `-v` no comando **runmqsc**. A sinalização `-v` verifica se os comandos são válidos, mas não os executa. Esteja ciente de que alguns comandos poderão falhar se eles forem aplicáveis ao gerenciador de filas remotas, mas não forem aplicáveis ao gerenciador de filas locais:

```
runmqsc -v source.queue.manager < myCmdFile.in > results.out
```

O `myCmdFile.in` contém os comandos MQSC para verificar, e o arquivo `results.out` contém os resultados de verificação para os comandos.

- c. Execute o arquivo de comando no gerenciador de filas remotas inserindo um dos comandos a seguir:

- Se o gerenciador de filas remotas estiver no z/OS, insira o comando a seguir:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager < myCmdFile.in >  
results.out
```

- Se o gerenciador de filas remotas estiver em Multiplataformas, insira o seguinte comando:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager < myCmdFile.in >  
results.out
```

Os parâmetros usados são os parâmetros a seguir:

-w seconds

Especifica que os comandos MQSC são executados no modo indireto, em que os comandos são colocados na fila de entrada do servidor de comandos e executados em ordem.

A variável `seconds` especifica a duração de tempo de espera, em segundos, para uma resposta do gerenciador de filas remotas. Quaisquer respostas recebidas após esse tempo são descartadas, mas os comandos MQSC ainda são executados no gerenciador de filas remotas. A mensagem a seguir é gerada no gerenciador de filas locais quando o comando atinge o tempo limite:

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

Ao parar de emitir comandos do MQSC, o gerenciador de filas local exibe quaisquer respostas de tempo limite que chegaram e descarta quaisquer respostas adicionais.

-x

Especifica que o gerenciador de filas remotas é um gerenciador de filas do z/OS.

-m localQMgrName

Especifica o nome do gerenciador de filas locais que você deseja usar para enviar comandos para o gerenciador de filas remotas

Como proceder a seguir

Se você tiver dificuldade em executar comandos MQSC remotamente:

- Verifique se o gerenciador de filas remotas está em execução.
- Verifique se o servidor de comandos está em execução no sistema remoto.
- Verifique se o intervalo de desconexão do canal não expirou. Por exemplo, se um canal foi iniciado, mas foi encerrado após algum tempo. Isso é especialmente importante se você iniciar os canais manualmente.
- Assegure-se de que as solicitações que são enviadas do gerenciador de filas locais façam sentido para o gerenciador de filas de destino. Por exemplo, solicitações que incluem parâmetros que não são suportados no gerenciador de filas remotas.
- Veja também [Resolvendo problemas com comandos MQSC](#).

Conversão de dados entre conjuntos de caracteres codificados

Os dados da mensagem em formatos definidos pelo IBM MQ (também conhecidos como formatos integrados) podem ser convertidos pelo gerenciador de filas de um conjunto de caracteres codificados para outro, desde que ambos os conjuntos de caracteres se relacionem a um único idioma ou grupo de idiomas semelhantes.

Por exemplo, a conversão entre conjuntos de caracteres codificados com identificadores (CCSIDs) 850 e 500 é suportada porque ambos se aplicam aos idiomas europeus ocidentais.

Para conversões de caracteres EBCDIC newline (NL) para ASCII, consulte [Todas as sub-rotinas de gerenciadores de filas do mqsc.ini](#) arquivo e a variável de ambiente **AMQ_CONVEBCDICNEWLINE** .

As conversões suportadas são definidas em [Processamento de conversão de dados](#).

A conversão entre os CCSIDs 37 e 500 é suportada nos IBM MQ Appliance, Windows, Linux e macOS

Quando um gerenciador de filas não puder converter mensagens em formatos integrados

O gerenciador de filas não pode converter mensagens em formatos integrados automaticamente se seus CCSIDs representam diferentes grupos de idioma nacional. Por exemplo, a conversão entre o CCSID 850 e o CCSID 1025 (que é um conjunto de caracteres EBCDIC para idiomas codificados utilizando script cirílico) não é suportada porque muitos dos caracteres em um conjunto de caracteres codificado não podem ser representados no outro. Se você tiver uma rede de gerenciadores de filas funcionando em diferentes idiomas nacionais e a conversão de dados entre alguns dos conjuntos de caracteres codificados não for suportada, é possível ativar uma conversão padrão.

Para as plataformas às quais o `ccsid_part2.tbl` se aplica, consulte [“Especificando a conversão de dados padrão”](#) na página 210 usando `ccsid_part2.tbl` para obter informações adicionais. A conversão de dados padrão em plataformas diferentes daquelas para as quais o arquivo `ccsid_part2.tbl` se aplica é descrita em [“Conversão de dados padrão”](#) na página 208

Suporte de Conversão de Dados Unicode Aprimorado

O produto suporta todos os caracteres Unicode definidos no padrão Unicode 8.0 em conversão de dados. Isso inclui suporte completo para UTF-16, incluindo pares substitutos (um par de caracteres UTF-16 de 2 bytes no intervalo de X' D800 ' a X' DFFF ' que representam um ponto de código Unicode acima de U+FFFF).

As sequências de caracteres de combinação também são suportadas nos casos em que um caractere pré-editado em um CCSID é mapeado para uma sequência de caracteres de combinação em outro CCSID.

A conversão de Dados para e a partir de Unicode e CCSIDs 1388, 1390, 1399, 4933, 5488 e 16884 foi estendida, em algumas plataformas, para suportar todos os pontos de código atualmente definidos para esses CCSIDs, incluindo aqueles que são mapeados para pontos de código em planos complementares do Unicode.

No caso dos CCSIDs 1390, 1399 e 16884, isso inclui caracteres definidos no padrão JIS X 0213 (JIS2004).

Também foi incluído suporte para conversão de e para Unicode e seis novos CCSIDs (de 1374 a 1379).

Arquivo `ccsid_part2.tbl`


Um arquivo adicional, `ccsid_part2.tbl`, é fornecido


O arquivo `ccsid_part2.tbl` tem precedência sobre o arquivo `ccsid.tbl` e:

- Permite incluir ou modificar entradas de CCSID
- Especificar conversão de dados padrão
- Especificar dados para diferentes níveis de comando

O `ccsid_part2.tbl` é aplicável somente às plataformas a seguir:

-  Linux - todas as versões
-  Windows

 No IBM MQ for Windows, `ccsid_part2.tbl` está localizado no diretório `MQDataRoot\conv\table` por padrão. Além disso, no IBM MQ for Windows, ele registra todos os conjuntos de códigos suportados.


 No IBM MQ for Linux, o `ccsid_part2.tbl` está localizado no diretório `MQDataRoot/conv/table` e os conjuntos de códigos suportados são mantidos em tabelas de conversão fornecidas pelo IBM MQ.

Embora o arquivo `ccsid_part2.tbl` substitua o arquivo `ccsid.tbl` existente usado em versões anteriores do IBM MQ para fornecer informações CCSID adicionais, o arquivo `ccsid.tbl` continua sendo analisado pelo IBM MQ e, portanto, não deve ser excluído.

Para obter informações adicionais, consulte [“O arquivo `ccsid_part2.tbl`” na página 209](#).

Arquivo `ccsid.tbl`

Em plataformas diferentes daquelas às quais o `ccsid_part2.tbl` se aplica, o arquivo `ccsid.tbl` é usado para os propósitos a seguir:

-  No AIX, os conjuntos de códigos suportados são retidos internamente pelo sistema operacional.
- Ele especifica quaisquer conjuntos de códigos adicionais. Para especificar conjuntos de códigos adicionais, é necessário editar `ccsid.tbl` (orientação sobre como fazer isso é fornecido no arquivo).
- Ele especifica qualquer conversão de dados padrão.

É possível atualizar as informações registradas em `ccsid.tbl`; você pode desejar fazer isso se, por exemplo, uma liberação futura de seu sistema operacional suportar conjuntos de caracteres codificados adicionais.

Conversão de dados padrão

Se você configurar canais entre duas máquinas nas quais a conversão de dados normalmente não é suportada, deve-se ativar a conversão de dados padrão para os canais funcionarem.

Em plataformas diferentes daquelas às quais o `ccsid_part2.tbl` se aplica, para ativar a conversão de dados padrão, edite o arquivo `ccsid.tbl` para especificar um EBCDIC CCSID padrão e um ASCII CCSID padrão. Instruções sobre como fazer isto estão incluídas no arquivo. Deve-se fazer isso em todas as máquinas que serão conectadas utilizando os canais. Reinicie o gerenciador de filas para que a alteração entre em vigor.

O padrão de conversão de dados do processo é o seguinte:

- Se a conversão entre os CCSIDs de origem e destino não é suportada, mas os CCSIDs dos ambientes de origem e destino são ambos EBCDIC ou ASCII, os dados de caracteres são transmitidos para o aplicativo de destino sem conversão.
- Se um CCSID representa um conjunto de caracteres codificados ASCII e o outro representa um conjunto de caracteres codificados EBCDIC, o IBM MQ converte os dados utilizando os CCSIDs de conversão de dados padrão definidos em `ccsid.tbl`.

Nota: Tente restringir os caracteres que estão sendo convertidos àqueles que têm os mesmos valores de código no conjunto de caracteres codificados especificado para a mensagem e no conjunto de caracteres codificados padrão. Se você usar apenas o conjunto de caracteres que é válido para nomes de objetos do IBM MQ (conforme definido em [Nomeando objetos IBM MQ](#)), você atenderá, em geral, a esse requisito. Exceções ocorrem com os CCSIDs do EBCDIC 290, 930, 1279 e 5026 utilizados no Japão, onde os caracteres minúsculos possuem códigos diferentes daqueles utilizados em outros CCSIDs do EBCDIC.

Convertendo mensagens em formatos definidos pelo usuário

O gerenciador de filas não pode converter mensagens em formatos definidos pelo usuário a partir de um conjunto de caracteres codificado para outro. Se você precisar converter os dados em um formato definido pelo usuário, deve-se fornecer uma saída de conversão de dados para cada formato desse. Não utilize CCSIDs padrão para converter dados de caractere em formatos definidos pelo usuário. Para obter mais informações sobre a conversão de dados em formatos definidos pelo usuário e sobre a gravação de saídas de conversão de dados, consulte [Gravando saídas de conversão de dados](#).

Mudando o CCSID do gerenciador de filas

Quando tiver usado o atributo **CCSID** do comando **ALTER QMGR** para alterar o CCSID do gerenciador de fila, pare e reinicie o gerenciador de filas para assegurar que todos os aplicativos em execução, incluindo o servidor de comandos e os programas de canal, sejam parados e reiniciados.

Isso é necessário porque todos os aplicativos que estão em execução quando o gerenciador de filas CCSID é mudado continuam a utilizar o CCSID existente.

O arquivo `ccsid_part2.tbl`

O arquivo `ccsid_part2.tbl` é usado para fornecer informações adicionais de CCSID. O arquivo `ccsid_part2.tbl` substitui o arquivo `ccsid.tbl` que era usado antes da IBM MQ 9.0.

Nota: O arquivo `ccsid.tbl`, que foi usado antes de IBM MQ 9.0 para fornecer informações adicionais do CCSID, continua a ser analisado por IBM MQ e não deve ser excluído. No entanto, as entradas em `ccsid_part2.tbl` têm precedência sobre outras entradas em `ccsid.tbl`.

É necessário usar `ccsid_part2.tbl` em vez de `ccsid.tbl` porque `ccsid_part2.tbl`:

- Contém suporte para os valores de codificação Unicode. No IBM MQ 9.0, o produto suporta todos os caracteres Unicode definidos no padrão Unicode 8.0 na conversão de dados, incluindo o suporte completo para UTF-16. Para obter mais informações, consulte [“Conversão de dados entre conjuntos de caracteres codificados”](#) na página 207.
- Permite especificar a versão de entradas CCSID, para que as entradas sejam aplicáveis somente aos níveis de comando selecionados.



É possível usar o arquivo `ccsid_part2.tbl` para:

- Incluir ou modificar entradas CCSID
- Especificar conversão de dados padrão
- Especificar dados para diferentes níveis de comando

O arquivo `ccsid_part2.tbl` é aplicável apenas às seguintes plataformas:

-  Linux - todas as versões
-  Windows

O local do arquivo `ccsid_part2.tbl` depende de sua plataforma:

-  O diretório `MQDataRoot/conv/table` em todas as versões de Linux.
-  O diretório `MQDataRoot\conv\table` no Windows.

Incluindo ou modificando entradas CCSID

Uma entrada no arquivo `ccsid_part2.tbl` possui o formato a seguir:

```
<CCSID number> <Base CCSID> <DBCS CodePage> <SBCS CodePage>  
<Type> <Encoding> <ACRI> <Name>
```

Uma entrada de exemplo para o CCSID 1200 (UTF-16) é:

```
1200 1200 1200 1200 3 8 0 UTF-16
```

Nota: Para obter mais detalhes sobre o valor para ACRI, consulte o comentário no arquivo `ccsid_part2.tbl`.

No formato `ccsid_part2.tbl`:

Tipo pode ser igual:

1=SBCS

2=DBCS

3=MBCS

Codificação pode ser igual:

1=EBCDIC

2 = ASCII

3 = ISO

4 = UCS-2

5 = UTF-8

6 = Euc

7 = GB18030

8 = UTF-16

9 = UTF-32

Ao editar o arquivo, você:

- Pode especificar um comentário usando o símbolo `#` no início de uma linha. Isso impede o IBM MQ de tentar analisar a linha.
- Não pode fornecer comentários sequenciais.
- Deve assegurar que não crie linhas em branco.
- Não deve incluir novas entradas no fim do arquivo.

Novas entradas CCSID devem ser incluídas antes das informações de tabela ACRI.

Especificando a conversão de dados padrão

Será possível definir CCSIDs de conversão padrão, que são usados para converter entre ASCII ou semelhante e EBCDIC CCSIDs, se nenhuma conversão for suportada entre dois CCSIDs.

Se você ativar essa função, a conversão padrão será usada para transmissão e cabeçalhos da mensagem e também poderá ser usada na conversão de dados do usuário.

As conversões padrão são ativadas criando duas linhas semelhantes ao seguinte:

```
default      0      500    1      1      0
default      0      850    1      2      0
```

A primeira linha configura o padrão para EBCDIC CCSIDs como 500 e a segunda linha configura o padrão para ASCII e CCSIDs semelhantes como 850.

Especificando dados para diferentes níveis de comando

Para especificar entradas CCSID para diferentes níveis de comando do IBM MQ, use um símbolo de dois pontos seguido pelo nível de comando (ou níveis de comando) do IBM MQ ao qual deseja que a próxima seção seja aplicável.

O número representa o nível de comando mínimo em que o gerenciador de filas ou cliente deve estar em execução. Por exemplo, se o gerenciador de filas atual for o nível de comando 900 e encontrar uma sinalização de nível de comando 800 ou 900, os CCSIDs serão lidos.

Entretanto, um gerenciador de filas no nível 800 ignora quaisquer CCSIDs na seção 900.

O nível de comando especificado é aplicável a todas as entradas CCSID encontradas após uma sinalização de nível de comando, até que uma sinalização de nível de comando seja localizada.

Se for necessário configurar o nível de comando para todos os níveis de comando, especifique o número zero.

Ao analisar primeiro `ccsid_part2.tbl`, o IBM MQ tratará todos os CCSIDs encontrados como válidos para todos os níveis de comando do IBM MQ.

A versão começa a ser usada somente quando o IBM MQ encontra a primeira sinalização de nível de comando.

O fragmento de código a seguir mostra um exemplo do uso de Versão:

```
# Comment Block
# End of Comment Block
# Because no command level flag is specified and we're at the start of the file
# the following CCSIDs will be read on all versions
 819 819 0 819 1 3 0 IS08859-1
 923 923 0 923 1 3 0 IS08859-15
1051 1051 0 1051 1 3 0 IBM-1051
# The colon :900 below shows that the CCSIDs after will only be for MQ cmd level 900 and above
:900
 8629 437 0 437 1 2 0 IBM-437
12725 437 0 437 1 2 0 IBM-437
16821 437 0 437 1 2 0 IBM-437
20917 437 0 437 1 2 0 IBM-437
# The colon :0 below shows that the CCSIDs after will be for all version of MQ
:0
 4946 850 0 850 1 2 0 IBM-850
33618 850 0 850 1 2 0 IBM-850
61697 850 0 850 1 2 0 IBM-850
61698 850 0 850 1 2 0 IBM-850
```

Administrando Managed File Transfer

Use comandos do Managed File Transfer para administrar o Managed File Transfer. É possível também usar o IBM MQ Explorer para algumas das tarefas administrativas.

Iniciar a transferência colocando uma mensagem em uma fila de comandos do agente

Também é possível iniciar uma transferência de arquivos colocando uma mensagem de transferência de arquivos na fila de comandos do agente de origem. Um exemplo de nome da fila de comando é `SYSTEM.FTE.COMMAND.AGENT01`. Deve-se assegurar que a mensagem alcance a fila de comando do agente de origem correta; se a mensagem for recebida por um agente que não corresponda às informações de origem no XML, a mensagem será rejeitada.

O XML de solicitação de transferência deve estar em conformidade com o esquema `FileTransfer.xsd` e usar o elemento `<request>` como o elemento raiz. Consulte [Formato da Mensagem de Pedido de Transferência de Arquivos](#) para obter informações sobre a estrutura e o conteúdo de uma mensagem de pedido de transferência. O modo como a mensagem de solicitação de transferência é colocada em uma fila de comandos do agente é específico da tarefa. Por exemplo, é possível usar a API IBM MQ Java para colocar uma mensagem na fila programaticamente.

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens” na página 267](#)

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

[“Transferindo Dados de Mensagens para Arquivos” na página 283](#)

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

[“A ponte de protocolo” na página 294](#)

A ponte de protocolo permite que a rede do Managed File Transfer (MFT) acesse arquivos armazenados em um servidor de arquivos fora da rede do MFT, em seu domínio local ou um local remoto. Este servidor de arquivos pode usar os protocolos de rede FTP, FTPS ou SFTP. Cada servidor de arquivos precisa de pelo menos um agente dedicado. O agente dedicado é conhecido como o agente de ponte de protocolo. Um agente de ponte pode interagir com múltiplos servidores de arquivos.

[“Trabalhando com o MFT no IBM Integration Bus” na página 334](#)

É possível trabalhar com o Managed File Transfer a partir do IBM Integration Bus usando os nós `FTEOutput` e `FTEInput`.

[“Recuperação e reinicialização do MFT” na página 334](#)

Se seu agente ou gerenciador de filas estiver indisponível por alguma razão, por exemplo, devido a uma falha de energia ou de rede, o Managed File Transfer se recuperará da seguinte forma nestes cenários:

Tarefas relacionadas

[“Iniciando um agente MFT” na página 213](#)

Antes de usar um agente do Managed File Transfer para uma transferência de arquivos, primeiro deve-se iniciar o agente.

[“Iniciando uma Nova Transferência de Arquivos” na página 220](#)

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

[“Monitorando Transferências de Arquivos que Estão em Andamento” na página 227](#)

É possível monitorar uma transferência de arquivos que está em andamento usando a guia **Transferência de arquivos gerenciada - Progresso da transferência atual** no IBM MQ Explorer. Esta transferência de arquivos pode ser uma iniciada a partir do IBM MQ Explorer ou da linha de comandos. A guia também exibe o progresso da transferências planejadas no ponto em que a transferência inicia.

[“Visualizando o Status de Transferências de Arquivos no Log de Transferência” na página 229](#)

É possível visualizar os detalhes de transferências de arquivos usando o **Log de transferência** no IBM MQ Explorer. Essas transferências podem ser aquelas iniciadas da linha de comandos ou do IBM MQ Explorer. Também é possível customizar o que é exibido no **Log de Transferência**.

[“Monitorando recursos do MFT” na página 232](#)

É possível monitorar recursos do Managed File Transfer; por exemplo, uma fila ou um diretório. Quando uma condição nesse recurso é satisfeita, o monitor de recurso inicia a tarefa, como uma transferência de arquivos. É possível criar um monitor de recurso usando o comando `fteCreateMonitor` ou a visualização **Monitores** no plug-in do Managed File Transfer para o IBM MQ Explorer.

[“Trabalhando com modelos de transferência de arquivos” na página 264](#)

É possível usar os modelos de transferência de arquivos para armazenar as configurações de transferência de arquivos comuns para transferências repetidas ou complexas. Crie um modelo de transferência da linha de comandos usando o comando `fteCreateTemplate` ou use o IBM MQ Explorer

para criar um modelo de transferência, usando o assistente **Criar novo modelo para transferência de arquivos gerenciada** ou salve um modelo enquanto estiver criando uma transferência de arquivos, marcando a caixa de seleção **Salvar configurações de transferência como um modelo**. A janela **Modelos de Transferência** exibe todos os modelos de transferência criados na rede do Managed File Transfer.

[“Listando Agentes do MFT” na página 218](#)

Você pode listar os agentes Managed File Transfer registrados em um determinado gerenciador de filas usando a linha de comandos ou o IBM MQ Explorer.

[“Parando um agente MFT” na página 219](#)

É possível parar um agente Managed File Transfer na linha de comandos. Ao parar um agente, você suspende o agente e permite que ele conclua sua transferência de arquivos atual antes de parar. Também é possível especificar o parâmetro **-i** na linha de comandos para parar um agente imediatamente. Quando o agente for parado, você não poderá utilizá-lo para transferir arquivos até que ele seja reiniciado.

[Configurando um criador de logs do MFT](#)

Referências relacionadas

[Diretrizes para Transferências de Arquivos](#)

Iniciando um agente MFT

Antes de usar um agente do Managed File Transfer para uma transferência de arquivos, primeiro deve-se iniciar o agente.

Sobre esta tarefa

É possível iniciar um Managed File Transfer Agent na linha de comandos. Nesse caso, o processo do agente parará quando você efetuar logoff do sistema.

ALW No AIX, Linux, and Windows, é possível configurar um agente para que ele continue em execução quando você efetuar logoff do sistema e poder continuar a receber transferências de arquivos.

z/OS No z/OS, é possível configurar o agente para ser iniciado como uma tarefa iniciada do JCL sem a necessidade de uma sessão interativa.

Observe que, se um agente encontra um erro irreversível quando ele está em execução, uma primeira captura de dados com falha (FDC) é gerada e o agente é interrompido.

Procedimento

- Para iniciar um agente na linha de comandos, use o comando **fteStartAgent**.

Para obter mais informações, consulte [fteStartAgent](#).

- **ALW**

Para configurar um agente para que ele continue em execução quando você efetuar logoff do sistema:

- **Windows** No Windows, configure o agente para ser executado como um serviço do Windows. Para obter informações adicionais, consulte [“Iniciando um agente MFT como serviço do Windows” na página 214](#).

- **Linux** **AIX** No AIX and Linux, configure o agente para iniciar automaticamente durante uma reinicialização usando um arquivo de script. Para obter mais informações, consulte [“Iniciando um agente do MFT na inicialização do sistema AIX and Linux” na página 216](#).

- **z/OS**

No z/OS, configure o agente para ser iniciado como uma tarefa iniciada do JCL sem a necessidade de uma sessão interativa.

Para obter informações adicionais, consulte [“Starting an MFT agent on z/OS” na página 218](#).

Tarefas relacionadas

“Listando Agentes do MFT” na página 218

Você pode listar os agentes Managed File Transfer registrados em um determinado gerenciador de filas usando a linha de comandos ou o IBM MQ Explorer.

“Parando um agente MFT” na página 219

É possível parar um agente Managed File Transfer na linha de comandos. Ao parar um agente, você suspende o agente e permite que ele conclua sua transferência de arquivos atual antes de parar. Também é possível especificar o parâmetro **-i** na linha de comandos para parar um agente imediatamente. Quando o agente for parado, você não poderá utilizá-lo para transferir arquivos até que ele seja reiniciado.

Referências relacionadas

Valores de status do agente MFT

fteStartAgent

Windows

Iniciando um agente MFT como serviço do Windows

É possível iniciar um agente como um serviço do Windows para que, quando você efetuar logoff do Windows, seu agente continue em execução e possa receber transferências de arquivos.

Sobre esta tarefa

No Windows, quando você inicia um agente a partir da linha de comandos, o processo do agente é executado usando o nome de usuário que você usou para efetuar logon no Windows. Quando você efetua logoff do sistema, o processo do agente é parado. Para evitar que o agente pare, é possível configurar um agente para ser executado como um serviço do Windows. Executar como um serviço do Windows também permite configurar os agentes para que sejam iniciados automaticamente quando o ambiente do Windows for iniciado ou for reiniciado.

Conclua as etapas a seguir para iniciar um agente que é executado como um serviço do Windows. Deve-se estar executando o Managed File Transfer em uma das versões suportadas do Windows para executar o agente como um serviço do Windows. Para obter a lista de ambientes suportados, consulte [Requisitos do sistema para o IBM MQ](#).

As etapas exatas dependem de você já ter criado um agente ou se você está criando um agente. As duas opções são descritas nas seguintes etapas.

Procedimento

1. Se estiver criando um agente Managed File Transfer, use o comando **fteCreateAgent**, **fteCreateCDAgent** ou **fteCreateBridgeAgent**. Especifique o parâmetro **-s** para executar o agente como um serviço do Windows. No seguinte exemplo, o agente AGENT1 é criado, possuindo um gerenciador de filas do agente QMGR1. O serviço do Windows é executado usando um nome de usuário `fteuser`, que possui uma senha associada `ftepassword`.

```
fteCreateAgent -agentName AGENT1 -agentQMGR QMGR1 -s -su fteuser -sp ftepassword
```

Opcionalmente, é possível especificar um nome para o serviço após o parâmetro **-s**. Se você não especificar um nome, o serviço será nomeado como `mqmftAgentAGENTQMGR`, em que, *AGENT* é o nome do agente especificado e *QMGR* é o nome do gerenciador de filas do agente. Nesse exemplo, o nome padrão para o serviço é `mqmftAgentAGENT1QMGR1`.

Nota: A conta do usuário do Windows que você especifica usando o parâmetro **-su** deve ter os direitos **Log on as a service**. Para obter informações sobre como configurar isso, consulte [Resolução de problemas de um agente MFT ou criador de logs em execução como um Windows serviço](#).

Para obter mais informações, consulte [fteCreateAgent](#), [fteCreateCDAgent](#): criar um Connect:Direct agente de ponte ou [fteCreateBridgeAgent](#) (criar e configurar um MFT agente de ponte de protocolo)

2. Se você seguiu a etapa anterior para criar um agente, execute os comandos MQSC gerados pelo comando **fteCreateAgent**, **fteCreateCDAgent** ou **fteCreateBridgeAgent**. Esses comandos criam as filas do IBM MQ que são necessárias ao agente.

Por exemplo, para um agente chamado *AGENT1*, um gerenciador de filas do agente chamado *QMGR1* e um gerenciador de filas de coordenação chamado *COORDQMGR1*, execute o seguinte comando:

```
runmqsc QMGR1 MQ_DATA_PATH\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

3. Se você não seguiu as etapas anteriores para criar um agente e, em vez disso, deseja configurar um agente existente para ser executado como um serviço do Windows, primeiro pare o seu agente se ele estiver em execução e, em seguida, modifique sua configuração.

- a) O seguinte exemplo usa um agente chamado *AGENT1*. Execute o seguinte comando:

```
fteStopAgent AGENT1
```

- b) Use o comando **fteModifyAgent** para configurar o agente para ser executado como um serviço do Windows:

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

Para obter mais informações, consulte [fteModifyAgent: executar um agente MFT como um serviço Windows](#).

4. Inicie o agente usando o comando **fteStartAgent**. Como alternativa, é possível usar a ferramenta Windows Service, que está disponível em Ferramentas Administrativas no Painel de Controle, selecionada no menu de início da área de trabalho do Windows, para iniciar o serviço.

```
fteStartAgent AGENT1
```

O serviço continua em execução mesmo que você efetue logoff do Windows. Para assegurar que o serviço também seja reiniciado quando o Windows for reiniciado após um encerramento, o campo **Tipo de inicialização** na ferramenta Windows Service é configurado para **Automático** por padrão. Mude isso para **Manual** se você não desejar que o serviço seja reiniciado quando Windows for reiniciado.

5. Opcional: Para parar o agente, use o comando [fteStopAgent](#) ou use a ferramenta de serviços do Windows. Por exemplo, na linha de comandos, execute o seguinte comando:

```
fteStopAgent AGENT1
```

- Ao executar o comando **fteStopAgent** como um serviço, o comando sempre é executado usando o parâmetro **-i**, independentemente de este parâmetro ter sido especificado. O parâmetro **-i** para o agente imediatamente, sem concluir as transferências que estão em andamento. Isso é causado por uma limitação do serviço do Windows.

Como proceder a seguir

Se você tiver problemas ao iniciar o serviço Windows, consulte [Resolução de problemas de um MFT agente ou criador de logs em execução como um Windows serviço](#). Este tópico também descreve o local dos arquivos de log de serviço do Windows.

Tarefas relacionadas

[“Listando Agentes do MFT” na página 218](#)

Você pode listar os agentes Managed File Transfer registrados em um determinado gerenciador de filas usando a linha de comandos ou o IBM MQ Explorer.

[“Parando um agente MFT” na página 219](#)

É possível parar um agente Managed File Transfer na linha de comandos. Ao parar um agente, você suspende o agente e permite que ele conclua sua transferência de arquivos atual antes de parar. Também

é possível especificar o parâmetro **-i** na linha de comandos para parar um agente imediatamente. Quando o agente for parado, você não poderá utilizá-lo para transferir arquivos até que ele seja reiniciado.

Referências relacionadas

[fteCreateAgent](#) (criar um agente do MFT)

[fteCreateCDAgent](#) (criar um agente de ponte Connect:Direct)

[fteCreateBridgeAgent](#) (criar e configurar um agente de ponte de protocolo do MFT)

[fteModifyAgent](#) (executar um agente do MFT como um serviço Windows)

Informações relacionadas

O arquivo MFT [agent.properties](#)

Linux

AIX

Iniciando um agente do MFT na inicialização do sistema

AIX and Linux

Um Managed File Transfer Agent pode ser configurado para iniciar uma inicialização do sistema no AIX and Linux. Ao efetuar logoff, seu agente continua em execução e pode receber transferências de arquivo.

Quando você tiver criado e configurado um agente usando um desses comandos Managed File Transfer ; **fteCreateAgent**, **fteCreateCDAgent** ou **fteCreateBridgeAgent**, será possível configurá-lo para iniciar automaticamente durante uma reinicialização em máquinas AIX and Linux usando um arquivo de script que simplesmente executa o comando a seguir:

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

Em que *mq_install_root* é o diretório-raiz da instalação necessária do Managed File Transfer, o padrão é: */opt/mqm* e *agent_name* é o nome do Managed File Transfer Agent a ser iniciado. O uso deste arquivo de script varia dependendo do sistema operacional específico. Por exemplo, há opções adicionais disponíveis em Linux.

Linux

Linux

Para sistemas Linux, há várias maneiras que você pode iniciar aplicativos durante o processo de inicialização do sistema. No geral, considere seguir estas etapas:

1. Crie um arquivo chamado */etc/rc.mqmft* com conteúdo:

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name"
```

Em que *mqmft_user* é o ID do usuário sob o qual o processo do agente deve ser executado. Esse ID do usuário deve ser um membro do grupo *mqm*.

2. Torne o arquivo executável, por exemplo:

```
chmod 755 /etc/rc.mqmft
```

3. Em seguida, inclua a seguinte linha em */etc/inittab*:

```
mqmft:5:boot:/etc/rc.mqmft
```

Outras formas de iniciar um agente durante a inicialização no Linux incluem adicionar as linhas de script no arquivo */etc/rc.d/rc.local* ou no Linux SuSe, adicionando as linhas de script ao arquivo */etc/init.d/boot.local*. Deve-se selecionar o método que funciona melhor para o seu ambiente. Aqui estão mais algumas informações sobre outras maneiras de iniciar um agente durante a inicialização em distribuições específicas do Linux suportadas:

SLES 10 e 11

Para sistemas SUSE Linux Enterprise Server (SLES) 10 e 11, siga estas etapas:

1. Como o ID do usuário raiz do sistema, crie seu próprio arquivo `/etc/init.d/rc.rclocal`.
2. Inclua as linhas a seguir no arquivo `rc.rclocal`:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: rc.rclocal
# Required-Start: $network $syslog
# Required-Stop: $network $syslog
# Default-Stop: 0 1 2 6
# Description: MQMFT agent startup
### END INIT INFO
su -l mqmft_user"-c mq_install_root/bin/fteStartAgent agent_name"
```

3. Execute os comandos a seguir:

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```

Iniciando agentes do Managed File Transfer no Linux com systemd

Linux

Execute o seguinte procedimento:

1. Crie um arquivo na pasta do sistema `/etc/systemd/` e o nomeie, por exemplo, `<agentname>.service`. Inclua o conteúdo a seguir, em que `<agentname>` é `MFT_AGT_LNX_0`.

```
# vi /etc/systemd/system/MFT_AGT_LNX_0.service
[Unit]
Description=IBM MQ MFT MFT_AGT_LNX_0
[Service]
ExecStart=/opt/mqm/bin/fteStartAgent MFT_AGT_LNX_0
ExecStop=/opt/mqm/bin/fteStopAgent MFT_AGT_LNX_0
Type=forking
User=mqm
Group=mqm
KillMode=none
```

2. Para ativar o serviço, execute os comandos a seguir:

```
# systemctl enable MFT_AGT_LNX_0
# systemctl daemon-reload
```

3. Para iniciar o agente e verificar seu status. execute os comandos a seguir:

```
# systemctl start MFT_AGT_LNX_0
# systemctl status MFT_AGT_LNX_0
```

Tarefas relacionadas

[“Parando um agente MFT” na página 219](#)

É possível parar um agente Managed File Transfer na linha de comandos. Ao parar um agente, você suspende o agente e permite que ele conclua sua transferência de arquivos atual antes de parar. Também é possível especificar o parâmetro `-i` na linha de comandos para parar um agente imediatamente. Quando o agente for parado, você não poderá utilizá-lo para transferir arquivos até que ele seja reiniciado.

Referências relacionadas

[fteCreateAgent](#)

[fteCreateCDAgent: criar um agente de ponte Connect:Direct](#)

[fteCreateBridgeAgent](#) (criar e configurar um agente de ponte de protocolo do MFT)

Starting an MFT agent on z/OS

On z/OS, in addition to running the **fteStartAgent** command from a z/OS UNIX System Services session, you can start an agent as a started task from JCL without the need for an interactive session.

A started task is used because it runs under a specific user ID and is not affected by users logging off.

Note: Started tasks are typically run under an administrative user that might not have log-on privileges and so it is not possible to log on to the z/OS system as the user that the agent is running under. The **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

You can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. You can define a security manager group, for example MFTADMIN and then add the started task userid and administrator TSO ids to this group. Edit the agent properties file and set the **adminGroup** property to be the name of this security manager group.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

For more information, see the **adminGroup** property in [The MFT agent.properties file](#).

As a Java application, an agent is a z/OS UNIX System Services application that you can run from JCL by using the BFGAGSTP member, from a generated Managed File Transfer command PDSE library data set for an agent. For more information about how to create an MFT command PDSE library data set, and customize it for the required agent, see [Creating an MFT Agent or Logger command data set](#).

Related concepts

[Enabling MFT agents to connect to remote z/OS queue managers](#)

Related reference

[“Stopping an MFT agent on z/OS” on page 219](#)

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

[The MFT agent.properties file](#)

Listando Agentes do MFT

Você pode listar os agentes Managed File Transfer registrados em um determinado gerenciador de filas usando a linha de comandos ou o IBM MQ Explorer.

Sobre esta tarefa

Para listar agentes utilizando a linha de comandos, consulte o comando [fteListAgents](#).

Para listar agentes usando o IBM MQ Explorer, na visualização Navegador, clique em **Agentes** com o nome do gerenciador de filas de coordenação.

Se um agente não for listado pelo comando **fteListAgents** ou não for exibido no IBM MQ Explorer, use o fluxograma de diagnóstico no tópico a seguir para localizar e corrigir o problema: [O que fazer se seu agente MFT não for listado pelo comando fteListAgents](#).

Referências relacionadas

[fteListAgents: listar os agentes MFT para um gerenciador de filas de coordenação](#)

[Valores de status do agente MFT](#)

[fteShowAgentDetails](#)

Parando um agente MFT

É possível parar um agente Managed File Transfer na linha de comandos. Ao parar um agente, você suspende o agente e permite que ele conclua sua transferência de arquivos atual antes de parar. Também é possível especificar o parâmetro **-i** na linha de comandos para parar um agente imediatamente. Quando o agente for parado, você não poderá utilizá-lo para transferir arquivos até que ele seja reiniciado.

Antes de começar

Se desejar verificar os nomes dos agentes associados a um gerenciador de filas, será possível listar agentes usando o IBM MQ Explorer ou a linha de comandos; veja o [comando fteListAgents](#).

Sobre esta tarefa

Para parar um agente da linha de comandos, veja [fteStopAgent](#).

Se um agente for interrompido de uma maneira controlada usando o **fteStopAgent**, o agente não aceitará nenhuma nova solicitação de transferência gerenciada e esperará que quaisquer transferências em andamento sejam concluídas antes que ele realmente se encerre. Em IBM MQ 9.3.0, para mostrar que o agente ainda está em um estado temporário e, portanto, ainda não foi encerrado e ainda não pode ser reiniciado, o agente entra no estado STOPPING até que quaisquer transferências em andamento sejam concluídas. This status appears in the output of the [fteListAgents](#) and [fteShowAgentDetails](#) commands, and in [consultas do MFT REST API](#), and in the [MFT plugin's Agentes](#) view of IBM MQ Explorer.

Windows Se você tiver configurado seu agente para ser executado como um serviço do Windows, executar o comando **fteStopAgent** também parará o serviço do Windows. Como alternativa, é possível parar o agente parando o serviço, usando a ferramenta Serviços do Windows. Para obter mais informações, consulte o tópico [“Iniciando um agente MFT como serviço do Windows”](#) na página 214.

Tarefas relacionadas

[“Iniciando um agente MFT”](#) na página 213

Antes de usar um agente do Managed File Transfer para uma transferência de arquivos, primeiro deve-se iniciar o agente.

Referências relacionadas

[Valores de status do agente MFT](#)

[fteStopAgent](#)

[“Stopping an MFT agent on z/OS”](#) na página 219

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

z/OS Stopping an MFT agent on z/OS

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

A started task is used because it runs under a specific user ID and is not affected by users logging off.

Note: Started tasks are typically run under an administrative user that might not have log-on privileges and so it is not possible to log on to the z/OS system as the user that the agent is running under. The **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

You can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. You can define a security manager group, for example MFTADMIN and then add the started task userid and administrator TSO ids to this group. Edit the agent properties file and set the **adminGroup** property to be the name of this security manager group.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

For more information, see the **adminGroup** property in [The MFT agent.properties file](#).

Controlled agent shutdown by using the z/OS MODIFY command (F)

The **MODIFY** command allows you to stop an agent in a controlled way as an alternative to the **fteStopAgent** command. The agent completes any transfers currently in progress but the agent does not start any new transfers.

For example:

```
F job_name,APPL=STOP
```

where *job_name* is the job that the agent process is running under.

Immediate agent shutdown by using the z/OS STOP command (P)

The **STOP** command is equivalent to an immediate stop by using the **fteStopAgent** command with the **-i** parameter. The agent is stopped immediately even if the agent is currently transferring a file.

For example:

```
P job_name
```

where *job_name* is the job that the agent process is running under.

Related concepts

[Enabling MFT agents to connect to remote z/OS queue managers](#)

Related reference

[“Starting an MFT agent on z/OS” on page 218](#)

On z/OS, in addition to running the **fteStartAgent** command from a z/OS UNIX System Services session, you can start an agent as a started task from JCL without the need for an interactive session.

[The MFT agent.properties file](#)

Iniciando uma Nova Transferência de Arquivos

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Sobre esta tarefa

Para iniciar uma nova transferência de arquivos a partir da linha de comandos, consulte [Comando fteCreateTransfer](#).

Para iniciar uma nova transferência de arquivos usando o assistente **Criar nova transferência de arquivos gerenciada** no IBM MQ Explorer, use as etapas a seguir:

Procedimento

1. Na visualização do Navegador, clique em **Transferência de Arquivos Gerenciados**. A **Central de Transferências de Arquivos Gerenciados** é exibida na visualização Conteúdo.
2. Todos os gerenciadores de filas da coordenação são exibidos na visualização Navegador. Expanda o nome do gerenciador de filas da coordenação do registrado para o agente que você deseja utilizar para a transferência. Se você estiver atualmente conectado a um gerenciador de filas da coordenação que não seja o que deseja utilizar para a transferência, clique com o botão direito do mouse nesse nome do gerenciador de filas da coordenação na visualização Navegador e clique em **Desconectar**. Em seguida,

clique com o botão direito do mouse no nome do gerenciador de filas da coordenação que você deseja utilizar e clique em **Conectar**.

3. Inicie o assistente **Criar Nova Transferência de Arquivos Gerenciada** usando qualquer um dos seguintes métodos:
 - a) Clique com o botão direito do mouse no nome de qualquer um dos seguintes nós na visualização Navegador: o respectivo gerenciador de filas de coordenação, **Modelos de Transferência**, **Log de Transferência** ou **Transferências Pendentes**. Em seguida, clique em **Nova Transferência** para iniciar o assistente.
 - b) Clique em **Arquivo > Novo > Outro > Assistentes Transferência de Arquivos Gerenciados > Assistente Nova Transferência**
4. Siga as instruções nos painéis do assistente. Uma ajuda sensível ao contexto também é fornecida em cada painel. Para acessar a ajuda contextual no Windows, pressione F1. No Linux, pressione Ctrl+F1 ou Shift+F1.

Conceitos relacionados

[“Utilizando Arquivos de Definição de Transferência” na página 221](#)

É possível especificar um arquivo de definição de transferência que pode ser usado para criar uma transferência de arquivo. O arquivo de definição de transferência é um arquivo XML que define algumas ou todas as informações necessárias para criar a transferência.

Tarefas relacionadas

[“Criando uma Transferência de Arquivo Planejada” na página 224](#)

É possível planejar uma nova transferência de arquivos por meio do IBM MQ Explorer ou da linha de comandos. A transferência planejada pode conter arquivos individuais ou um grupo com vários arquivos. É possível executar uma transferência de arquivos uma vez ou repeti-la várias vezes.

[“Acionando a Transferência de Arquivo” na página 226](#)

É possível configurar que determinadas condições de acionamento em uma transferência de arquivo sejam verdadeiras para que a transferência ocorra. Se as condições de acionamento não forem verdadeiras, a transferência de arquivos não ocorrerá e uma mensagem de log será enviada opcionalmente para registrar o fato de que a transferência não ocorreu. O pedido de transferência de arquivo será então, descartado. Por exemplo, é possível configurar uma transferência de arquivos que ocorrerá apenas se um arquivo nomeado no sistema em que o agente de origem está localizado exceder um tamanho especificado, ou se um determinado arquivo nomeado existir no sistema em que o agente de origem está localizado. É possível configurar uma transferência de arquivos acionada do IBM MQ Explorer ou da linha de comandos.

[“Configurando um tempo limite para recuperação de transferências paralisadas” na página 335](#)

É possível configurar um tempo limite de recuperação de transferência para transferências de arquivos paralisadas que se aplicam a todas as transferências de um agente de origem. Também é possível configurar um tempo limite de recuperação de transferência de uma transferência individual. Se você configurar um período de tempo específico, em segundos, durante o qual um agente de origem continue tentando recuperar uma transferência de arquivos paralisada e a transferência não for bem-sucedida quando o agente atingir o tempo limite, a transferência falhará.

Referências relacionadas

[**fteCreateTransfer**: iniciar uma nova transferência de arquivos](#)

[Formato da mensagem de solicitação para transferência de arquivos](#)

[Diretrizes para Transferências de Arquivos](#)

Utilizando Arquivos de Definição de Transferência

É possível especificar um arquivo de definição de transferência que pode ser usado para criar uma transferência de arquivo. O arquivo de definição de transferência é um arquivo XML que define algumas ou todas as informações necessárias para criar a transferência.

A transferência de arquivos de definição é útil quando você deseja especificar múltiplos arquivos de origem e múltiplos arquivos de destino em uma única operação de transferência. É possível usar um

arquivo de definição de transferência para enviar uma transferência de arquivos complexa. É possível reutilizar e compartilhar o arquivo de definição de transferência.

É possível usar dois formatos para um arquivo de definição de transferência, e enquanto esses formatos variam um pouco, ambos estão em conformidade com o esquema `FileTransfer.xsd`. É possível encontrar esse esquema no diretório do `samples\schema` da instalação do Managed File Transfer.

Os seguintes dois formatos de arquivos de definição de transferência são suportados:

- Uma definição dos arquivos de origem e de destino para uma transferência. Essa definição usa um elemento **transferSpecifications** como a raiz.
- Uma definição da transferência inteira, incluindo arquivos de origem e de destino e os agentes de origem e de destino. Essa definição usa um elemento **request** como a raiz.
 - Arquivos com este formato podem ser gerados a partir do comando **fteCreateTransfer** usando o parâmetro **-gt**.

O exemplo a seguir mostra um formato de arquivo de definição de transferência que especifica apenas os arquivos de origem e de destino para uma transferência:

```
<?xml version="1.0" encoding="UTF-8"?>
<transferSpecifications xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <item checksumMethod="MD5" mode="text">
    <source recursive="false" disposition="leave">
      <file>textTransferTest.txt</file>
    </source>
    <destination type="directory" exist="overwrite">
      <file>c:\targetfiles</file>
    </destination>
  </item>
</transferSpecifications>
```

Para enviar esse formato de arquivo de definição de transferência é necessário especificar os agentes de origem e de destino na linha de comandos:

```
fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml
```

O exemplo a seguir é um formato de arquivo de definição de transferência que especifica todas as informações necessárias para uma transferência:

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="3.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>fteuser</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="agent1qm"/>
    <destinationAgent agent="AGENT2" QMgr="agent2qm"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:\sourcefiles\*.jpg</file>
        </source>
        <destination type="directory" exist="error">
          <file>/targetfiles/images</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

É possível gerar um arquivo com este formato usando o parâmetro **-gt** no comando **fteCreateTransfer**. Ao enviar um arquivo de definição de transferência com esse formato, não é necessário especificar nada mais na linha de comandos:

```
fteCreateTransfer -td c:\definitions\example2.xml
```

É possível substituir informações do agente de origem e de destino sobre a linha de comandos, transmitindo os parâmetros normais além do arquivo de definição de transferência. Por exemplo:

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

Este exemplo usa as opções da linha de comandos para substituir o agente de destino definido no arquivo de definição de transferência com **AGENT9** e o gerenciador de filas de destino definido no arquivo de definição de transferência como **agent9qm**.

Os dois formatos descritos podem conter um ou mais elementos <item>. Para obter mais informações sobre o elemento <item>, consulte [Formato de mensagem de solicitação de transferência](#). Cada um desses itens de transferência define um par de arquivos de origem e de destino com atributos adicionais para controlar o comportamento da transferência. Por exemplo, você pode especificar o seguinte comportamento:

- Se a transferência utilizará uma soma de verificação
- Se a transferência será de texto ou binária
- Se o arquivo de origem deverá ser excluído após a conclusão da transferência
- Se o arquivo de destino será sobrescrito, se existir

Uma vantagem de usar os arquivos de definição de transferência é que é possível especificar opções adicionais que não estão disponíveis na linha de comandos. Por exemplo, ao executar transferências de mensagem para arquivo, é possível especificar o atributo `groupId` usando um arquivo de definição de transferência. Esse atributo especifica o ID do grupo do IBM MQ das mensagens que são lidas na fila. Uma outra vantagem dos arquivos de definição de transferência é que é possível especificar opções diferentes para cada par de arquivos. Por exemplo, é possível especificar se será usada uma soma de verificação ou se o arquivo será transferido em modo de texto ou binário, em uma base arquivo por arquivo. Se a linha de comandos for usada, as mesmas opções se aplicam a cada arquivo em uma transferência.

Por exemplo:

```
<item checksumMethod="none" mode="binary">
  <source disposition="leave">
    <file>c:\sourcefiles\source1.doc</file>
  </source>
  <destination type="file" exist="error">
    <file>c:\destinationfiles\destination1.doc</file>
  </destination>
</item>

<item checksumMethod="MD5" mode="text">
  <source disposition="delete">
    <file>c:\sourcefiles\source2.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file encoding="UTF8" EOL="CRLF">c:\destinationfiles\destination2.txt</file>
  </destination>
</item>

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>c:\originfiles\source3.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file>c:\targetfiles\destination3.txt</file>
  </destination>
</item>
```

z/OS É possível usar itens para transferir um arquivo de um sistema distribuído para um sistema z/OS:

z/OS

```
<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>textTransferTest.txt</file>
  </source>
  <destination type="dataset" exist="overwrite">
    <file encoding="IBM-1047">//TEXT.TRANS.TEST</file>
  </destination>
</item>
```

z/OS Este exemplo transfere o arquivo `textTransferTest.txt` do agente de origem para o conjunto de dados do `//TEXT.TRANS.TEST` no agente de destino no modo texto. Esta transferência converte os dados de origem da codificação padrão do agente de origem (nenhum atributo de codificação de origem é especificado) na página de códigos: IBM-1047.

Criando uma Transferência de Arquivo Planejada

É possível planejar uma nova transferência de arquivos por meio do IBM MQ Explorer ou da linha de comandos. A transferência planejada pode conter arquivos individuais ou um grupo com vários arquivos. É possível executar uma transferência de arquivos uma vez ou repeti-la várias vezes.

Sobre esta tarefa

É possível configurar um planejamento de transferência de arquivo para ocorrer uma vez ou nos seguintes intervalos:

- A cada minuto
- A cada hora
- A cada dia
- A cada semana
- A cada mês
- A cada ano

É possível especificar, então, as paradas das ocorrências para os seguintes pontos:

- Em uma data e hora definida
- Depois de um número definido de ocorrências

Como alternativa, é possível especificar se a ocorrência continuará indefinidamente.

Se uma transferência planejada for executada no mesmo horário todos os dias, use o atributo **adjustScheduleTimeForDaylightSaving** no arquivo de propriedades do agente para ajustar a hora em que o planejamento ocorre quando os relógios mudam. Consulte [O arquivo MFT agent.properties](#) para obter mais informações.

Para criar uma nova transferência de arquivo planejada usando a linha de comando, use os parâmetros de agendamento (**-tb**, **-ss**, **-oi**, **-of**, **-oc** e **-es**) para o [comando fteCreateTransfer](#).

Para criar uma nova transferência de arquivos planejada usando o assistente **Criar nova transferência de arquivos gerenciada** no IBM MQ Explorer, use as etapas a seguir:

Procedimento

1. Na visualização do Navegador, clique em **Transferência de Arquivos Gerenciados**. A **Central de Transferências de Arquivos Gerenciados** é exibida na visualização Conteúdo.
2. Todos os gerenciadores de filas da coordenação são exibidos na visualização Navegador. Expanda o nome do gerenciador de filas da coordenação do registrado para o agente que você deseja utilizar para

a transferência. Se você estiver atualmente conectado a um gerenciador de filas da coordenação que não seja o que deseja utilizar para a transferência, clique com o botão direito do mouse nesse nome do gerenciador de filas da coordenação na visualização Navegador e clique em **Desconectar**. Em seguida, clique com o botão direito do mouse no nome do gerenciador de filas da coordenação que você deseja utilizar e clique em **Conectar**.

3. Inicie o assistente **Criar Nova Transferência de Arquivo Gerenciada** utilizando um dos seguintes métodos:
 - a) Clique com o botão direito do mouse no nome de qualquer um dos seguintes nós na visualização Navegador: o respectivo gerenciador de filas de coordenação, **Modelos de Transferência**, **Log de Transferência** ou **Transferências Pendentes**. Em seguida, clique em **Nova Transferência** para iniciar o assistente.
 - b) Clique em **Arquivo > Novo > Outro > Assistentes Transferência de Arquivos Gerenciados > Assistente Nova Transferência**
4. Siga as instruções nos painéis do assistente. Certifique-se de selecionar a caixa de seleção **Ativar transferência de planejamento** e insira os detalhes do planejamento na guia **Planejamento**. O arquivo planejado transfere o início dentro de um minuto do horário de início do planejamento, desde que não haja nenhum problema que possa afetar a transferência. Por exemplo, pode haver problemas com a rede ou agente que impeçam o início da transferência planejada. Também é fornecida em cada painel ajuda sensível ao contexto. Para acessar a ajuda contextual no Windows, pressione F1. No Linux, pressione Ctrl+F1 ou Shift+F1.

Resultados

Para obter informações sobre as mensagens envolvidas em transferências de arquivos planejadas, veja [Formatos da mensagem de log de transferência de arquivos planejada](#).

Trabalhando com Transferências de Arquivos Pendentes

É possível visualizar transferências de arquivos planejadas que estão pendentes do IBM MQ Explorer. A janela **Transferências Pendentes** exibe todas as transferências pendentes registradas com o gerenciador de filas de coordenação ao qual você está conectado atualmente.

Sobre esta tarefa

Para visualizar o status de uma transferência de arquivos planejada ainda não iniciada, utilize as seguintes etapas:

Procedimento

1. Expanda **Transferência de Arquivos Gerenciados** na visualização Navegador. A **Central de Transferências de Arquivos Gerenciados** é exibida na visualização Conteúdo.
2. Todos os gerenciadores de filas da coordenação são exibidos na visualização Navegador. Expanda o nome do gerenciador de filas da coordenação que você utilizou para a transferência planejada. Para alterar o gerenciador de filas de coordenação ao qual você está conectado, clique com o botão direito do mouse no nome do gerenciador de filas de coordenação, para usar a visualização Navegador, e clique em **Conectar**.
3. Clique em **Transferências Pendentes**. A janela **Transferências Pendentes** é exibida na visualização Conteúdo.
4. A janela **Transferências Pendentes** exibe os seguintes detalhes sobre suas transferências de arquivos planejadas:
 - a) **Nome** O número da transferência de arquivo planejada. Esse número é atribuído automaticamente.
 - b) **Origem** O nome do agente e origem.
 - c) **Arquivo de Origem** O nome do arquivo a ser transferido em seu sistema de host.
 - d) **Destino** O nome do agente de destino.

- e) **Arquivo de Destino** O nome do arquivo depois de ser transferido para o sistema de destino.
- f) **Início Planejado (Fuso Horário Selecionado)** A data e hora em que a transferência de arquivo está planejada para iniciar no fuso horário selecionado do administrador. Para mudar o fuso horário exibido, clique em **Janela > Preferências > IBM MQ Explorer > Transferência de arquivos gerenciada** e selecione um fuso horário alternativo na lista **Fuso horário:**. Clique em **OK**.
- g) **Repetir Todos** Caso tenha optado por repetir a transferência planejada, o intervalo especificado (como número) em que deseja repetir a transferência.
- h) **Repetir Tipo** Caso tenha optado por repetir a transferência planejada, o tipo de intervalo de repetição especificado para a transferência de arquivo. O tipo pode ser um dos seguintes valores: minutos, horas, dias, semanas, meses ou anos.
- i) **Repetir Até** Caso tenha optado por repetir a transferência planejada, os detalhes de quando deseja que a repetição da transferência pare. Por exemplo, uma data e hora especificada, ou depois de determinado número de ocorrências.

Resultados

Para atualizar o que é exibido na janela **Transferências Pendentes**, clique no botão Atualizar  na barra de ferramentas da visualização Conteúdo.

Para cancelar uma transferência de arquivo pendente, clique com o botão direito do mouse em determinada transferência e clique em **Cancelar**. Cancelar uma transferência completamente descarta totalmente o pedido de transferência de arquivo.

Acionando a Transferência de Arquivo

É possível configurar que determinadas condições de acionamento em uma transferência de arquivo sejam verdadeiras para que a transferência ocorra. Se as condições de acionamento não forem verdadeiras, a transferência de arquivos não ocorrerá e uma mensagem de log será enviada opcionalmente para registrar o fato de que a transferência não ocorreu. O pedido de transferência de arquivo será então, descartado. Por exemplo, é possível configurar uma transferência de arquivos que ocorrerá apenas se um arquivo nomeado no sistema em que o agente de origem está localizado exceder um tamanho especificado, ou se um determinado arquivo nomeado existir no sistema em que o agente de origem está localizado. É possível configurar uma transferência de arquivos acionada do IBM MQ Explorer ou da linha de comandos.

Sobre esta tarefa

É possível monitorar um recurso continuamente para que uma condição acionadora seja atendida. Para obter informações adicionais sobre monitoramento de recursos, consulte: [“Monitorando recursos do MFT” na página 232](#).

Há três condições de acionamento diferentes que podem ser configuradas. As condições são as que seguem:

- Se um determinado arquivo existir no mesmo sistema que o agente de origem
- Se um determinado arquivo não existir no mesmo sistema que o agente de origem
- Se um determinado arquivo estiver acima do tamanho no sistema em que o agente de origem está localizado (o tamanho pode ser expresso em bytes, KB, MB ou GB). Essas unidades de medida usam a convenção ¹⁰, por exemplo, 1 KB igual a 1024 bytes e 1 MB igual a 1024 KB.

Os tipos de acionamentos na lista anterior podem ser combinados de duas maneiras:

- Para uma única condição, é possível especificar mais de um arquivo no sistema em que o agente de origem está localizado. Isso acionará a transferência se qualquer um dos arquivos satisfizer a condição (operador booleano OR).
- É possível especificar várias condições. Isso acionará a transferência apenas se todas as condições forem satisfeitas (operador booleano AND).

Também é possível combinar uma transferência acionada com uma transferência planejada. Consulte [Criando uma transferência de arquivo planejada](#) para obter informações adicionais. Nesse caso, as condições de acionamento serão avaliadas no momento programado para o início do planejamento ou em cada início programado (caso seja um planejamento que se repete).

Transferências acionadas não são suportadas em agentes de ponte de protocolo.

Para criar uma transferência de arquivo acionada usando a linha de comando, use o parâmetro **-tr** no comando `fteCreateTransfer`.

Para criar uma transferência de arquivos planejada usando o assistente **Criar nova transferência de arquivos gerenciada** no IBM MQ Explorer, use as etapas a seguir:

Procedimento


1. Na visualização do Navegador, clique em **Transferência de Arquivos Gerenciados**. A **Central de Transferências de Arquivos Gerenciados** é exibida na visualização Conteúdo.
2. Todos os gerenciadores de filas da coordenação são exibidos na visualização Navegador. Expanda o nome do gerenciador de filas da coordenação que você utilizou para a transferência planejada. Para alterar o gerenciador de filas de coordenação ao qual você está conectado, clique com o botão direito do mouse no nome do gerenciador de filas de coordenação, para usar a visualização Navegador, e clique em **Conectar**.
3. Inicie o assistente **Criar Nova Transferência de Arquivos Gerenciada** usando qualquer um dos seguintes métodos:
 - a) Clique com o botão direito do mouse no nome de qualquer um dos seguintes nós na visualização Navegador: o respectivo gerenciador de filas de coordenação, **Modelos de Transferência**, **Log de Transferência** ou **Transferências Pendentes**. Em seguida, clique em **Nova Transferência** para abrir o assistente.
 - b) Clique em **Arquivo > Novo > Outro > Assistentes Transferência de Arquivos Gerenciados > Assistente Nova Transferência**
4. Siga as instruções nos painéis do assistente. Certifique-se de selecionar a caixa de seleção **Ativar transferência acionada** na guia **Acionadores** e de preencher os campos nessa guia para configurar o acionamento. Também é fornecida em cada painel ajuda sensível ao contexto. Para acessar a ajuda contextual no Windows, pressione F1. No Linux, pressione **Ctrl+F1** ou Shift+F1.

Monitorando Transferências de Arquivos que Estão em Andamento

É possível monitorar uma transferência de arquivos que está em andamento usando a guia **Transferência de arquivos gerenciada - Progresso da transferência atual** no IBM MQ Explorer. Esta transferência de arquivos pode ser uma iniciada a partir do IBM MQ Explorer ou da linha de comandos. A guia também exibe o progresso da transferências planejadas no ponto em que a transferência inicia.

Sobre esta tarefa

Se desejar usar o IBM MQ Explorer para monitorar transferências associadas a um gerenciador de filas de coordenação em um sistema remoto, siga as instruções no tópico [“Configurando o IBM MQ Explorer para monitorar um gerenciador de filas de coordenação remoto”](#) na página 229.


As informações de transferência de arquivos anteriores não são retidas após você parar e reiniciar o IBM MQ Explorer. No reinício, as informações sobre transferências decorridas são limpas a partir da guia **Progresso de Transferência Atual**. É possível limpar transferências concluídas usando **Remover transferências concluídas**  em qualquer ponto quando IBM MQ Explorer estiver aberto.


Procedimento

Após ter iniciado uma nova transferência de arquivos usando o IBM MQ Explorer ou a linha de comandos, é possível monitorar o progresso de sua transferência na guia **Progresso da transferência atual**. As informações a seguir são exibidas para cada transferência em progresso:

- a) **Origem.** O nome do agente utilizado para transferir o arquivo do sistema de origem.
- b) **Destino.** O nome do agente utilizado para receber o arquivo no sistema de destino.
- c) **Arquivo Atual.** O nome do arquivo que está sendo transferido atualmente. A parte do arquivo individual que já foi transferida é exibida em B, KiB, MiB, GiB ou TiB junto com o tamanho total do arquivo em parêntesis. A unidade de medição exibida dependendo do tamanho do arquivo.
B é bytes por segundo. KiB/s é kibibytes por segundo, onde 1 kibibyte é igual a 1024 bytes. MiB/s é mebibytes por segundo, onde 1 mebibyte é igual a 1.048.576 bytes. GiB/s é gibibytes por segundo, onde 1 gibibyte é igual a 1.073.741.824 bytes. TiB/s é tebibytes por segundo onde 1 tebibyte é igual a 1.099.511.627.776 bytes.
- d) **Número do Arquivo.** Se você estiver transferindo mais de um arquivo, esse número representa quanto no grupo total de arquivos a transferência avançou.
- e) **Progresso.** A barra de progresso mostra a conclusão da transferência atual do arquivo em porcentagem.
- f) **Taxa.** A taxa do arquivo que está sendo transferido em KiB/s (kibibytes por segundo, onde 1 kibibyte é igual a 1024 bytes).
- g) **Iniciado (Fuso Horário Selecionado).** O horário em que a transferência de arquivo iniciou, apresentado no fuso horário selecionado do administrador. Para mudar o fuso horário exibido, clique em **Janela > Preferências > IBM MQ Explorer > Transferência de arquivos gerenciada** e selecione um fuso horário alternativo na lista **Fuso horário**. Clique em **OK**.
Se a transferência entra em um estado de recuperação ao transferir o arquivo, o horário de início é atualizado para refletir o horário em que a transferência de arquivo foi retomada.

Resultados

Essa guia atualiza regularmente suas informações automaticamente, mas para forçar uma visualização atualizada do que é exibido na guia **Progresso da transferência atual**, clique em **Atualizar**  na barra de ferramentas da visualização Conteúdo.

Para excluir transferências de arquivos da guia **Progresso da Transferência Atual**, clique em **Remover Transferências Concluídas**  na barra de ferramentas da visualização Conteúdo. Clicar nesse botão apenas removerá os detalhes da transferência de arquivos da guia, e não parará nem cancelará a transferência atual ou planejada.

Se você quiser retornar à guia **Progresso da Transferência Atual** depois de fechá-la, poderá exibir a guia clicando em **Janela > Mostrar Visualização > Outro > Outro > Transferência de Arquivos Gerenciados - Progresso da Transferência Atual**. Clique em **OK**.

Como proceder a seguir

Além disso, é possível desenvolver aplicativos para monitoramento de transferência de arquivos customizados. Isso pode ser feito criando uma assinatura do tópico administrativo apropriado do Managed File Transfer (de forma programática ou administrativa) e o aplicativo monitor pode então receber as publicações da atividade de transferência de arquivos do Managed File Transfer no tópico. Para obter mais informações sobre o tópico de assinatura e o formato da mensagem de publicação, veja [Exemplos de mensagem de progresso de transferência de arquivos](#).

Tarefas relacionadas

[“Configurando o IBM MQ Explorer para monitorar um gerenciador de filas de coordenação remoto” na página 229](#)

Use o IBM MQ Explorer para monitorar transferências de arquivos associadas a um gerenciador de filas de coordenação em execução em um sistema remoto. Você requer um sistema capaz de executar o IBM MQ Explorer. O componente IBM MQ Explorer precisa ser instalado para ser capaz de se conectar ao gerenciador de filas de coordenação remoto.

[“Visualizando o Status de Transferências de Arquivos no Log de Transferência” na página 229](#)

É possível visualizar os detalhes de transferências de arquivos usando o **Log de transferência** no IBM MQ Explorer. Essas transferências podem ser aquelas iniciadas da linha de comandos ou do IBM MQ Explorer. Também é possível customizar o que é exibido no **Log de Transferência**.

Configurando o IBM MQ Explorer para monitorar um gerenciador de filas de coordenação remoto

Use o IBM MQ Explorer para monitorar transferências de arquivos associadas a um gerenciador de filas de coordenação em execução em um sistema remoto. Você requer um sistema capaz de executar o IBM MQ Explorer. O componente IBM MQ Explorer precisa ser instalado para ser capaz de se conectar ao gerenciador de filas de coordenação remoto.

Sobre esta tarefa

Suposições: Autoridade para conectar-se ao gerenciador de filas de coordenação remota ao configurar o gerenciador de filas para permitir conexões remotas.

Para obter mais informações sobre como configurar isso, consulte [Conectando-se a um gerenciador de filas no modo cliente com autenticação de canal](#) e [Gerenciando autoridades para recursos específicos do MFT](#).

Para monitorar gerenciadores de filas e transferências de arquivos entre agentes em um sistema que não está executando o Windows ou o Linux, configure o IBM MQ Explorer para se conectar ao sistema remoto usando as etapas a seguir:

Procedimento

1. Inicie o IBM MQ Explorer local.
2. Quando o IBM MQ Explorer for carregado, clique com o botão direito na pasta **Managed File Transfer** e selecione **Nova configuração**.
3. Continue com o assistente, selecionando o gerenciador de filas de Coordenação e Comandos, em seguida, defina um nome para a configuração.
4. Clique em **concluir** para concluir a definição.
5. Quando a definição estiver concluída, clique com o botão direito do mouse na definição e selecione **Conectar**.

Resultados

Agora inicie o IBM MQ Explorer e use-o para monitorar a atividade de transferência para a rede do Managed File Transfer associada ao gerenciador de filas de coordenação.

Tarefas relacionadas

[“Monitorando Transferências de Arquivos que Estão em Andamento” na página 227](#)

É possível monitorar uma transferência de arquivos que está em andamento usando a guia **Transferência de arquivos gerenciada - Progresso da transferência atual** no IBM MQ Explorer. Esta transferência de arquivos pode ser uma iniciada a partir do IBM MQ Explorer ou da linha de comandos. A guia também exibe o progresso da transferências planejadas no ponto em que a transferência inicia.


[“Visualizando o Status de Transferências de Arquivos no Log de Transferência” na página 229](#)

É possível visualizar os detalhes de transferências de arquivos usando o **Log de transferência** no IBM MQ Explorer. Essas transferências podem ser aquelas iniciadas da linha de comandos ou do IBM MQ Explorer. Também é possível customizar o que é exibido no **Log de Transferência**.

Visualizando o Status de Transferências de Arquivos no Log de Transferência

É possível visualizar os detalhes de transferências de arquivos usando o **Log de transferência** no IBM MQ Explorer. Essas transferências podem ser aquelas iniciadas da linha de comandos ou do IBM MQ Explorer. Também é possível customizar o que é exibido no **Log de Transferência**.



Procedimento

1. Expanda **Transferência de Arquivos Gerenciada** na visualização Navegador e, em seguida, expanda o nome do gerenciador de filas da coordenação para o qual deseja visualizar o log de transferência.
2. Clique em **Log de Transferência** na visualização do Navegador. O **Log de Transferências** é exibido na visualização Conteúdo.
3. A janela **Log de Transferências** exibe os seguintes detalhes sobre as transferências de arquivos:
 - a) **Origem** O nome do agente no sistema em que o arquivo de origem está localizado.
 - b) **Destino** O nome do agente no sistema para o qual deseja transferir o arquivo.
 - c) **Estado de Conclusão** O status da transferência de arquivos. O estado pode ser um dos seguintes valores: "Iniciado", "Em progresso", "Bem-sucedido", "Parcialmente Bem-sucedido", "Cancelado" ou "Com Falha".
 - d) **Proprietário** O ID do usuário no host que submeteu o pedido de transferência.
 - e) **Iniciado (fuso horário selecionado)** O horário e a data em que a solicitação de transferência de arquivos foi aceita pelo agente do Managed File Transfer, apresentado no fuso horário selecionado do administrador. Para mudar o fuso horário exibido, clique em **Janela > Preferências > IBM MQ Explorer > Managed File Transfer** e selecione um fuso horário alternativo na lista **Fuso horário**. Clique em **OK**.
 - f) **Estado Registrado (fuso horário selecionado)** (Esta coluna não é exibida por padrão. É possível escolher para exibir a coluna usando a janela **Configurar Colunas do Log de Transferência** ). A hora e data de registro do estado de conclusão, no fuso horário selecionado pelo administrador.
 - g) **Nome da tarefa** Um identificador especificado pelo usuário usando o parâmetro **-jn** de **fteCreateTransfer** ou em um script do Ant
 - h) **ID de Transferência** O identificador exclusivo para a transferência de arquivos.
 - i) **Conexão: Direta** Detalhes sobre **Número do Processo, Nome do Processo, Nó Primário, Nó Secundário, Tipo de Origem e Tipo de Destino** são listados.

Resultados

Nota: O formato interno do Log de transferência foi mudado no IBM MQ 8.0.0 Fix Pack 1 para o APAR IC99545. Como resultado, se um IBM MQ Explorer for submetido a upgrade para a V8.0.0.1 ou mais recente e, em seguida, restaurado para a V8.0.0.0, nenhum XML de auditoria será exibido para transferências ocorridas enquanto o IBM MQ Explorer estava na V8.0.0.1. O painel XML na janela **Propriedades** para essas transferências conterá uma caixa de texto vazia.

Para ver mais detalhes sobre uma transferência concluída, expanda a transferência de seu interesse clicando no sinal de mais (+). É possível ver todos os nomes de arquivo de origem e destino incluídos nessa transferência. No entanto, se a transferência estiver em andamento e consistir em muitos arquivos, será possível visualizar apenas os arquivos que já foram transferidos até o momento.

Para atualizar o que é exibido no **Log de transferência**, clique no botão **Atualizar**  na barra de ferramentas da visualização Conteúdo. As informações de transferência de arquivos no Log de transferência permanecerão no log depois que o IBM MQ Explorer for parado e reiniciado. Se desejar excluir todas as transferências de arquivos concluídas do log, clique em **Remover Transferências Concluídas**  na barra de ferramentas da visualização Conteúdo.

Para excluir uma transferência de arquivo concluída individual do log, clique com o botão direito do mouse na transferência e clique em **Excluir**. Se você excluir uma transferência, isto não parará ou cancelará uma transferência que está em andamento ou que foi planejada; estão sendo excluídos apenas os dados históricos armazenados.

Para copiar um identificador exclusivo de uma transferência para a área de transferência, clique com o botão direito do mouse nessa transferência e clique em **Copiar ID**.

Os metadados e o XML de auditoria completo para a transferência estão disponíveis no menu pop-up, sob a ação **Propriedades**.

Tarefas relacionadas

[“Monitorando Transferências de Arquivos que Estão em Andamento”](#) na página 227

É possível monitorar uma transferência de arquivos que está em andamento usando a guia **Transferência de arquivos gerenciada - Progresso da transferência atual** no IBM MQ Explorer. Esta transferência de arquivos pode ser uma iniciada a partir do IBM MQ Explorer ou da linha de comandos. A guia também exibe o progresso da transferências planejadas no ponto em que a transferência inicia.

[“Configurando o Log de Transferência”](#) na página 231

É possível configurar quais informações serão exibidas e como serão exibidas no **Log de transferência** no IBM MQ Explorer.

[“Configurando um tempo limite para recuperação de transferências paralisadas”](#) na página 335

É possível configurar um tempo limite de recuperação de transferência para transferências de arquivos paralisadas que se aplicam a todas as transferências de um agente de origem. Também é possível configurar um tempo limite de recuperação de transferência de uma transferência individual. Se você configurar um período de tempo específico, em segundos, durante o qual um agente de origem continue tentando recuperar uma transferência de arquivos paralisada e a transferência não for bem-sucedida quando o agente atingir o tempo limite, a transferência falhará.


Configurando o Log de Transferência

É possível configurar quais informações serão exibidas e como serão exibidas no **Log de transferência** no IBM MQ Explorer.


Sobre esta tarefa

Para reorganizar a ordem das colunas no **Log de Transferência**, clique no título da coluna que você deseja mover e arraste-a para a nova posição. A nova ordem da coluna será retida somente até você parar e reiniciar o IBM MQ Explorer da próxima vez.

Para filtrar as entradas no **Log de Transferência**, digite uma cadeia no campo **Filtrar as Entradas de Log Exibidas**. Para restaurar todas as entradas para o log, exclua a cadeia inserida no campo. É possível usar qualquer expressão regular Java válida nesse campo. Para obter mais informações, consulte [Expressões regulares usadas pelo MFT](#).

Para customizar quais colunas são exibidas no log de transferência, use **Configurar colunas do log de transferência** . Use as seguintes etapas para iniciar e usar a janela **Configurar Colunas do Log de Transferência**.

Procedimento

1. O **Log de Transferência** deve estar aberto na visualização Conteúdo. Clique em **Configurar colunas do log de transferência**  na barra de ferramentas de visualização Conteúdo. A janela **Configurar Colunas do Log de Transferência** é exibida.
2. Para customizar a visualização do **Log de Transferência**, selecione ou limpe caixas de seleção individuais para as colunas que você deseja mostrar ou ocultar. Pode-se clicar em **Selecionar Tudo** e, em seguida, em **OK** para selecionar todas as caixas de opções ou em **Desmarcar Tudo** e em **OK** para limpar todas as caixas de opções.

Tarefas relacionadas

[“Monitorando Transferências de Arquivos que Estão em Andamento”](#) na página 227

É possível monitorar uma transferência de arquivos que está em andamento usando a guia **Transferência de arquivos gerenciada - Progresso da transferência atual** no IBM MQ Explorer. Esta transferência de arquivos pode ser uma iniciada a partir do IBM MQ Explorer ou da linha de comandos. A guia também exibe o progresso da transferências planejadas no ponto em que a transferência inicia.

[“Visualizando o Status de Transferências de Arquivos no Log de Transferência”](#) na página 229

É possível visualizar os detalhes de transferências de arquivos usando o **Log de transferência** no IBM MQ Explorer. Essas transferências podem ser aquelas iniciadas da linha de comandos ou do IBM MQ Explorer. Também é possível customizar o que é exibido no **Log de Transferência**.

Monitorando recursos do MFT

É possível monitorar recursos do Managed File Transfer; por exemplo, uma fila ou um diretório. Quando uma condição nesse recurso é satisfeita, o monitor de recurso inicia a tarefa, como uma transferência de arquivos. É possível criar um monitor de recurso usando o comando **fteCreateMonitor** ou a visualização **Monitores** no plug-in do Managed File Transfer para o IBM MQ Explorer.

Sobre esta tarefa

O monitoramento de recursos do Managed File Transfer usa a seguinte terminologia:

Monitor de Recurso

Um monitor de recurso é o processo que pesquisa um recurso (como um diretório ou uma fila) em um intervalo regular predefinido para ver se o conteúdo do recurso mudou. Se tiver sido, o conteúdo será comparado com o conjunto de condições para esse monitor. Se houver uma correspondência, a tarefa para esse monitor será iniciada.

Recurso

O recurso do sistema que o monitor de recurso examina em cada intervalo de pesquisa a ser comparado com as condições acionadoras. Filas, diretórios ou estruturas de diretório aninhadas podem ser o recurso monitorado.

Condição e condição acionadora

Uma condição é uma expressão que é avaliada (tipicamente com relação ao conteúdo do recurso monitorado). Se a expressão for avaliada como verdadeira, a condição contribuirá para a condição acionadora geral.

A condição acionadora é a condição geral, que é atendida quando todas as condições são atendidas. Quando a condição acionadora é satisfeita, a tarefa pode prosseguir.

Tarefa

Uma tarefa é a operação iniciada quando a condição acionadora ou o conjunto de condições é atendido. As tarefas suportadas são transferência de arquivo e chamada de comando.

Arquivo acionador

Um arquivo acionador é aquele colocado em um diretório monitorado para indicar que uma tarefa (geralmente uma transferência) pode começar. Por exemplo, poderá indicar que todos os arquivos a serem processados chegaram em um local desconhecido e podem ser transferidos, ou então agir de acordo. O nome do arquivo acionador pode ser usado para especificar os arquivos a serem transferidos utilizando substituição de variável. Para obter informações adicionais, consulte [“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.”](#) na página 244.

O arquivo de acionador também é conhecido como arquivo pronto ou ativo. No entanto, nesta documentação, ele é referido geralmente como o arquivo acionador.

O monitoramento de recursos não é suportado em agentes de ponte de protocolo ou agentes de ponte Connect:Direct .

Conceitos relacionados

[Orientação para configurar um monitor de recurso do MFT para evitar sobrecarregar um agente](#)

Referências relacionadas

fteCreateMonitor: criar um monitor de recurso do MFT

fteListMonitors: listar monitores de recurso do MFT

fteDeleteMonitor: excluir um monitor de recurso do MFT

[Formatos de mensagem de solicitação do monitor do MFT](#)

Conceitos de monitoramento de recurso do MFT

Uma visão geral dos principais conceitos do recurso de monitoramento de recursos do Managed File Transfer.

Monitores de Recursos

Você cria um monitor de recurso usando o comando `fteCreateMonitor`, que cria e inicia um novo monitor de recurso na linha de comandos. O monitor de recursos está associado a um agente do Managed File Transfer e só está ativo quando esse agente está em execução. Quando o agente de monitoramento para, o monitor de recurso também para. Se o agente já está em execução quando o monitor de recursos é criado, o monitor de recursos inicia imediatamente. O agente de monitoramento também deve ser o agente de origem da tarefa que é iniciada pelo monitor de recurso.

Os nomes do monitor de recurso devem ser exclusivos dentro de seus agentes. O nome do monitor de recursos deve ter no mínimo um caractere de comprimento e não deve conter caracteres de asterisco (*), porcentagem (%) ou ponto de interrogação (?). A letra na qual um nome do monitor de recurso é fornecido é ignorada e o nome do monitor de recurso é convertido em maiúsculas. Se você tentar criar um monitor de recurso com um nome que já esteja presente, a solicitação será ignorada e a tentativa será registrada no tópico de log do monitor de recurso.

Nota: Não é possível criar um monitor de recurso com uma definição de tarefa que contenha transferências planejadas

Antes de IBM MQ 9.3.0, a única maneira de parar um monitor de recurso é parando o agente que está executando a operação do monitor. Para reiniciar um monitor de recurso, deve-se reiniciar completamente o agente. A partir do IBM MQ 9.3.0, é possível iniciar e parar monitores de recursos sem precisar parar ou reiniciar um agente. Para obter informações adicionais, consulte [“Iniciando e parando monitores de recursos”](#) na página 235.

Não há restrição no número de monitores de recurso que podem ser criados em um agente e todos são executados com a mesma prioridade. Considere as implicações da sobreposição de recursos monitorados, condições acionadoras em conflito e a frequência com que os recursos são pesquisados.

A sobreposição dos monitores de recurso pode causar:

- Possível contenção no local/itens de origem.
- Possíveis solicitações de transferência duplicada para os mesmos itens de origem.
- Erros inesperados ou falhas para transferências devido a conflitos de itens de origem.

Se vários monitores varrem o mesmo local e acionar nos mesmos itens, potencialmente é possível ocorrer o problema de dois monitores diferentes enviando solicitações de transferência gerenciada para o mesmo item.

Os monitores de recurso examinam o conteúdo dos recursos após cada período de intervalo de pesquisa. O conteúdo do recurso será comparado às condições acionadoras e, se elas forem atendidas, a tarefa associada ao monitor de recurso será chamada.

A tarefa é iniciada de modo assíncrono. Se houver uma correspondência de condição e a tarefa for iniciada, o monitor de recurso continuará a pesquisar mudanças adicionais nos conteúdos do recurso. Assim, por exemplo, se uma correspondência ocorria porque um arquivo chamado `reports.go` chegou em um diretório monitorado, a tarefa seria iniciada uma vez. No próximo intervalo de pesquisa, mesmo que o arquivo ainda exista, a tarefa não será iniciada novamente. No entanto, se o arquivo for excluído e, em seguida, colocado no diretório novamente, ou o arquivo for atualizado (de forma que o atributo da data da última modificação seja alterado), a próxima verificação de condição acionadora fará a tarefa ser chamada novamente.

Antes da IBM MQ 9.1.5, se um monitor de recurso executasse uma pesquisa que demorasse mais que o intervalo de pesquisa, isso significava que a próxima pesquisa seria iniciada assim que a atual fosse concluída sem diferença entre elas, o que poderia ter um efeito sobre a rapidez com que os monitores de recurso enviariam o trabalho para um agente. Isso poderá causar problemas de desempenho se os itens localizados durante a primeira pesquisa ainda estiverem lá quando a segunda ocorrer.

O monitor de recurso usa o serviço `ScheduledExecutor` inicia a próxima pesquisa somente após a conclusão da pesquisa anterior mais o tempo de intervalo de pesquisa configurado.. Isso significa que sempre haverá uma diferença entre os intervalos da pesquisa, em vez de ter outra pesquisa começando imediatamente após a pesquisa anterior se o tempo de pesquisa for mais longo que o intervalo da pesquisa.

Se um arquivo tiver falhado na transferência, será possível limpar o histórico do monitor de recurso, o que permite que outro pedido de transferência seja enviado sem a necessidade de excluir o arquivo e colocá-lo no diretório novamente ou atualizar o arquivo para alterar seu atributo de data da última modificação. Limpar o histórico é útil, por exemplo, em situações em que é necessário que o arquivo seja transferido, mas a modificação do arquivo não é possível. Para obter informações adicionais, consulte [“Limpando histórico do monitor de recurso”](#) na página 262.

Recursos

Os monitores de recurso no Managed File Transfer podem pesquisar o conteúdo dos dois tipos de recurso a seguir:

Diretórios ou estruturas de diretório aninhado

Um cenário comum é monitorar um diretório quanto a presença de arquivo acionador. Um aplicativo externo poderá estar processando vários arquivos e colocando-os em um diretório de origem conhecido. Quando o aplicativo tiver concluído seu processamento, ele indicará que os arquivos estão prontos para transferência, ou então colocará um arquivo acionador em um local monitorado. O arquivo de disparo pode ser detectado por um monitor de recurso Managed File Transfer e a transferência desses arquivos do diretório de origem para outro Managed File Transfer Agent é iniciada.

Por padrão, o diretório especificado é monitorado. Para também examinar subdiretórios, configure o nível de recursão no comando **fteCreateTransfer**.

Dois exemplos de monitoramento de um diretório são como a seguir:

- Monitore por um arquivo acionador (por exemplo `trigger.file`) e, em seguida, transfira um caractere curinga (por exemplo, `*.zip`).
- Monitore para `*.zip` e, em seguida, transfira `${FilePath}` (por exemplo, o arquivo que acionou a transferência). Para obter informações adicionais sobre a substituição de variável, consulte [“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.”](#) na página 244.

Nota: Não crie um monitor que monitore para `*.zip` e, em seguida, transfere `*.zip`. O monitor tenta iniciar uma transferência de `*.zip` para cada arquivo `.zip` em seu sistema. Ou seja, o monitor gera * número de transferências para `*.zip`.

Para obter um exemplo de criação de um monitor de recurso para monitorar um diretório, consulte [“Monitorando um Diretório e Usando a Substituição de Variável”](#) na página 241.

Filas do IBM MQ

Um exemplo de monitoramento de uma fila é que um aplicativo externo pode estar gerando mensagens e colocando-as em uma fila conhecida com o mesmo ID do grupo. Quando o aplicativo concluir a colocação dessas mensagens na fila, ele indicará que o grupo está completo. O grupo completo de mensagens pode ser detectado por um monitor de recurso Managed File Transfer e a transferência do grupo de mensagens da fila de origem para um arquivo é iniciada. Para obter um exemplo de como criar um monitor de recurso para monitorar uma fila, consulte [“Exemplo: configurando um recurso do MFT”](#) na página 243.

Nota: É possível especificar apenas um monitor por fila. Se você especificar mais de um monitor para pesquisar uma fila do IBM MQ, ocorrerá um comportamento imprevisível.

Conjuntos de dados de monitoramento não são suportados.

Condições e condições acionadoras

A condição é atendida quando o recurso contém um valor que corresponde a alguma outra cadeia ou padrão. As condições podem ser uma das seguintes:

- Correspondência de nome de arquivo (padrão)
- Sem correspondência de nome de arquivo (padrão)
- Tamanho do arquivo

- Corresponder se o tamanho do arquivo permanecer o mesmo para um número de conjuntos

A correspondência de nome de arquivo pode ser expressa como:

- Correspondência de cadeia exata
- Correspondência de curinga simples conforme descrito em [Usando caracteres curinga com o MFT](#)
- Correspondência de expressão regular

Os nomes de arquivos também podem ser excluídos da correspondência de nome do arquivo usando um curinga ou uma expressão regular Java que identifica nomes de arquivos que nunca correspondem.

Quando é detectado um arquivo de correspondência, o registro de data e hora de sua última modificação é retido. Se as pesquisas subsequentes detectarem que o arquivo foi alterado, a condição acionadora será atendida novamente e a tarefa será iniciada. Se a condição for para detectar quando um arquivo não existe, se nenhum arquivo no diretório monitorado corresponder ao padrão de nome de arquivo, a tarefa será iniciada. Se um arquivo for incluído no diretório que corresponde ao padrão de nome do arquivo, a tarefa será iniciada apenas se depois o arquivo for excluído.

Tarefas

O Managed File Transfer suporta os dois seguintes tipos de tarefa possíveis de configurar para serem iniciados por monitores de recurso:

Tarefas de transferência de arquivos

As tarefas de transferência de arquivo são definidas da mesma maneira que qualquer outra transferência de arquivo. Uma maneira útil de gerar o XML de tarefa exigido por um monitor é executar o comando `fteCreateTransfer` com o parâmetro **-gt**. Este comando gera uma definição da tarefa como um documento XML, incluindo a especificação de transferência. Em seguida, você passa o nome do documento XML da tarefa como o valor do parâmetro **-mt** no comando `fteCreateMonitor`. Quando o **fteCreateMonitor** for executado, ele lerá o documento XML da tarefa. Após a execução do **fteCreateMonitor**, as mudanças feitas no arquivo XML da tarefa não serão usadas pelo monitor.

Quando usar uma tarefa de transferência de arquivo, poderá selecionar quantas condições do acionador serão especificadas em lote em uma tarefa. O padrão é para uma condição acionadora para iniciar uma tarefa. É possível executar o comando `fteCreateMonitor` com a opção **-bs** de selecionar o número de condições de disparo agrupadas em uma tarefa.

Tarefas de comando

As tarefas de comando podem executar scripts Ant, chamar programas executáveis ou executar tarefas JCL. Para obter mais informações, consulte [“Configurando tarefas de monitor do MFT para iniciar comandos e scripts”](#) na página 237.

Arquivos acionadores

É possível usar o conteúdo de um arquivo acionador em um monitor de recurso para definir um conjunto de arquivos para transferência em uma única solicitação de transferência. Sempre que um arquivo acionador correspondente é detectado, seu conteúdo é analisado para caminhos do arquivo de origem e, opcionalmente, para caminhos do arquivo de destino. Estes caminhos do arquivo são então usados para definir itens do arquivo no arquivo XML de transferência de tarefa especificado, que é enviado como uma única solicitação de transferência para o agente. A definição do monitor de recurso determina se o conteúdo do acionador é ativado.

O formato de cada arquivo acionador é um caminho do arquivo único para transferência em cada linha de texto. O formato padrão para a linha é um caminho do arquivo de origem único ou um caminho do arquivo de origem e de destino, separado por uma vírgula.

Para obter informações e exemplos adicionais, consulte [“Usando um arquivo acionador”](#) na página 253.

Iniciando e parando monitores de recursos

Antes de IBM MQ 9.3.0, a única maneira de parar um monitor de recurso é parando o agente que está executando a operação do monitor. Para reiniciar um monitor de recurso, deve-se reiniciar

completamente o agente. Para obter mais informações, consulte o [“Iniciando um agente MFT”](#) na página 213 e o [“Parando um agente MFT”](#) na página 219.

No IBM MQ 9.3.0, é possível iniciar e parar os monitores de recursos sem precisar parar ou reiniciar um agente usando os comandos **fteStartMonitor** e **fteStopMonitor**. Isso é útil, por exemplo, nas situações a seguir:

- Se um agente tiver vários monitores de recursos, e somente alguns deles tiverem encontrado erros, mas os monitores de recursos restantes ainda estiverem funcionando bem, você desejará apenas reiniciar os monitores de recursos com falha.
- Se você deseja parar um monitor de recurso para realizar algum trabalho de manutenção, ou se o monitor de recurso não é necessário por determinado tempo e você não deseja que ele seja executado desnecessariamente, consumindo, assim, recursos preciosos do sistema.

Para obter mais informações, veja [“Iniciando um monitor de recurso do MFT”](#) na página 258 e [“Parando um monitor de recurso do MFT”](#) na página 259.

Comando:	Comportamento do monitor de recurso
fteStartMonitor	Se o agente estiver em execução, o monitor de recurso será iniciado se ele estiver atualmente interrompido.
fteStopMonitor	Se o agente estiver em execução, o monitor de recurso será interrompido se ele foi iniciado atualmente.
fteStartAgent	O monitor de recurso é iniciado como parte da inicialização do agente do, independentemente de chamadas anteriores para o fteStopMonitor
fteStopAgent	Quaisquer monitores de recurso que estão em execução são interrompidos.

Fazendo backup e restaurando monitores de recursos

É possível fazer backup dos monitores de recursos que você já definiu para que possa reutilizá-los futuramente. Existem várias opções que você pode usar como a seguir:

- Use o comando **fteCreateMonitor** com o parâmetro **-ox** para exportar uma configuração do monitor de recurso para um arquivo XML e, com o parâmetro **-ix**, para restaurar um monitor de recurso importando a configuração do monitor de recurso de um arquivo XML.
- Use o comando **fteListMonitors** com o **-ox** para exportar a definição de um único monitor de recursos para um arquivo XML.
- Use o comando **fteListMonitors** com o **-od** para exportar várias definições do monitor de recursos para um diretório especificado. Cada definição do monitor de recurso é salva no arquivo XML separado. Também é possível usar a opção **-od** para exportar uma única definição de monitor de recurso para um diretório especificado.

Para obter informações adicionais, consulte [“Fazendo backup e restaurando os monitores de recursos do MFT”](#) na página 260.

Criação de log do monitor de recurso

Managed File Transfer inclui criação de log do monitor de recurso. Para obter mais informações, consulte [“Criando logs de monitores de recursos do MFT”](#) na página 256.

Conceitos relacionados

[“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.”](#) na página 244

Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.

Tarefas relacionadas

[“Configurando tarefas de monitor do MFT para iniciar comandos e scripts” na página 237](#)

Monitores de recursos não estão limitados a executar transferências de arquivo como suas tarefas associadas. É também possível configurar o monitor para chamar outros comandos a partir do agente de monitoramento, incluindo programas executáveis, scripts Ant ou tarefas JCL. Para chamar comandos, edite o XML de definição de tarefa de monitoramento para incluir um ou mais elementos de comando nos parâmetros de chamada de comando correspondentes, como argumentos e propriedades.

[“Exemplo: configurando um recurso do MFT” na página 243](#)

É possível especificar uma fila do IBM MQ como o recurso a ser monitorado por um monitor de recurso usando o parâmetro **-mq** com o comando **fteCreateMonitor**.

[“Monitorando uma Fila e Usando Substituição de Variável” na página 249](#)

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

Referências relacionadas

fteCreateMonitor: [criar um monitor de recurso do MFT](#)

fteListMonitors: [listar monitores de recurso do MFT](#)

fteDeleteMonitor: [excluir um monitor de recurso do MFT](#)

Configurando tarefas de monitor do MFT para iniciar comandos e scripts

Monitores de recursos não estão limitados a executar transferências de arquivo como suas tarefas associadas. É também possível configurar o monitor para chamar outros comandos a partir do agente de monitoramento, incluindo programas executáveis, scripts Ant ou tarefas JCL. Para chamar comandos, edite o XML de definição de tarefa de monitoramento para incluir um ou mais elementos de comando nos parâmetros de chamada de comando correspondentes, como argumentos e propriedades.

Sobre esta tarefa

O caminho de arquivo para o programa executável, o script do Ant ou a tarefa JCL que você deseja que o agente de monitoramento chame deve ser incluído no `commandPath` do agente de monitoramento. Para obter informações sobre a propriedade do caminho de comando, consulte [Propriedade `commandPath` MFT](#).

É possível criar o documento XML de definição da tarefa de uma das seguintes maneiras:

- Crie o documento XML de definição de tarefa manualmente de acordo com o esquema `FileTransfer.xsd`.
- Use um documento XML gerado como a base de sua definição da tarefa.

Se você deseja uma tarefa de transferência ou uma tarefa de comando, a definição da tarefa deve começar com um elemento raiz `<request>`. O elemento-filho de `<request>` deve ser `<managedTransfer>` ou `<managedCall>`. Geralmente, você escolhe `<managedCall>` quando há um único comando ou script a ser executado, e `<managedTransfer>` se você deseja que a tarefa inclua uma transferência de arquivos e, opcionalmente, até quatro chamadas de comando.

Procedimento

- Para criar o documento XML de definição de tarefa manualmente de acordo com o esquema `FileTransfer.xsd`, consulte [“Criando um XML de definição de tarefa manualmente de acordo com o esquema”](#) na página 238.
- Para criar uma definição da tarefa modificando um documento gerado, edite o documento XML gerado pelo parâmetro **fteCreateTransfer -gt**. Para obter informações adicionais, consulte [“Criando um documento de definição de tarefa modificando um documento gerado”](#) na página 240.

Criando um XML de definição de tarefa manualmente de acordo com o esquema

É possível criar manualmente um arquivo XML de definição de tarefa de acordo com o esquema `FileTransfer.xsd`.

Sobre esta tarefa

O esquema `FileTransfer.xsd` pode ser encontrado no `MQ_INSTALLATION_PATH/mqft/samples/schema`. Para obter mais informações sobre esse esquema, veja [Formato da mensagem de solicitação de transferência de arquivos](#).

exemplo

O exemplo a seguir mostra um documento XML de definição de tarefa de exemplo salvo como `cleanuptask.xml`, que usa o elemento `<managedCall>` para chamar um script Ant denominado `RunCleanup.xml`. O script `RunCleanup.xml` Ant deve estar localizado no `commandPath` do agente de monitoramento.

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>hostName</hostName>
      <userID>userID</userID>
      <mqmdUserID>mqmdUserID</mqmdUserID>
    </originator>
    <agent QMgr="QM1" agent="AGENT1"/>
    <reply QMGR="QM1">reply</reply>
    <transferSet priority="1">
      <metaDataSet>
        <metaData key="name1">value1</metaData>
      </metaDataSet>
      <call>
        <command name="RunCleanup.xml" type="antscript" retryCount="2"
          retryWait="30" successRC="0">
          <target>check_exists</target>
          <target>copy_to_archive</target>
          <target>rename_temps</target>
          <target>delete_files</target>
          <property name="trigger.filename" value="{FileName}"/>
          <property name="trigger.path" value="{FilePath}"/>
        </command>
      </call>
    </transferSet>
  </managedCall>
</request>
```

O elemento `<agent>` especifica o Managed File Transfer Agent configurado com o script Ant nomeado em seu `commandPath`.

A estrutura `<call><command> . . .` define o executável ou o script que você deseja executar. O comando obtém um atributo `type` opcional que pode ter um dos seguintes valores:

antscript

Execute um script do Ant em uma JVM separada.

executable

Chame um programa executável.

jcl

Invoque uma tarefa JCL.

Se você omitir o atributo `type`, o valor padrão `executable` será usado.

O atributo `name` especifica o nome do script Ant, executável ou JCL que você deseja executar, sem qualquer informação de caminho. O agente procura o script ou programa nos locais especificados pela propriedade `commandPath` no arquivo `agent.properties` do agente.

O atributo `retrycount` especifica o número de vezes para tentar chamar o programa novamente se o programa não retornar um código de retorno de sucesso. O valor designado a esse atributo não deve ser negativo. Se você não especificar o atributo `retrycount`, será usado um valor padrão de zero.

O atributo `retrywait` especifica o tempo de espera, em segundos, antes de tentar chamar o programa novamente. O valor designado a esse atributo não deve ser negativo. Se você não especificar o atributo `retrywait`, será usado um valor padrão de zero.

O atributo `successrc` é uma expressão usada para determinar quando a chamada do programa é executada com sucesso. O código de retorno do processo para o comando é avaliado usando essa expressão. O valor pode ser composto de uma ou mais expressões combinadas com uma barra vertical (`|`) para significar o Booleano OR, ou um e comercial (`&`) para significar o Booleano AND. Cada expressão pode ser um dos seguintes tipos de expressão:

- um número para indicar um teste de igualdade entre o código de retorno e o número do processo.
- Um número prefixado com um caractere maior que (`>`) para indicar um teste de superioridade entre o número e o código de retorno do processo.
- Um número prefixado com um caractere de inferioridade (`<`) para indicar um teste de inferioridade entre o número e o código de retorno do processo.
- Um número prefixado com um caractere de ponto de exclamação (`!`) para indicar um teste diferente de entre o número e o código de retorno do processo. Por exemplo: `> 2 & < 7 &! 5 | 0 | 14` é interpretado como os códigos de retorno a seguir sendo bem-sucedidos: 0, 3, 4, 6, 14. Todos os outros códigos de retorno são interpretados como malsucedidos.

Se você não especificar o atributo `successrc`, será usado um valor padrão de zero. Isso significa que o comando de julgado como bem-sucedido se apenas se, retornar um código zero.

Para um script Ant, você geralmente especificaria elementos `<target>` e `<property>`. Os valores de elemento `<target>` devem corresponder aos nomes de destino no script Ant.

Para programas executáveis, é possível especificar elementos `<argument>`. Elementos de argumentos aninhados especificam os argumentos a serem transmitidos ao programa que está sendo chamado como parte da chamada do programa. Os argumentos do programa são criados a partir dos valores especificados pelos elementos de argumentos na ordem em que eles são encontrados. É possível especificar zero ou mais elementos do argumento como elementos aninhados de uma chamada de programa.

O administrador define e inicia o monitor como normal usando o documento XML de definição de tarefa que inclui o elemento `<managedCall>`. Por exemplo:

```
fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt
/tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*.go
```

O caminho para o documento XML de definição de transferência deve estar no sistema de arquivos local por meio do qual você executa o comando **fteCreateMonitor** (neste exemplo, `/tasks/cleanuptask.xml`). O documento `cleanuptask.xml` é usado para criar apenas o monitor de recursos. Todas as tarefas às quais o documento `cleanuptask.xml` faz referência (scripts Ant ou tarefas JCL) devem estar no caminho de comando do agente de monitoramento. Quando a condição acionadora do monitor é satisfeita, todas as variáveis no XML de definição de tarefa são substituídas pelos valores reais do monitor. Assim, por exemplo, `${FilePath}` é substituído na mensagem de solicitação enviada ao agente com `/monitored/cleanup.go`. A mensagem de solicitação é colocada na fila de comandos do agente. O processador de comando detecta se o pedido é para uma chamada de programa e inicia o programa especificado. Se um comando do tipo `antscript` for chamado, uma nova JVM será iniciada e a tarefa

Ant será executada na nova JVM. Para obter mais informações sobre como usar substituição de variável, consulte [Customizando Tarefas com Substituição de Variável](#).

Conceitos relacionados

“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.” na página 244
Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.

Referências relacionadas

[Formato da mensagem de solicitação para transferência de arquivos](#)

[Propriedade commandPath do MFT](#)

Criando um documento de definição de tarefa modificando um documento gerado

É possível criar o documento de definição da tarefa do monitor modificando o documento XML gerado pela opção **-gt** de **fteCreateTransfer**.

Sobre esta tarefa

O documento gerado tem um `<request>` seguido pelo elemento `<managedTransfer>`. Para converter essa definição de tarefa para uma estrutura `<managedCall>` válida, siga estas etapas:

Procedimento

1. Substitua as tags de início e final `<managedTransfer>` pelas tags `<managedCall>`.
2. Remova qualquer elemento `<schedule>` e nós filho.
3. Substitua as tags de início e final `<sourceAgent>` por `<agent>` para corresponder aos detalhes de configuração do agente de monitoramento.
4. Remova os elementos `<destinationAgent>` e `<trigger>`.
5. Remova os elementos `<item>`.
6. Remova quaisquer elementos `preSourceCall`, `postSourceCall`, `preDestinationCall` ou `postDestinationCall`.
7. Insira uma nova estrutura `<call>...</call>` dentro do elemento `<transferSet>`. Esta estrutura contém a definição de comando conforme mostrado no exemplo a seguir:

```
<call>
  <command name="RunCleanup.xml" type="antscript" retryCount="2"
  retryWait="30" successRC="0">
    <target>check_exists</target>
    <target>copy_to_archive</target>
    <target>rename_temps</target>
    <target>delete_files</target>
    <property name="trigger.filename" value="{FileName}"/>
    <property name="trigger.path" value="{FilePath}"/>
  </command>
</call>
```

exemplo

Também é possível reter o elemento `<managedTransfer>` incluindo todos os detalhes de transferência de arquivo e inserir até quatro chamadas de comandos. Neste caso, insira qualquer seleção dos elementos de chamada a seguir entre os elementos `<metaDataSet>` e `<item>`:

preSourceCall

Chame um programa no agente de origem antes de iniciar a transferência.

postSourceCall

Chame um programa no agente de origem depois de concluir a transferência.

preDestinationCall

Chame um programa no agente de destino antes de iniciar a transferência.

postDestinationCall

Chame um programa no agente de destino depois de concluir a transferência.

Cada um desses elementos leva a estrutura de elemento <command> conforme descrito no exemplo anterior. O esquema FileTransfer.xsd define os tipos usados pelos vários elementos de chamada.

O exemplo a seguir mostra preSourceCall, postSourceCall, preDestinationCall e postDestinationCall em um documento de definição de tarefa:

```

:
<transferSet priority="1">
  <metaDataSet>
    <metaData key="key1">value1</metaData>
  </metaDataSet>
  <preSourceCall>
    <command name="send.exe" retryCount="0" retryWait="0" successRC="0"
      type="executable">
      <argument>report1.pdf</argument>
      <argument>>true</argument>
    </command>
  </preSourceCall>
  <postSourceCall>
    <command name="//DO_IT.JCL" retryCount="0" retryWait="0" successRC="0"
      type="jcl">
      <argument>argument</argument>
    </command>
  </postSourceCall>
  <preDestinationCall>
    <command name="ant_script.xml" retryCount="0" retryWait="0" successRC="0"
      type="antscript">
      <target>step1</target>
      <property name="name" value="value"/>
    </command>
  </preDestinationCall>
  <postDestinationCall>
    <command name="runit.cmd" retryCount="0" retryWait="0" successRC="0" />
  </postDestinationCall>
  <item checksumMethod="none" mode="binary">
:

```

É possível combinar diferentes tipos de comando na transferência. Argumento, destino e elementos de propriedade são opcionais.

Monitorando um Diretório e Usando a Substituição de Variável

É possível monitorar um diretório usando o comando **fteCreateMonitor**. O valor de uma variável de substituição pode ser substituído na definição XML da tarefa e ser usado para definir o comportamento da transferência.

Sobre esta tarefa

Neste exemplo, o agente de origem é chamado AGENT_HOP. O diretório que o AGENT_HOP monitora é chamado de /test/monitored. O agente pesquisa o diretório a cada 5 minutos.

Depois que um arquivo .zip é gravado no diretório, o aplicativo que grava o arquivo no diretório grava um arquivo acionador para o mesmo diretório. O nome do arquivo de disparo é o mesmo que o nome do arquivo .zip, mas tem uma extensão de arquivo diferente. Por exemplo, depois que o arquivo file1.zip é gravado no diretório, o arquivo file1.go é gravado no diretório. O monitor de recursos monitora o diretório para arquivos que combinam com o padrão *.go então usa a substituição de variáveis para solicitar uma transferência do arquivo .zip associado.

Procedimento

1. Crie o XML de tarefa que define a tarefa que o monitor executa quando for acionado.

```
<?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>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/${fileName}{token=1}{separator=.}.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/${fileName}{token=1}{separator=.}.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

As variáveis que são substituídas pelos valores associados ao arquivo do acionador são destacadas em **negrito**. Esse XML de tarefa é salvo no arquivo /home/USER1/task.xml

2. Crie um monitor de recursos para monitorar o diretório /test/monitored.

Execute o seguinte comando:

```

fteCreateMonitor -ma AGENT_HOP -mm QM_HOP -md /test/monitored
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr match,*.go -pi 5 -pu minutes

```

3. Um usuário ou programa grava o arquivo jump.zip no diretório /test/monitored, em seguida, grava o arquivo jump.go no diretório.
4. O monitor é acionado pela existência do arquivo jump.go. O agente substitui as informações sobre o arquivo do acionador no XML da tarefa.

Isso resulta na transformação da tarefa XML to:

```

<?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>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/jump.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/jump.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

Resultados

A transferência definida pelo XML de tarefa é executada. O arquivo jump.zip é lido no diretório /test/monitored por AGENT_HOP e é transferido para um arquivo chamado /out/jump.zip localizado no sistema em que o AGENT_SKIP está em execução.

Conceitos relacionados

[“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.” na página 244](#)

Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.

Tarefas relacionadas

[“Configurando tarefas de monitor do MFT para iniciar comandos e scripts” na página 237](#)

Monitores de recursos não estão limitados a executar transferências de arquivo como suas tarefas associadas. É também possível configurar o monitor para chamar outros comandos a partir do agente de monitoramento, incluindo programas executáveis, scripts Ant ou tarefas JCL. Para chamar comandos, edite o XML de definição de tarefa de monitoramento para incluir um ou mais elementos de comando nos parâmetros de chamada de comando correspondentes, como argumentos e propriedades.

Referências relacionadas

[fteCreateMonitor](#): criar um monitor de recurso do MFT

Exemplo: configurando um recurso do MFT

É possível especificar uma fila do IBM MQ como o recurso a ser monitorado por um monitor de recurso usando o parâmetro **-mq** com o comando **fteCreateMonitor**.

Sobre esta tarefa

Neste exemplo, o recurso a ser monitorado é a fila *MONITORED_QUEUE*. Essa fila deve estar no gerenciador de filas do agente de monitoramento, *QM_NEPTUNE*. A condição pela qual a fila é monitorada é a presença de um grupo completo de mensagens. A tarefa a ser executada se a condição for satisfeita é definida no arquivo *task.xml*.

Nota: Não crie mais de um monitor de recurso para monitorar uma fila individual. Se você criar, ocorrerá um comportamento imprevisível.

Procedimento

Digite o seguinte comando:

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE  
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

O monitor verifica a fila a cada cinco minutos para ver se a condição *completeGroups* é verdadeira. Se houver um ou mais grupos completos na fila, o monitor executa a tarefa definida no arquivo *task.xml* uma vez para cada grupo completo.

Conceitos relacionados

[“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.” na página 244](#)

Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.

Tarefas relacionadas

[“Configurando tarefas de monitor do MFT para iniciar comandos e scripts” na página 237](#)

Monitores de recursos não estão limitados a executar transferências de arquivo como suas tarefas associadas. É também possível configurar o monitor para chamar outros comandos a partir do agente de monitoramento, incluindo programas executáveis, scripts Ant ou tarefas JCL. Para chamar comandos,

edite o XML de definição de tarefa de monitoramento para incluir um ou mais elementos de comando nos parâmetros de chamada de comando correspondentes, como argumentos e propriedades.

[“Monitorando uma Fila e Usando Substituição de Variável” na página 249](#)

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

Referências relacionadas

fteCreateMonitor: [criar um monitor de recurso do MFT](#)

Customizando as tarefas do monitor de recurso do MFT com substituição de variável.

Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.



Atenção: Os nomes de variáveis não fazem distinção entre maiúsculas e minúsculas.

As variáveis usadas para substituição estão disponíveis somente para condições acionadoras positivas. Apenas as condições acionadoras `match` e `fileSize` farão com que as variáveis sejam substituídas. Se uma condição `noMatch` for usada e houver nomes de variáveis de substituição na definição da tarefa, a tarefa não será chamada e o monitor gerará um código de retorno de 110 e uma mensagem de erro BFGDM0060E.

Se o Recurso Monitorado é uma Fila

O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa.

As propriedades de mensagem definidas pelo usuário são prefixadas com `usr.` mas não inclua este prefixo no nome da variável. Os nomes de variáveis devem ser precedidos por um caractere de símbolo de dólar (\$) e colocados entre chaves {}.

Por exemplo, `${destFileName}` é substituído pelo valor da propriedade da mensagem `usr.destFileName` da primeira mensagem a ser lida a partir da fila de origem. Para obter mais informações, veja [Propriedades de mensagem do MQ lidas pelo MFT de mensagens em filas de origem e “Monitorando uma Fila e Usando Substituição de Variável” na página 249](#).

Se uma variável não estiver definida como uma propriedade de mensagem, o monitor relatará um erro BFGDM0060E e retornará o código de retorno 110 (falha na substituição da variável da tarefa do Monitor). Além disso, o agente grava a seguinte mensagem de erro em seu log de eventos (`outputN.log`):

```
BFGDM0113W: Trigger failure for <monitor name> for reason BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Se o log do monitor de recursos moderado ou detalhado estiver ativado para o monitor, o monitor gravará a seguinte mensagem no log de eventos do monitor de recursos do agente (`resmoneventN.log`):

```
BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Veja [“Criando logs de monitores de recursos do MFT” na página 256](#) para obter mais informações sobre a criação de log do monitor de recursos.

A tabela a seguir mostra quais variáveis de substituição são fornecidas por padrão. Por exemplo, \$ {AGENTNAME} é substituído pelo nome do agente do monitor de recurso.

Tabela 10. Variáveis de substituição fornecidas por padrão

Variável	Descrição
AGENTNAME	O nome do agente do monitor de recurso.
QUEUENAME	O nome da fila que está sendo monitorada.
ENCODING	A codificação de caractere da primeira mensagem na fila ou da primeira mensagem em um grupo.
MESSAGEID	O ID de mensagem do IBM MQ da primeira mensagem na fila ou da primeira mensagem no grupo.
GROUPID	O ID do grupo do IBM MQ do grupo ou o ID de mensagem se somente uma única mensagem for localizada. Essa variável é configurada apenas se você estiver monitorando para grupos completos.
CurrentTimeStamp	Um registro de data e hora baseado no horário local no qual o monitor é acionado. O valor de registro de data e hora é exclusivo para o agente.
CurrentTimeStamp UTC	Um registro de data e hora baseado no horário, no fuso horário UTC, no qual o monitor é acionado. O valor de registro de data e hora é exclusivo para o agente.

Se o recurso monitorado for um diretório

A tabela a seguir mostra o conjunto de nomes de variáveis que podem ser substituídos na definição do XML da tarefa.

Tabela 11. Variáveis que podem ser substituídas

Variável	Descrição
FilePath	O nome do caminho completo do arquivo do acionador.
FileName	A parte de nome do arquivo do acionador.
LastModifiedTime	O horário em que o arquivo acionador foi alterado pela última vez. Esse horário é expresso como o horário local do fuso horário no qual o agente está em execução e é formatado como um horário ISO 8601.
LastModifiedDate	A data em que o arquivo acionador foi modificado pela última vez. Essa data é expressa como a data local do fuso horário no qual o agente está em execução e é formatada como uma data ISO 8601.
LastModifiedTimeUTC	O horário em que o arquivo acionador foi alterado pela última vez. Esse horário é expresso como o horário local convertido para o fuso horário UTC e é formatado como um horário ISO 8601.
LastModifiedDateUTC	A data em que o arquivo acionador foi modificado pela última vez. Esta data é expressa como a data local convertida para o fuso horário UTC e é formatada como uma data ISO 8601.
AgentName	O nome do agente do monitor de recurso.
CurrentTimeStamp	Um registro de data e hora baseado no horário local no qual o monitor é acionado. O valor de registro de data e hora é exclusivo para o agente.
CurrentTimeStampUTC	Um registro de data e hora baseado no horário do fuso horário UTC no qual o monitor é acionado. O valor de registro de data e hora é exclusivo para o agente.

Se o recurso monitorado for um arquivo acionador

A tabela a seguir mostra o conjunto de nomes de variáveis que podem ser substituídos quando um monitor de recursos está usando o conteúdo de um arquivo acionador para determinar os arquivos que precisam ser transferidos.

Variável	Descrição
contentSource	O nome completo do caminho do arquivo de origem.
contentDestination	O nome completo do caminho do arquivo de destino.

Os nomes de variáveis devem ser precedidos por um caractere cifrão (\$) e colocados entre chaves, {}. Por exemplo, `${FilePath}` é substituído pelo caminho de arquivo totalmente qualificado do arquivo acionador correspondente.

Há duas palavras-chave especiais que podem ser aplicadas a nomes de variáveis para fornecer refinamento melhor. São elas:

token

O índice de token a ser substituído (começando com 1 da esquerda e começando com -1 da direita)

separador

Um único caractere para converter em token o valor da variável. O padrão é o caractere barra (/) nas plataformas AIX and Linux ou o caractere barra invertida (\) nas plataformas Windows, mas o separador pode ser qualquer caractere válido que possa aparecer no valor da variável.

Se a palavra-chave do separador for especificado em um nome de variável, o valor da variável será dividido em tokens de acordo com o caractere separador.

O valor designado à palavra-chave do token é usado como um índice para selecionar qual token será usado para substituir o nome da variável. O índice de token é relativo ao primeiro caractere na variável e começa em 1. Se a palavra-chave do token não for especificada, a variável inteira será inserida.

Os valores substituídos em um nome do agente no XML da mensagem são tratados sem distinção entre maiúsculas e minúsculas. Todos os nomes do Managed File Transfer Agent são em maiúsculas. Se o valor Paris for substituído em um atributo de agente no XML da mensagem, esse valor será interpretado como uma referência ao agente PARIS.

Conceitos relacionados

[“Exemplos: substituição de variável para definições de monitor de recurso” na página 246](#)

[Exemplos de substituição de variável para definições de monitor de recurso usando XML e IBM MQ Explorer.](#)

Tarefas relacionadas

[O que fazer se a substituição de variável causar a ida de vários arquivos para um único nome de arquivo](#)

Exemplos: substituição de variável para definições de monitor de recurso

[Exemplos de substituição de variável para definições de monitor de recurso usando XML e IBM MQ Explorer.](#)

Exemplos mostrando como funciona a substituição de variável

Supondo que o caminho de arquivo para o arquivo acionador correspondente seja `c:\MONITOR\REPORTS\Paris\Report2009.doc` nas plataformas Windows e `/MONITOR/REPORTS/Paris/Report2009.doc` nas AIX and Linux, as variáveis serão substituídas conforme mostrado na tabela a seguir:

Tabela 13. Como as variáveis são substituídas

Especificação da variável	Após a substituição da variável
<code>\${FilePath}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux : /MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=1}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=2}{separator=.}}</code>	Windows: doc AIX and Linux : doc
<code>\${FilePath{token=3}}</code>	Windows : RELATÓRIOS AIX and Linux : Paris

É possível também especificar um índice de token negativo para selecionar tokens relativos ao último caractere da variável, conforme mostrado na tabela a seguir. Os exemplos na tabela usam o mesmo valor de variável, c:\MONITOR\REPORTS\Paris\Report2009.doc on Windows e /MONITOR/REPORTS/Paris/Report2009.doc on AIX and Linux

Tabela 14. Exemplos de como usar um índice de token negativo

Especificação da variável	Após a substituição da variável
<code>\${FilePath}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux : /MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=-2}{separator=\}}</code>	Windows : Paris AIX and Linux : Paris
<code>\${FilePath{token=-4}}</code>	Windows : MONITOR AIX and Linux : MONITOR

As variáveis usadas para substituição estão disponíveis apenas para as seguintes condições acionadoras positivas e a opção `noSizeChange`, que é uma exceção à regra de condição acionadora positiva:

- `match`
- `fileSize`
- `noSizeChange`

Se uma condição `noMatch` for usada e houver nomes de variáveis de substituição na definição da tarefa, a tarefa não será chamada e o monitor gerará um código de retorno de 110 e uma mensagem de erro BFGDM0060E.

Exemplo usando XML

O exemplo de XML de definição de tarefa a seguir usa o nome do agente do monitor como agente de origem para a transferência (Paris), usa o penúltimo nome de diretório no caminho de arquivo como nome do agente de destino para a transferência (Report2009) e renomeia o arquivo transferido para ser a raiz do nome do arquivo acionador com uma extensão .rpt.

```
<?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="{AgentName}" QMgr="QM1" />
    <destinationAgent agent="{FilePath{token=-2}}" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${FileName{token=1}}{separator=}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Isso resulta na transformação da tarefa XML to:

```
<?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="Paris" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/Report2009.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

A variável `{FilePath{token=-2}}` no atributo `agent` do elemento `<destinationAgent>` é substituída pelo valor `Paris`. Esse valor é tratado sem distinção entre maiúsculas e minúsculas e interpretado como uma referência ao agente `PARIS`.

Exemplos usando IBM MQ Explorer

Ao criar um monitor de recurso por meio do IBM MQ Explorer e quando as propriedades do monitor e as condições acionadoras tiverem sido especificadas, será fornecida a opção para incluir itens de transferência no monitor. Os exemplos a seguir demonstram como as variáveis `{FilePath}` e `{FileName}` podem ser usadas no **"painel Incluir um item de transferência"** para customizar as transferências resultantes de uma correspondência do monitor de recurso.

Exemplo 1

Para apenas transferir o arquivo de origem para outro local quando uma condição acionadora for atendida, a variável `{FilePath}` poderá ser usada:

- Configure o **Nome do arquivo** de origem para ser `${FilePath}`.
- No menu suspenso **Tipo** para o destino, selecione **Diretório**.
- Configure o destino **Nome do arquivo** para ser o local para o qual você deseja que o arquivo de origem seja transferido, por exemplo, isso pode ser `C:\MFT\out\`.

Exemplo 2

Para transferir o arquivo de origem para outro local e mudar a extensão dele, a variável `${FileName}` pode ser usada em conjunto com a variável `${FilePath}`:

No exemplo a seguir, assume-se que o caminho do arquivo de origem é igual a `C:\MONITOR\REPORTS\Paris\Report2009.doc`:

- Configure o **Nome do arquivo** de origem para ser `${FilePath}`.
- Configure o destino **Nome do arquivo** para ser o local para o qual você deseja que o arquivo de origem seja transferido, seguido por `${FileName}{token=1}{separator=.}`, seguido pela nova extensão do arquivo. Por exemplo, isso poderia ser `C:\MFT\out\${FileName}{token=1}{separator=.}.rpt`, o que equivaleria a `C:\MFT\out\Report2009.rpt` com o nome do arquivo de origem.

Exemplo 3

Para usar parte do caminho do arquivo de origem para determinar o destino da transferência, a variável `${FilePath}` pode ser usada em conjunto com as especificações de token e separador.

No exemplo a seguir, assume-se que o caminho do arquivo de origem seja igual a `C:\MONITOR\REPORTS\Paris\Report2009.doc`.

É possível usar parte do caminho de arquivo de origem para determinar o destino do arquivo. Usando o exemplo de caminho de arquivo de `C:\MONITOR\REPORTS\Paris\Report2009.doc`, se o arquivo fosse transferido para uma pasta dependendo da localização do arquivo de origem, ou seja, `Paris` neste exemplo, o seguinte poderia ser feito:

- Configure o **Nome do arquivo** de origem para ser `${FilePath}`.
- Configure o destino **Nome do arquivo** para ser o destino das pastas de cada local e, em seguida, anexe a parte do destino do caminho de arquivo e o nome do arquivo. Por exemplo, isso poderia ser `C:\MFT\out\${FilePath}{token=-2}{separator=}\/\${FileName}`, o que equivaleria a `C:\MFT\out\Paris\Report2009.doc` com o nome do arquivo de origem.

Conceitos relacionados

[“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.”](#) na página 244

Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.

Tarefas relacionadas

[O que fazer se a substituição de variável causar a ida de vários arquivos para um único nome de arquivo](#)

Monitorando uma Fila e Usando Substituição de Variável

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

Sobre esta tarefa

Nesse exemplo, o agente de origem é chamado de AGENT_VENUS, que se conecta ao QM_VENUS. A fila que o AGENT_VENUS monitora é chamada de START_QUEUE e está localizada no QM_VENUS. O agente sonda a fila a cada 30 minutos.

Quando um grupo completo de mensagens é gravado na fila, o monitor de tarefa envia o grupo de mensagens para um arquivo em um número de agentes de destinos, todos se conectando ao gerenciador de filas QM_MARS. O nome do arquivo para o qual o grupo de mensagens é transferido é definido pela IBM MQ propriedade de mensagens `usr.fileName` na primeira mensagem no grupo. O nome do agente para o qual o grupo de mensagens é enviado é definido pela IBM MQ propriedade de mensagem `usr.toAgent` na primeira mensagem no grupo. Se o cabeçalho `usr.toAgent` não estiver configurado, o valor padrão a ser usado para o agente de destino será AGENT_MAGENTA.

Ao especificar `useGroups="true"`, se você também não especificar `groupId="{GROUPID}"`, a transferência obterá somente a primeira mensagem na fila. Por exemplo, se estiver usando a substituição de variável para gerar o `fileName`, será possível que o conteúdo de `a.txt` não esteja correto. Isto ocorre porque `fileName` é gerado pelo monitor, mas a transferência realmente obtém uma mensagem que não é a que deve gerar o arquivo chamado `fileName`.

Procedimento

1. Crie o XML de tarefa que define a tarefa que o monitor executa quando for acionado.

```
<?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="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="{toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/{fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

As variáveis que são substituídas com os valores dos cabeçalhos da mensagem do IBM MQ são destacados em **negrito**. Esse XML de tarefa é salvo no arquivo `/home/USER1/task.xml`

2. Crie um monitor de recursos para monitorar a fila START_QUEUE.

Execute o seguinte comando:

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. Um usuário ou programa grava um grupo de mensagens na fila START_QUEUE.

A primeira mensagem nesse grupo tem o conjunto de propriedades de mensagens do IBM MQ a seguir:

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. O monitor é acionado quando o grupo completo é gravado. O agente substitui as propriedades de mensagem do IBM MQ no XML de tarefa.

Isso resulta na transformação da tarefa XML to:

```
<?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="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Resultados

A transferência definida pela tarefa XML é executada. O grupo completo de mensagens que são lidos a partir da START_QUEUE por AGENT_VENUS é gravado em um arquivo chamado /reports/larmer.rpt no sistema em que o AGENT_VIOLET está em execução.

Como proceder a seguir

Transferindo cada mensagem para um arquivo separado

Se desejar monitorar uma fila e fazer com que cada mensagem seja transferida para um arquivo separado, será possível usar uma técnica semelhante àquela descrita anteriormente neste tópico.

1. Crie o monitor conforme descrito anteriormente, especificando o parâmetro **-tr completeGroups** no comando **fteCreateMonitor ..**
2. No XML da tarefa, especifique o seguinte:

```
<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
```

No entanto, ao colocar as mensagens na fila de origem, não as coloque em um grupo do IBM MQ. Inclua propriedades de mensagem do IBM MQ em cada mensagem. Por exemplo, especifique a propriedade `usr.filename` com um valor de nome do arquivo exclusivo para cada mensagem. Isso efetivamente faz com que o Managed File Transfer Agent trate cada mensagem na fila de origem como um grupo separado.

Conceitos relacionados

[“Transferindo Dados de Mensagens para Arquivos” na página 283](#)

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

[“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.” na página 244](#)

Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando

o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.

O que Fazer se os Arquivos de Destino Criados por uma Transferência Iniciada por um Monitor de Recurso de Fila Contiverem Dados Incorretos

Tarefas relacionadas

“Configurando tarefas de monitor do MFT para iniciar comandos e scripts” na página 237

Monitores de recursos não estão limitados a executar transferências de arquivo como suas tarefas associadas. É também possível configurar o monitor para chamar outros comandos a partir do agente de monitoramento, incluindo programas executáveis, scripts Ant ou tarefas JCL. Para chamar comandos, edite o XML de definição de tarefa de monitoramento para incluir um ou mais elementos de comando nos parâmetros de chamada de comando correspondentes, como argumentos e propriedades.

“Exemplo: configurando um recurso do MFT” na página 243

É possível especificar uma fila do IBM MQ como o recurso a ser monitorado por um monitor de recurso usando o parâmetro **-mq** com o comando **fteCreateMonitor**.

Referências relacionadas

fteCreateMonitor: criar um monitor de recurso do MFT

Propriedades de mensagem do MQ lidas pelo MFT de mensagens em filas de origem

Configurando o comportamento de repetição do monitor para transferências de mensagem para arquivo

Se uma transferência de mensagem para arquivo acionada por um monitor de recurso falhar e deixar o grupo de mensagens que acionou o monitor na fila, essa transferência será reenviada em intervalos de pesquisa subsequentes. O número de vezes que a transferência é reenviada é limitado pela propriedade **monitorGroupRetryLimit** do agente de monitoramento.

Sobre esta tarefa

Sempre que uma nova transferência de mensagem para arquivo for acionada, um novo ID de transferência será gerado para a tarefa de transferência.

Se o agente for reiniciado, o monitor acionará uma transferência novamente, mesmo que o número de vezes que a transferência tenha sido acionada tenha excedido o valor de **monitorGroupRetryLimit** no arquivo agent.properties. O valor da propriedade **monitorGroupRetryLimit** é o número máximo de vezes que um monitor acionará uma transferência de mensagem para arquivo novamente se o grupo de mensagens ainda existir na fila. O valor padrão desta propriedade é 10. O valor desta propriedade pode ser configurado como qualquer valor de número inteiro positivo ou -1. Se o valor -1 for especificado para esta propriedade, o monitor acionará a transferência novamente um número ilimitado de vezes, até que a condição acionadora não seja atendida.

Se uma tentativa de transferência fizer com que o número de vezes que a transferência foi acionada exceda o valor de **monitorGroupRetryLimit**, o agente gravará um erro em seu log de eventos.

Uma única mensagem é tratada como se fosse um único grupo e a transferência é acionada novamente em cada intervalo de pesquisa enquanto a mensagem permanecer na fila e enquanto o número de vezes que a transferência foi acionada for menor que o valor de **monitorGroupRetryLimit**.

Para configurar a propriedade **monitorGroupRetryLimit** no agente de monitoramento, execute as seguintes etapas:

Procedimento

1. Pare o agente de monitoramento usando o comando **fteStopAgent**.
2. Edite o arquivo agent.properties para o agente de monitoramento para incluir a seguinte linha:

```
monitorGroupRetryLimit=number_of_retries
```

O arquivo `agent.properties` está localizado no diretório `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_name`.

3. Inicie o agente de monitoramento usando o comando **`fteStartAgent`**.

Tarefas relacionadas

“Exemplo: configurando um recurso do MFT” na página 243

É possível especificar uma fila do IBM MQ como o recurso a ser monitorado por um monitor de recurso usando o parâmetro `-mq` com o comando **`fteCreateMonitor`**.

Usando um arquivo acionador

É possível usar o conteúdo de um arquivo acionador em um monitor de recurso para definir um conjunto de arquivos para transferência em uma única solicitação de transferência. Sempre que um arquivo acionador correspondente é detectado, seu conteúdo é analisado para caminhos do arquivo de origem e, opcionalmente, para caminhos do arquivo de destino. Estes caminhos do arquivo são então usados para definir itens do arquivo no arquivo XML de transferência de tarefa especificado, que é enviado como uma única solicitação de transferência para o agente. A definição do monitor de recurso determina se o conteúdo do acionador é ativado.

É possível ativar o acionamento de conteúdo do arquivo ao criar um monitor, especificando o parâmetro `-tc` (conteúdo do acionador). Este parâmetro `-tc` se aplica apenas às opções do acionador do arquivo `match` e `noSizeChange`. Para obter mais informações sobre como criar um monitor, consulte **`fteCreateMonitor`**: criar um monitor de recurso MFT.

Ao usar um arquivo de conteúdo do acionador, o formato padrão de cada linha é:

- Um único caminho de arquivo de origem ou
- Um caminho de arquivo de origem e um caminho de arquivo de destino, separados por uma vírgula

em que os caracteres de espaço em branco são tratados como parte dos caminhos de arquivo. É possível mudar o formato de linha padrão especificando os parâmetros `-tcr` e `-tcc` no comando **`fteCreateMonitor`**. Para obter mais informações, consulte “Opções Avançadas” na página 254.

Após a análise de um arquivo acionador, uma lista de caminhos do arquivo é gerada e aplicada ao XML de tarefa de transferência especificado. Assim como todos os monitores, o formato do XML de tarefa de transferência é um XML de tarefa de transferência completo gerado pelo comando **`fteCreateTransfer`** com um único item ou arquivo definido. O único item deve usar as variáveis de substituição `${contentSource}` e, opcionalmente, `${contentDestination}`, como substituições para os caminhos do arquivo de origem e de destino. O monitor expande o XML de tarefa de transferência para incluir um item de arquivo para cada linha (caminho do arquivo) no arquivo acionador.

Não é possível usar o acionamento de conteúdo do arquivo com o parâmetro `-bs` porque o parâmetro `-tc` implica em uma solicitação de transferência para cada arquivo acionador.

exemplo

O exemplo a seguir define um monitor para acionar em um arquivo que termina em `trig` e lê os caminhos de arquivo nesse arquivo.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*trig"
-tc
```

O comando **`fteCreateTransfer`** cria um arquivo que é chamado `task.xml` para um único arquivo com um caminho de arquivo de origem de `${contentSource}`. Por exemplo:

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
  </source>
</item>
```

O comando **fteCreateMonitor** varre arquivos que terminam em `trig` no diretório `/home/trigdir` e usa o conteúdo para criar uma única solicitação de transferência baseada em `task.xml` para todos os caminhos nesse arquivo acionador. O formato do arquivo acionador deve ser um caminho do arquivo (apenas de origem) em cada linha, sem separador de vírgula. Por exemplo:

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

Todos os arquivos são entregues para o diretório `/file/destdir` com seu nome de arquivo e não seu caminho de arquivo, ou seja, `/home/file/first.txt` é entregue a `/file/destdir/first.txt`.

Como alternativa, se você mudar o parâmetro **-dd /file/destdir** no comando **fteCreateTransfer** para `-df ${contentDestination}` e o formato do conteúdo de um arquivo acionador para *caminho do arquivo de origem,caminho do arquivo de destino*, será possível definir diferentes caminhos de destino para o mesmo agente de destino.. Por exemplo:

```
/home/file/first.txt,/home/other/sixth.txt
```

O local de destino então se torna `/home/other/sixth.txt`.

As variáveis de substituição podem ser convertidas em token. Por exemplo, é possível separar a parte do nome do arquivo do caminho fornecido usando `${contentDestination{token=-1}}`.

Portanto, se o destino **fteCreateTransfer** for definido como `-df /file/destdir/${contentDestination{token=-1}}`, o novo destino para `/home/file/first.txt` será `/file/destdir/sixth.txt`.

Opções Avançadas

É possível alterar o formato de linha padrão para o conteúdo do arquivo acionador usando o parâmetro **-tcr regex**. Forneça uma expressão regular que corresponda ao formato de linha necessário e forneça um ou dois grupos de captura. O primeiro grupo de captura é a origem e o segundo grupo de captura, opcional, é o destino. Por exemplo:

- Os caminhos de origem e destino são separados por um hífen:

```
((?:[^-]+)-((?:[^-]+)+)
```

Neste exemplo, o separador é definido em três locais e todas as três instâncias do hífen, `-`, podem ser alteradas para qualquer caractere. Certifique-se de escapar todos os caracteres especiais.

- Os caminhos de origem e destino são separados por uma vírgula com espaços à direita. Os comentários indicados por um sinal de número (`#`) são ignorados.

```
((?:[^\, ]+),((?:[^\, ]+)+) *(?:[#. ]+)
```

Os caminhos de arquivo não podem conter o sinal de número (`#`) Geralmente uma entrada é a seguinte: `/home/source/from.txt,/home/destination/to.txt # some comment`.

Se você usar o parâmetro **-tcr**, certifique-se de que a expressão regular também seja projetada e testada para que a expressão possa detectar erros e analisar corretamente os arquivos acionadores.

É possível reverter a ordem da captura usando o parâmetro **-tcc destSrc**. Se você especificar esse parâmetro, o primeiro grupo de captura será o caminho do arquivo de destino e o segundo grupo será o caminho do arquivo de origem.

Como os Erros São Manipulados

Arquivo acionador vazio

Se o arquivo acionador estiver vazio, o resultado será nenhuma transferência de arquivos. Ou seja, o monitor cria uma solicitação de transferência, mas nenhum item de arquivo é especificado.

Arquivo acionador com erros

Se uma entrada em um arquivo acionador falhar ao analisar no formato esperado, nenhuma solicitação de transferência será gerada. Um log de erro de monitor é publicado e o erro também é registrado no log de eventos. O arquivo acionador é marcado como processado e o monitor não tentará processar o arquivo novamente, até que o arquivo tenha sido atualizado.

XML de tarefa de transferência incompatível

O XML da tarefa de transferência deve corresponder ao arquivo acionador, ou seja, se o XML da tarefa de transferência tiver `{contentSource}` e `{contentDestination}`, todos os arquivos acionadores para esse monitor devem ter caminhos de arquivo de origem e destino e da mesma forma para o inverso. No primeiro caso, o monitor relata uma falha de substituição do `{contentDestination}` se o arquivo acionador fornece apenas o caminho do arquivo de origem.

Exemplos

O exemplo a seguir é um acionador de conteúdo básico no qual o conteúdo de um arquivo acionador tem apenas um caminho do arquivo de origem:

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir {contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*trig"
-tc
```

O parâmetro **-tcr** define dois grupos de captura de uma sequência de quaisquer caracteres que são separados por um caractere de espaço. O parâmetro **-tcc destSrc** e a opção indicam que os grupos de captura devem ser processados como destino, em seguida, origem.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -df {contentDestination} $
{contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*trig"
-tc
-tcr "((?:[^\ ])+) ((?:[^\ ])+)" -tcc destSrc
```

Conceitos relacionados

[“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.”](#) na página 244

Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.

Tarefas relacionadas

[“Monitorando uma Fila e Usando Substituição de Variável”](#) na página 249

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

Referências relacionadas

fteCreateMonitor: criar um monitor de recurso do MFT

fteCreateTransfer: iniciar uma nova transferência de arquivos

Criando logs de monitores de recursos do MFT

É possível obter informações de diagnóstico sobre monitores de recurso usando a criação de log.

Sobre esta tarefa

É possível usar a criação de log para monitores de recurso usando o comando `fteSetAgentLogLevel` ou o arquivo `agent.properties` para controlar a criação de log do monitor de recurso.

Observe que os pontos de rastreamento existentes ainda são usados para capturar informações.

Os logs do monitor de recursos são gravados em um arquivo denominado `resmoneventN.log`, em que *N* representa um número; por exemplo, `resmonevent0.log`. Os arquivos de log de eventos registram várias ações que ocorrem quando um monitor pesquisa um recurso, por exemplo, um diretório ou uma fila.





Atenção: Todos os monitores de recurso de um agente gravam no mesmo arquivo de log.

Para obter alguns exemplos de saída de um arquivo `resmoneventN.log`, consulte [O que fazer se o monitor de recursos de diretório MFT não estiver acionando arquivos](#).

A tabela a seguir lista o tipo de eventos que o monitor de recurso grava no arquivo de log. A terceira coluna descreve o nível de log necessário para capturar cada evento no qual o nível mais baixo é INFO e o mais alto é VERBOSE.

Observe que configurar um nível de log mais alto também grava eventos de nível mais baixos. Por exemplo, configurar o nível de log como MODERATE também grava eventos de nível INFO, mas não grava eventos de nível VERBOSE.

Número	Evento	Nível de log	Descrição
1	Monitor criado	INFORMAÇÃO	Um monitor de recurso foi criado.
2	Monitor excluído	INFORMAÇÃO	Um monitor de recurso foi excluído.
3	Monitor interrompido	INFORMAÇÃO	Um monitor de recurso foi interrompido.
4	Monitor iniciado	INFORMAÇÃO	Um monitor de recurso foi iniciado.
5	Pesquisa inicial	INFORMAÇÃO	Um monitor de recurso iniciou um novo ciclo de pesquisa.
6	Pesquisa final	INFORMAÇÃO	Um ciclo de pesquisa de monitor de recurso foi finalizado.
7	Correspondência de padrão	VERBOSE	Um arquivo no diretório de monitor acionador ou uma mensagem em uma fila que corresponde ao padrão especificado foi localizada.
8	Incompatibilidade de padrão	VERBOSE	Um arquivo sem correspondência no diretório de monitor acionador ou uma mensagem em uma fila que não corresponde ao padrão especificado foi localizada.
9	Solicitação de transferência	INFORMAÇÃO	Uma transferência foi iniciada pelo monitor de recurso.
10	Diretório muito profundo	VERBOSE	O diretório monitorado pelo monitor de recurso contém mais sub-diretórios para pesquisar que o número especificado na configuração do monitor de recurso.

Número	Evento	Nível de log	Descrição
11	Arquivo bloqueado	MODERATE	O arquivo acionador monitorado pelo monitor de recurso está bloqueado por outro processo.
12	Tamanho do arquivo pequeno	MODERATE	O arquivo acionador é menor que o tamanho especificado na configuração do monitor de recurso.
13	Tamanho do arquivo instável	MODERATE	O arquivo acionador está sendo mudado mais frequentemente do que o esperado pela configuração do monitor de recurso.
14	Número excessivo de pesquisas	MODERATE	Um monitor de recurso pesquisou um arquivo acionador instável um grande número de vezes.
15	Itens correspondentes	INFORMAÇÃO	Número total de arquivos acionadores localizados no diretório pesquisado por um monitor de recurso.
16	Itens de transferência	INFORMAÇÃO	Número total de itens na solicitação de transferência.
17	FDC gerado	MODERATE	Um monitor de recurso gerou uma exceção.
18	Solicitação de transferência	INFORMAÇÃO	Solicitação de transferência enviada por monitor de recurso.
19	Início de monitor com falha	MODERATE	Um monitor de recurso falhou ao iniciar.
20	Histórico limpo	INFORMAÇÃO	As informações do histórico do monitor foram excluídas.
21	Falha ao limpar histórico do monitor	INFORMAÇÃO	A tentativa de limpar as informações de histórico do monitor falhou.
22	ID da transferência	INFORMAÇÃO	O ID da solicitação de transferência foi enviado pelo monitor.
23	Envio em lote	INFORMAÇÃO	Número total de solicitações de transferência para itens correspondentes: N , em que N é um número.
 24	Conectado	VERBOSE	Um monitor de recurso foi conectado ao gerenciador de filas do agente.
 25	Desconectado	VERBOSE	Um monitor de recurso foi desconectado do gerenciador de filas do agente.
 26	Erro durante desconexão	VERBOSE	Um monitor de recurso encontrou um problema ao desconectar do gerenciador de filas do agente.

Procedimento

- Para usar o comando **fteSetAgentLogLevel** para ativar e desativar a criação de log do monitor de recurso, consulte [fteSetAgentLogNível](#) para obter uma descrição do parâmetro **logMonitor** e exemplos de como usar as diferentes opções..
- Para usar o arquivo `agent.properties` para controlar o registro do monitor de recursos, consulte [O arquivo MFT agent.properties](#) para obter uma descrição das propriedades adicionais que permitem realizar as seguintes atividades de registro:

- Ative ou desligue a criação de log
- Limite o tamanho de cada arquivo de log
- Limite o número de logs que os monitores de recurso podem gerar

Exemplo

A mensagem de amostra a seguir configura o registro de nível verbose para o agente HA2, no gerenciador de filas MFTDEMO:

```
<?xml version="1.0"?>
<log:log version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:log="https://www.ibm.com/log">
  <log:originator>
    <log:request>
      <log:hostName>192.168.7.1</log:hostName>
      <log:userID>johndoe</log:userID>
    </log:request>
  </log:originator>
  <log:endpoint agent="HA2" QMgr="MFTDEMO"/>
  <log:logMonitor>MON1="verbose"</log:logMonitor>
</log:log>
```

Referências relacionadas

[Comando fteSetAgentLogLevel](#)

[O arquivo MFT agent.properties](#)

Iniciando um monitor de recurso do MFT

No IBM MQ 9.3.0, é possível iniciar monitores de recursos sem precisar parar ou reiniciar um agente usando o comando **fteStartMonitor**.

Antes de começar

Se o gerenciamento de autoridade do usuário tiver sido ativado configurando o atributo **authorityChecking** como true no arquivo [agent.properties](#), deve-se ter a autoridade Monitorar ou Monitorar operações para iniciar um monitor de recurso. Para obter mais informações sobre gerenciamento de autoridade do usuário, consulte [Restringindo autoridades do usuário nas ações do agente do MFT](#).

Sobre esta tarefa

É possível executar o comando **fteStartMonitor** em qualquer sistema no qual o componente de comandos do Managed File Transfer esteja instalado, o que significa que é possível iniciar um monitor de recurso de qualquer lugar, e não se restringe ao sistema no qual o agente que possui o monitor de recurso está em execução. Para obter informações sobre os parâmetros necessários e opcionais para esse comando, consulte [fteStartMonitor \(iniciar um monitor de recurso do MFT\)](#).

Procedimento

- Para descobrir o estado de um agente antes ou depois de executar o comando **fteStartMonitor**, use o comando **fteListMonitors** com o parâmetro **-v**, conforme mostrado no exemplo a seguir:

```
fteListMonitors -ma monitoring_agent_name -v
```

- Para iniciar um monitor de recurso em um agente em execução na mesma máquina, insira o comando **fteStartMonitor** da seguinte forma:

```
fteStartMonitor -mn monitor_name -ma agent_name
```

- Para iniciar um monitor de recurso em um agente em execução em uma máquina diferente, insira o comando **fteStartMonitor** da seguinte forma:

```
fteStartMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Se o gerenciador de filas de comando também for o gerenciador de filas do agente para o agente de monitoramento, o parâmetro **-mm** será opcional, caso contrário, deve-se especificar o gerenciador de filas do agente com o parâmetro **-mm**.

Resultados

Se o agente estiver em execução, o monitor de recurso será iniciado se ele estiver atualmente interrompido. O comando gera as mensagens a seguir e registra um evento no `output0.log` do agente:

```
BFGCL0816I: uma solicitação para iniciar o monitor de recurso 'monitor_name' do agente 'agent_name' foi emitida.  
BFGCL0251I: O pedido foi concluído com êxito.
```

Para obter informações sobre as mensagens que o comando envia se não pode iniciar o monitor de recurso, consulte [fteStartMonitor \(iniciar um monitor de recurso do MFT\)](#).

Conceitos relacionados

[“Conceitos de monitoramento de recurso do MFT” na página 232](#)

Uma visão geral dos principais conceitos do recurso de monitoramento de recursos do Managed File Transfer.

Tarefas relacionadas

[“Parando um monitor de recurso do MFT” na página 259](#)

No IBM MQ 9.3.0, é possível parar monitores de recursos sem precisar parar ou reiniciar um agente, usando o comando **fteStopMonitor**.

Referências relacionadas

[fteStartMonitor \(iniciar um monitor de recurso do MFT\)](#)

Parando um monitor de recurso do MFT

No IBM MQ 9.3.0, é possível parar monitores de recursos sem precisar parar ou reiniciar um agente, usando o comando **fteStopMonitor**.

Antes de começar

Se o gerenciamento de autoridade do usuário tiver sido ativado configurando o atributo **authorityChecking** como `true` no arquivo `agent.properties`, deve-se ter autoridade `Monitorar` ou `Monitorar` operações para parar um monitor de recurso. Para obter mais informações sobre gerenciamento de autoridade do usuário, consulte [Restringindo autoridades do usuário nas ações do agente do MFT](#).

Sobre esta tarefa

É possível executar o comando **fteStopMonitor** em qualquer sistema no qual o componente de comandos do Managed File Transfer esteja instalado, o que significa que é possível parar um monitor de recurso de qualquer lugar, e não se restringe ao sistema no qual o agente que possui o monitor de recurso está em execução. Para obter informações sobre os parâmetros necessários e opcionais para esse comando, consulte [fteStopMonitor \(parar um monitor de recurso do MFT\)](#).

Quando um monitor de recursos é interrompido, ele grava uma mensagem para o log de eventos do monitor de recursos do agente, `resmoneventnumber.log`. Se o monitor de recurso for interrompido com o comando **fteStopMonitor**, a mensagem incluirá o nome do usuário que emitiu a solicitação de parada:

```
Monitor de recursos parado pelo usuário '<mquser_id>'
```

Um monitor de recurso é iniciado automaticamente se o seu agente é reiniciado, mesmo se o monitor de recurso foi interrompido anteriormente usando o comando **fteStopMonitor**.

Os agentes processam as solicitações de parada do monitor de forma seriada em vez de em paralelo, portanto, por exemplo, se um agente recebe uma solicitação para parar o monitor M1 e, em seguida, outra solicitação para parar o monitor M2 em rápida sucessão, ele para o M1 primeiro antes de tentar parar o M2.

Procedimento

- Para descobrir o estado de um agente antes ou depois de executar o comando **fteStopMonitor**, use o comando **fteListMonitors** com o parâmetro **-v**, conforme mostrado no exemplo a seguir:

```
fteListMonitors -ma monitoring_agent_name -v
```

- Para parar um monitor de recurso em um agente em execução na mesma máquina, insira o comando **fteStopMonitor** da seguinte forma:

```
fteStopMonitor -mn monitor_name -ma agent_name
```

- Para parar um monitor de recurso em um agente em execução em uma máquina diferente, insira o comando **fteStopMonitor** da seguinte forma:

```
fteStopMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Se o gerenciador de filas de comando também for o gerenciador de filas do agente para o agente de monitoramento, o parâmetro **-mm** será opcional, caso contrário, deve-se especificar o gerenciador de filas do agente com o parâmetro **-mm**.

Resultados

Se o agente estiver em execução, o monitor de recurso será interrompido se ele foi iniciado atualmente. O comando gera as mensagens a seguir e registra um evento no output0.log do agente:

```
BFGCL0813I: uma solicitação para parar o monitor de recurso 'MNTR' do agente 'SOURCE' foi emitida.  
BFGCL0251I: O pedido foi concluído com êxito.
```

Para obter informações sobre as mensagens que o comando envia se ele não pode parar o monitor de recurso, consulte [fteStopMonitor \(parar um monitor de recurso do MFT\)](#)

Conceitos relacionados

“Conceitos de monitoramento de recurso do MFT” na página 232

Uma visão geral dos principais conceitos do recurso de monitoramento de recursos do Managed File Transfer.

Tarefas relacionadas

“Iniciando um monitor de recurso do MFT” na página 258

No IBM MQ 9.3.0, é possível iniciar monitores de recursos sem precisar parar ou reiniciar um agente usando o comando **fteStartMonitor**.

Referências relacionadas

[fteStopMonitor \(parar um monitor de recurso do MFT\)](#)

Fazendo backup e restaurando os monitores de recursos do MFT

É possível fazer backup dos monitores de recursos que você deseja ter disponíveis para uso futuro exportando suas definições para um arquivo XML que depois possa ser importado para criar um novo monitor de recurso com base no backup.

Sobre esta tarefa

Você pode precisar fazer backup dos monitores de recursos que tenha definido anteriormente para que possa reutilizar suas definições no futuro, por exemplo, para recriar os monitores de recursos em uma infraestrutura diferente ou se um monitor de recurso precisar ser recriado devido a problemas de gerenciador de filas.

É possível fazer backup de uma única definição de gerenciador de recursos usando o comando **fteCreateMonitor** ou **fteListMonitors** com o parâmetro **-ox**. Em ambos os casos, o backup da definição do gerenciador de recursos é feito exportando-a para um arquivo XML. Você pode então usar o parâmetro **-ix** do comando **fteCreateMonitor** para criar um novo gerenciador de recursos importando a definição do arquivo XML.

Com o parâmetro **-ox**, é possível fazer backup apenas de uma definição de monitor de recurso por vez.

O parâmetro **-od** está incluído no comando **fteListMonitors**. Ao especificar esse parâmetro, é possível fazer backup de mais de um monitor de recurso ao mesmo tempo exportando suas definições em massa para um diretório especificado. Cada definição do monitor de recurso é salva em um arquivo XML separado, com um nome no formato *agent name.monitor name.xml*.

O parâmetro **-od** é particularmente útil se você tem um grande número de monitores de recursos que deseja fazer backup porque você precisa executar o comando **fteListMonitors -od** apenas uma vez, em vez de ter de executar o comando **fteListMonitors -ox** separadamente para cada definição de recurso, ou usar um script separado para executar o comando **fteListMonitors -ox** para cada monitor de recurso.

Procedimento

- Para fazer backup da definição de um monitor de recurso exportando-o para um arquivo XML, use um dos comandos a seguir:
 - O comando **fteCreateMonitor** com o parâmetro **-ox**.
 - O comando **fteListMonitors** com o parâmetro **-ox**.

Quando você estiver usando o parâmetro **-ox**, também é necessário especificar os parâmetros **-ma** e **-mn**, como mostrado no exemplo a seguir:

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

- Para fazer backup de várias definições de monitor de recurso exportando-as para arquivos XML em um diretório especificado, use o comando **fteListMonitors** com o parâmetro **-od** conforme mostrado no exemplo a seguir:

```
fteListMonitors -od /usr/mft/resmonbackup
```

Deve-se especificar um diretório de destino válido quando você está fazendo backup de monitores de recursos em massa. A não especificação de um caminho de destino resulta em uma mensagem de erro, conforme mostra o exemplo a seguir:

```
BFGCL0762E: Diretório de saída não especificado. Execute novamente o comando especificando um caminho válido.
```

O parâmetro **-od** não deve ser combinado com o parâmetro **-ox**, caso contrário, será exibida a seguinte mensagem de erro:

```
BFGCL0761E: Não é válido especificar os parâmetros '-od' e '-ox' juntos.
```

É possível definir um conjunto particular de monitores de recursos para inclusão no backup. Por exemplo, ao utilizar o parâmetro **-ma** para especificar o nome de um agente, é possível fazer backup de todos os monitores de recursos para aquele agente, como mostrado no exemplo a seguir:

```
fteListMonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

É possível também usar correspondência de curinga, incluindo um caractere de asterisco (*) ao definir um padrão para usar na correspondência de nomes de agente e/ou nomes de monitor. O exemplo a seguir faz backup de todos os monitores de recursos que possuem nomes correspondentes a um padrão especificado e que estão em um agente com um nome correspondente a um padrão especificado:

```
fteListMonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

Enquanto o comando está em execução, ele exibe as seguintes mensagens de relatório de progresso:

Um total de *number* correspondências de definições de monitor de recurso foram encontradas. *index* de *number* definições de monitor de recurso salvas no sistema de arquivos.

Se você estiver usando a opção detalhada, o total em execução ainda será exibido, mas em vez de exibir

index de *number* definições de monitor de recurso salvas no sistema de arquivos,

o comando exibe o nome da definição de monitor que está sendo salva, por exemplo:

BFGCL0762I: Definição do monitor 'FILEMON' do agente 'XFERAGENT' salvo como FILEMON.XFERAGENT.XML para o sistema de arquivos

- Para fazer backup de um monitor de recurso para um agente específico exportando-o para um arquivo XML em um diretório especificado, use o comando **fteListMonitors** com o parâmetro **-od** :

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

Usar o parâmetro **-od** para fazer backup de um único monitor de recursos é semelhante a usar o parâmetro **-ox**, exceto que o nome do arquivo de saída está no formato *agent name.monitor name.xml*.

- Para restaurar as definições do monitor de recurso com base em um backup, use o comando **fteCreateMonitor** com o parâmetro **-ix**, conforme mostrado no exemplo a seguir:

```
fteCreateMonitor -ix file name
```

Para obter mais exemplos de como usar o parâmetro **-od**, consulte [fteListMonitors: listar monitores de recursos MFT](#).

Referências relacionadas

fteCreateMonitor: criar um monitor de recurso do MFT

[fteListMonitors: listar monitores de recurso do MFT](#)

Limpar o histórico do monitor de recurso

É possível limpar o histórico de um monitor de recurso para que outra solicitação de transferência de arquivos possa ser enviada para um arquivo que não foi transferido anteriormente devido a uma falha. Para limpar o histórico do monitor de recurso, é possível usar o comando **fteClearMonitorHistory** ou o IBM MQ Explorer.

Antes de começar

Se o gerenciamento de autoridade do usuário tiver sido ativado ao configurar o atributo **authorityChecking** como **true** no arquivo [agent.properties](#), o usuário que limpa o histórico do monitor deve ter a autoridade apropriada, conforme mostrado na tabela a seguir.

Usuário limpando o histórico do monitor	Autoridade de acesso do MFT	Autoridade requerida
O mesmo usuário que aquele que criou o monitor de recurso.	Monitor	BROWSE on SYSTEM.FTE.AUTHMON1.<monitor_or_agent_name> Essa é a mesma autoridade que a autoridade necessária para criar ou excluir o monitor de recurso.

Tabela 15. Autoridade do usuário necessária para executar o comando **fteClearMonitorHistory** (continuação)

Usuário limpando o histórico do monitor	Autoridade de acesso do MFT	Autoridade requerida
Qualquer usuário que não o usuário que criou o monitor de recurso.	Operações do monitor	SET on SYSTEM.FTE.AUTHPS1.<agent_name> Essa é a mesma autoridade que a autoridade necessária para excluir o monitor de recurso.

Para obter mais informações sobre gerenciamento de autoridade do usuário, consulte [Restringindo autoridades do usuário nas ações do agente do MFT](#).


Se um usuário sem a autoridade necessária tentar limpar o histórico do monitor de recursos, o comando **fteClearMonitorHistory** gerará uma mensagem de erro e registrará a falha no arquivo output0.log do agente. Para obter mais informações, veja [fteClearMonitorHistory: histórico do monitor de recurso limpo](#).

Sobre esta tarefa

Se uma transferência de arquivo tiver sido iniciada e um arquivo não puder ser transferido por algum motivo, o monitor de recurso não selecionará esse arquivo para transferência novamente em sua próxima pesquisa porque o histórico do monitor indica que o arquivo foi visto em uma pesquisa anterior e que ele não foi modificado desde então (consulte [“Conceitos de monitoramento de recurso do MFT”](#) na página 232).

Antes do IBM MQ 9.1.3, se um arquivo falhar ao ser transferido, a transferência de arquivos poderá ser iniciada novamente se o arquivo for excluído e, em seguida, colocado no diretório novamente ou se o arquivo for atualizado para que o atributo de data da última modificação seja mudado ou se o próprio monitor de recurso for recriado.

No entanto, é possível limpar o histórico do monitor de recurso usando o comando **fteClearMonitorHistory** ou usando o IBM MQ Explorer. Limpar o histórico permite que outra solicitação de transferência para um arquivo que tenha falhado na transferência seja enviada sem a necessidade de excluir o arquivo e, em seguida, colocar no diretório novamente ou atualizar o arquivo para mudar seu último atributo de data modificado, o que é útil, por exemplo, em situações em que é necessário transferir o arquivo, mas modificar o arquivo não é possível. Estar apto para limpar o histórico de um monitor de recurso também significa que não há necessidade de recriar o monitor de recurso para enviar outra solicitação de transferência para um arquivo que falhou na transferência.

 O membro SCSQFCMD de amostra enviado com Managed File Transfer on z/OS inclui um script JCL para limpar o histórico de um monitor

Procedimento

- Para usar o comando **fteClearMonitorHistory** para limpar o histórico do monitor de recurso, insira o comando no formato a seguir:

```
fteClearMonitorHistory -p <configuration> -ma <agent name> -mn <monitor name> -w 1000
```

Apenas os parâmetros **-ma** e **-mn** são necessários. Todos os outros parâmetros são opcionais. Para obter informações adicionais sobre como utilizar o Comando **fteClearMonitorHistory**, incluindo exemplos, consulte [fteClearMonitorHistory: Limpar histórico do monitor de recurso](#).

Se o histórico for limpo com sucesso, o comando colocará a mensagem a seguir:

BFGCL0780I: uma solicitação para limpar o histórico do monitor de recurso '*monitor name*' do agente '*agent name*' foi emitida.

BFGCL0251I: O pedido foi concluído com êxito.

e registra o sucesso no arquivo `output0.log` do agente

Se a tentativa de limpar o histórico do monitor de recursos falhar, o **fteClearMonitorHistory** gerará uma mensagem de erro e registrará a falha no arquivo `output0.log` do agente

- Para usar a visualização do monitor de recurso no Plug-in IBM MQ ExplorerMFT para limpar o histórico do monitor de recurso, clique com o botão direito do mouse no monitor de recurso e selecione **Limpar Histórico** no menu suspenso.

Se o histórico for limpo com sucesso, a mensagem a seguir será exibida:

BFGUI00171: Histórico do monitor de recurso limpo com êxito.

Se a tentativa de limpar o histórico falhar, uma mensagem de erro será exibida. Por exemplo:

BFGUI0016E Failed to clear history of specified resource monitor - reason 2059

Trabalhando com modelos de transferência de arquivos

É possível usar os modelos de transferência de arquivos para armazenar as configurações de transferência de arquivos comuns para transferências repetidas ou complexas. Crie um modelo de transferência da linha de comandos usando o comando **fteCreateTemplate** ou use o IBM MQ Explorer para criar um modelo de transferência, usando o assistente **Criar novo modelo para transferência de arquivos gerenciada** ou salve um modelo enquanto estiver criando uma transferência de arquivos, marcando a caixa de seleção **Salvar configurações de transferência como um modelo**. A janela **Modelos de Transferência** exibe todos os modelos de transferência criados na rede do Managed File Transfer.

Sobre esta tarefa

Para criar um modelo de transferência a partir da linha de comandos, use o comando `fteCreateTemplate`. Em seguida, quando quiser enviar um modelo de transferência que tenha criado na linha de comandos, clique em **Enviar** no IBM MQ Explorer.

Para visualizar modelos de transferência no IBM MQ Explorer, use as etapas a seguir:

Procedimento

1. Expanda **Transferência de Arquivos Gerenciados** na visualização Navegador. A **Central de Transferências de Arquivos Gerenciados** é exibida na visualização Conteúdo.
2. Todos os gerenciadores de filas de coordenação estão listados na visualização do Navegador. Expanda o nome do gerenciador de filas da coordenação que você utilizou para a transferência planejada. Para alterar o gerenciador de filas de coordenação ao qual você está conectado, clique com o botão direito do mouse no nome do gerenciador de filas de coordenação, para usar a visualização Navegador, e clique em **Conectar**.
3. Clique em **Modelos de Transferência**. A janela **Modelos de Transferência** é exibida na visualização Conteúdo.
4. A janela **Modelos de Transferência** lista os seguintes detalhes sobre as transferências de arquivos:
 - a) **Nome** O nome de seu modelo de transferência de arquivos.
 - b) **Origem** O nome do agente utilizado para transferir o arquivo do sistema de origem.
 - c) **Arquivo de Origem** O nome do arquivo a ser transferido em seu sistema de host.
Expanda as informações do modelo de transferência para visualizar esse campo.
 - d) **Destino** O nome do agente utilizado para receber o arquivo no sistema de destino.
 - e) **Arquivo de Destino** O nome do arquivo depois de ser transferido para o sistema de destino.
Expanda as informações do modelo de transferência para visualizar esse campo.
 - f) **Início Planejado (fuso horário selecionado)** A hora e data em que a transferência de arquivos está planejada para iniciar no fuso horário usado pelo administrador. Para mudar o fuso horário exibido,

clique em **Janela > Preferências > IBM MQ Explorer > Managed File Transfer** e selecione um fuso horário alternativo na lista **Fuso horário**. Clique em **OK**.

- g) **Eventos Acionadores** O tipo de evento que aciona o início da transferência de arquivos. O tipo pode ser um dos seguintes valores: **existe**, **não existe** ou **excede**.

Resultados

Para atualizar o que é exibido na janela **Modelos de Transferência**, clique no botão Atualizar  na barra de ferramentas da visualização Conteúdo.

Para submeter um modelo de transferência e iniciar a transferência definida no modelo; clique com o botão direito do mouse no nome do modelo e clique em **Submeter**.

Para alterar um modelo de transferência, clique com o botão direito do mouse no nome do modelo e clique em **Editar**. Todos os arquivos incluídos no modelo original são relacionados como parte de um grupo de transferência, mesmo se eles não estiverem incluídos como parte de um grupo do modelo original. Se deseja remover um arquivo do modelo, deve selecionar a especificação do arquivo do grupo e clicar em **Remover selecionado**. Se você deseja incluir novas especificações de arquivo no modelo, utilize os campos do painel do modelo e clique no botão **Adicionar ao grupo**. Depois de fazer várias edições, você recebe um aviso para atribuir um novo nome ao modelo editado.

Para criar uma transferência de arquivos a partir de um modelo de transferência, clique com o botão direito do mouse no nome do modelo e clique em **Editar como Nova Transferência**.

Para criar uma cópia duplicata de um modelo de transferência, clique com o botão direito do mouse no nome do modelo e selecione **Duplicar**. O modelo de transferência duplicado é automaticamente salvo com o mesmo nome que o modelo original, anexado com "(copy)".

Para excluir um modelo de transferência, clique com o botão direito do mouse no nome do modelo e clique em **Excluir**.

Tarefas relacionadas

[“Criando um modelo de transferência de arquivos usando o IBM MQ Explorer” na página 265](#)

É possível criar um modelo de transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos. Você pode usar esse modelo para criar novas transferências de arquivos, usando os detalhes dos modelos, ou submetê-los para iniciar a transferência de arquivos.

Referências relacionadas

[fteCreateTemplate](#): criar novo modelo de transferência de arquivos

[fteListTemplates](#)

[fteDeleteTemplates](#)

Criando um modelo de transferência de arquivos usando o IBM MQ Explorer

É possível criar um modelo de transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos. Você pode usar esse modelo para criar novas transferências de arquivos, usando os detalhes dos modelos, ou submetê-los para iniciar a transferência de arquivos.

Sobre esta tarefa

Para criar um modelo de transferência de arquivos a partir da linha de comandos, use o comando [fteCreateTemplate](#).

Para criar um modelo de transferência de arquivos usando o assistente **Criar novo modelo para transferência de arquivos gerenciada** no IBM MQ Explorer, use as etapas a seguir:

Procedimento

1. Na visualização do Navegador, clique em **Transferência de Arquivos Gerenciados**. A **Central de Transferências de Arquivos Gerenciados** é exibida na visualização Conteúdo.

2. Todos os gerenciadores de filas da coordenação são exibidos na visualização Navegador. Expanda o nome do gerenciador de filas da coordenação que você utilizou para a transferência planejada. Para alterar o gerenciador de filas de coordenação ao qual você está conectado, clique com o botão direito do mouse no nome do gerenciador de filas de coordenação, para usar a visualização Navegador, e clique em **Conectar**.
3. Inicie o assistente **Criar Novo Modelo para Transferência de Arquivo Gerenciada** clicando com o botão direito do mouse em **Modelos de Transferência** e selecionando **Novo Modelo**.
4. Siga as instruções nos painéis do assistente. Também é fornecida em cada painel ajuda sensível ao contexto. Para acessar a ajuda contextual no Windows, pressione F1. No Linux, pressione Ctrl+F1 ou Shift+F1.

Se você criou um modelo que contém todos os detalhes de transferência necessários, certifique-se de selecionar a caixa de seleção **Salvar configurações de transferência como um modelo** na página **Resumo de Transferência**, se esta caixa de seleção ainda não estiver selecionada. Também insira um nome para o modelo no campo Nome. Se você criar um modelo que ainda não contenha todos os detalhes de transferência necessários, a caixa de opção **Salvar definições de transferência como um modelo** é automaticamente marcada para você.

Tarefas relacionadas

“Trabalhando com modelos de transferência de arquivos” na página 264

É possível usar os modelos de transferência de arquivos para armazenar as configurações de transferência de arquivos comuns para transferências repetidas ou complexas. Crie um modelo de transferência da linha de comandos usando o comando **fteCreateTemplate** ou use o IBM MQ Explorer para criar um modelo de transferência, usando o assistente **Criar novo modelo para transferência de arquivos gerenciada** ou salve um modelo enquanto estiver criando uma transferência de arquivos, marcando a caixa de seleção **Salvar configurações de transferência como um modelo**. A janela **Modelos de Transferência** exibe todos os modelos de transferência criados na rede do Managed File Transfer.

Referências relacionadas

fteCreateTemplate: [criar novo modelo de transferência de arquivos](#)

[fteListTemplates](#)

[fteDeleteTemplates](#)

Fazendo backup de uma definição de modelo de transferência de arquivos

Os modelos de transferência de arquivos contêm um documento XML que define as especificações de arquivo de origem e de destino da transferência. É possível usar esse arquivo XML como entrada para o Comando **fteCreateTemplate** para recriar um modelo de transferência de arquivos.

Sobre esta tarefa

Para fazer backup do documento XML que contém as especificações de arquivo de origem e de destino de um modelo de transferência, use o [comando fteCreateTransfer](#) ou o IBM MQ Explorer. Para criar um arquivo de backup formatado por XML do modelo de transferência, use as etapas a seguir:

Procedimento

- Método um: use o parâmetro **-gt** em um comando [fteCreateTransfer](#) para gerar uma mensagem XML do modelo de transferência em um novo arquivo.
- Método dois: crie o modelo usando o IBM MQ Explorer.
Quando você chegar à página *Resumo do modelo de transferência*:
 - a) Copie a *visualização XML da mensagem de solicitação*.
 - b) Salve essa mensagem XML do modelo de transferência em um novo arquivo.
- Método três: use o IBM MQ Explorer para fazer backup de modelos existentes.
 - a) Acesse **Transferência de arquivos gerenciados > Nome do gerenciador de filas > Modelos de transferência**.

- b) No painel Transferência, destaque o modelo que precisa ser submetido a backup, clique com o botão direito e selecione **Editar** no menu pop-up.
- c) Clique em **Avançar** até chegar à página *Resumo do modelo de transferência*.
- d) Copie a *visualização XML da mensagem de solicitação*.
- e) Salve essa mensagem XML do modelo de transferência em um novo arquivo.

Resultados

É possível usar o arquivo de mensagem XML do modelo de transferência, criado por um dos métodos acima, como entrada para o comando `fteCreateTemplate`. Consulte o comando **fteCreateTemplate** para obter detalhes de como usar esse comando.

Referências relacionadas

[Comando fteCreateTemplate](#)

[Comando fteListTemplates](#)

Transferindo dados de arquivos para mensagens

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Para obter informações sobre transferências de mensagem para arquivo, consulte [“Transferindo Dados de Mensagens para Arquivos”](#) na página 283.

O agente de destino para uma transferência de arquivo para mensagem não pode ser um agente de ponte de protocolo ou um agente de ponte Connect:Direct.

É possível transferir dados do arquivo para dados da mensagem do IBM MQ. As mensagens do IBM MQ podem ser lidas e usadas por aplicativos. Os seguintes tipos de transferência de arquivo para mensagem são suportados:

- Transferência de um único arquivo para uma única mensagem. A mensagem não tem um ID do grupo do IBM MQ configurado.
- Transferência de um único arquivo para várias mensagens, dividindo o arquivo em mensagens de um determinado comprimento. Todas as mensagens têm o mesmo ID do grupo do IBM MQ.
- Em um único arquivo para múltiplas mensagens, dividindo um arquivo de texto em um delimitador de expressão regular Java. Todas as mensagens têm o mesmo ID do grupo do IBM MQ.
- Transferência de um único arquivo para várias mensagens, dividindo um arquivo binário em um delimitador hexadecimal. Todas as mensagens têm o mesmo ID do grupo do IBM MQ.

Se desejar dividir um arquivo binário usando uma sequência de bytes como o delimitador, use o parâmetro **-sqdb** do comando **fteCreateTransfer**. Para obter mais informações, consulte [Parâmetro -sqdb](#).

Por padrão, as mensagens criadas por uma transferência de arquivo para mensagem são persistentes. As mensagens podem ser configuradas para não persistentes ou ter o valor de persistência definido pela fila de destino.

Se você especificar que um arquivo deve ser dividido em várias mensagens, todas as mensagens criadas a partir do arquivo terão o mesmo ID do grupo do IBM MQ. Se você não especificar que um arquivo deve ser dividido em várias mensagens, somente uma mensagem será criada a partir do arquivo e essa mensagem não terá o ID do grupo do IBM MQ configurado.

Se você estiver transferindo arquivos para mensagens grandes ou muitas mensagens pequenas, poderá ser necessário mudar algumas propriedades do IBM MQ ou do Managed File Transfer. Para obter informações, veja [Orientação para configurar atributos do MQ e propriedades do MFT associadas ao tamanho de mensagem](#).

Nota: Se a fila de destino for uma fila em cluster ou um alias para uma fila em cluster, você receberá uma mensagem de erro ao transferir um arquivo para uma fila, se a propriedade de agente

`enableClusterQueueInputOutput` não tiver sido configurada como `true`. Para obter mais informações, veja [O que fazer se a fila de destino for uma fila em cluster ou um alias para uma fila em cluster](#)

Tarefas relacionadas

[“Configurando um Agente para Executar Transferências de Arquivo para Mensagem” na página 269](#)

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente `enableQueueInputOutput` como `true`. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente `enableClusterQueueInputOutput` como `true`.

[“Exemplo: Transferindo um Único Arquivo para uma Única Mensagem” na página 270](#)

É possível especificar uma fila como o destino de uma transferência de arquivo usando o parâmetro **-dq** com o comando **fteCreateTransfer**. O arquivo de origem deve ser menor que o comprimento máximo da mensagem definido na fila de destino. A fila de destino não precisa estar no mesmo gerenciador de filas que o gerenciador de filas ao qual o agente de destino se conecta, mas estes dois gerenciadores de filas devem poder comunicar-se.

[“Exemplo: Dividindo um Único Arquivo em Várias Mensagens por Comprimento” na página 272](#)

É possível dividir um arquivo em várias mensagens do IBM MQ usando o parâmetro **-qs** do comando **fteCreateTransfer**. O arquivo é dividido em seções de comprimento fixo, cada uma delas é gravada em uma mensagem individual.

[“Exemplo: Dividindo um Arquivo de Texto com um Delimitador de Expressão Regular e Incluindo o Delimitador nas Mensagens” na página 275](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java e inclua a correspondência de expressão regular nas mensagens resultantes. Para fazer isso, use os parâmetros **-dqdt** e **-qi** do comando **fteCreateTransfer**.

[“Exemplo: Dividindo um Arquivo de Texto em Várias Mensagens Usando um Delimitador de Expressão Regular” na página 273](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java. Para fazer isso, use o parâmetro **-dqdt** do comando **fteCreateTransfer**.

[“Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem” na página 277](#)

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem” na página 279](#)

Os metadados definidos pelo usuário são configurados como uma propriedade de mensagem do IBM MQ na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagens do IBM MQ permitem que um aplicativo selecione mensagens para processar ou recuperar informações sobre uma mensagem sem acessar os cabeçalhos IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Iniciando uma Nova Transferência de Arquivos” na página 220](#)

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Referências relacionadas

[“Falha de uma transferência de arquivo para mensagem” na página 282](#)

Se uma transferência de arquivo para mensagem falhar após o agente ter iniciado a gravação de dados do arquivo na fila de destino, o agente gravará uma mensagem na fila para indicar a um aplicativo que está consumindo as mensagens que ocorreu uma falha.

[Propriedades de mensagem do MQ configuradas pelo MFT em mensagens gravadas nas filas de destino](#)
[Orientação para configurar atributos do MQ e propriedades do MFT associados ao tamanho da mensagem](#)

Configurando um Agente para Executar Transferências de Arquivo para Mensagem

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente `enableQueueInputOutput` como `true`. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente `enableClusterQueueInputOutput` como `true`.

Sobre esta tarefa

Se você tentar executar uma transferência de arquivo para mensagem para um agente de destino que não tem a propriedade `enableQueueInputOutput` configurada como `true`, a transferência falhará. A mensagem de log da transferência que é publicada para o gerenciador de filas de coordenação contém a seguinte mensagem:

```
BFGI00197E: An attempt to write to a queue was rejected by the destination agent. The agent must have enableQueueInputOutput=true set in the agent.properties file to support transferring to a queue.
```

Para habilitar o agente a gravar e ler a partir de filas, execute as seguintes etapas:

Procedimento

1. Pare o agente de destino usando o comando **fteStopAgent**.
2. Edite o arquivo `agent.properties` para incluir a linha `enableQueueInputOutput=true`.
O arquivo `agent.properties` está localizado no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name`.
3. Opcional: Edite o arquivo `agent.properties` para incluir a linha `enableClusterQueueInputOutput=true`. O arquivo `agent.properties` está localizado no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name`.
4. Inicie o agente de destino usando o comando **fteStartAgent**.

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens”](#) na página 267

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Tarefas relacionadas

[“Exemplo: Transferindo um Único Arquivo para uma Única Mensagem”](#) na página 270

É possível especificar uma fila como o destino de uma transferência de arquivo usando o parâmetro **-dq** com o comando **fteCreateTransfer**. O arquivo de origem deve ser menor que o comprimento máximo da mensagem definido na fila de destino. A fila de destino não precisa estar no mesmo gerenciador de filas que o gerenciador de filas ao qual o agente de destino se conecta, mas estes dois gerenciadores de filas devem poder comunicar-se.

[“Exemplo: Dividindo um Único Arquivo em Várias Mensagens por Comprimento”](#) na página 272

É possível dividir um arquivo em várias mensagens do IBM MQ usando o parâmetro **-qs** do comando **fteCreateTransfer**. O arquivo é dividido em seções de comprimento fixo, cada uma delas é gravada em uma mensagem individual.

[“Exemplo: Dividindo um Arquivo de Texto com um Delimitador de Expressão Regular e Incluindo o Delimitador nas Mensagens”](#) na página 275

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java e inclua a correspondência de expressão regular nas mensagens resultantes. Para fazer isso, use os parâmetros **-dqdt** e **-qi** do comando **fteCreateTransfer**.

[“Exemplo: Dividindo um Arquivo de Texto em Várias Mensagens Usando um Delimitador de Expressão Regular” na página 273](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java. Para fazer isso, use o parâmetro **-dqdt** do comando **fteCreateTransfer**.

[“Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem” na página 277](#)

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem” na página 279](#)

Os metadados definidos pelo usuário são configurados como uma propriedade de mensagem do IBM MQ na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagens do IBM MQ permitem que um aplicativo selecione mensagens para processar ou recuperar informações sobre uma mensagem sem acessar os cabeçalhos IBM MQ Message Descriptor (MQMD) ou MQRFH2.

Referências relacionadas

fteStopAgent

fteStartAgent

[O arquivo MFT `agent.properties`](#)

[“Falha de uma transferência de arquivo para mensagem” na página 282](#)

Se uma transferência de arquivo para mensagem falhar após o agente ter iniciado a gravação de dados do arquivo na fila de destino, o agente gravará uma mensagem na fila para indicar a um aplicativo que está consumindo as mensagens que ocorreu uma falha.

Exemplo: Transferindo um Único Arquivo para uma Única Mensagem

É possível especificar uma fila como o destino de uma transferência de arquivo usando o parâmetro **-dq** com o comando **fteCreateTransfer**. O arquivo de origem deve ser menor que o comprimento máximo da mensagem definido na fila de destino. A fila de destino não precisa estar no mesmo gerenciador de filas que o gerenciador de filas ao qual o agente de destino se conecta, mas estes dois gerenciadores de filas devem poder comunicar-se.

Sobre esta tarefa

O arquivo de origem é chamado `/tmp/single_record.txt` e está localizado no mesmo sistema que o agente de origem, AGENT_NEPTUNE. O agente de origem, AGENT_NEPTUNE, usa o gerenciador de filas QM_NEPTUNE. O agente de destino é AGENT_VENUS e este agente se conecta ao gerenciador de filas QM_VENUS. A fila de destino, RECEIVING_QUEUE, está localizada no gerenciador de filas QM_MERCURY. QM_MERCURY está na mesma rede do IBM MQ e pode ser acessado pelo gerenciador de filas QM_VENUS.

Procedimento

Digite o seguinte comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS
                 -dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

Se a fila de destino estiver em um gerenciador de filas diferente do gerenciador de filas usado pelo agente de destino, você deverá especificar o valor do parâmetro **-dq** no seguinte formato `queue_name@queue_manager_name`. Se você não especificar `@queue_manager_name` no valor, o agente de destino assumirá que a fila de destino está localizada no gerenciador de filas do agente de destino. A exceção é quando a propriedade do agente `enableClusterQueueInputOutput` foi configurada como `true`.

Neste caso o agente de destino usará procedimentos de resolução padrão IBM MQ para determinar onde a fila está localizada.

O agente de origem, AGENT_NEPTUNE, lê os dados do arquivo /tmp/single_record.txt e transfere esses dados para o agente de destino, AGENT_VENUS. O agente de destino, AGENT_VENUS, envia os dados para uma mensagem persistente na fila RECEIVING_QUEUE@QM_MERCURY. A mensagem não tem um ID do grupo do IBM MQ configurado.

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens” na página 267](#)

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Tarefas relacionadas

[“Configurando um Agente para Executar Transferências de Arquivo para Mensagem” na página 269](#)

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente enableQueueInputOutput como true. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente enableClusterQueueInputOutput como true.

[“Exemplo: Dividindo um Único Arquivo em Várias Mensagens por Comprimento” na página 272](#)

É possível dividir um arquivo em várias mensagens do IBM MQ usando o parâmetro **-qs** do comando **fteCreateTransfer**. O arquivo é dividido em seções de comprimento fixo, cada uma delas é gravada em uma mensagem individual.

[“Exemplo: Dividindo um Arquivo de Texto com um Delimitador de Expressão Regular e Incluindo o Delimitador nas Mensagens” na página 275](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java e inclua a correspondência de expressão regular nas mensagens resultantes. Para fazer isso, use os parâmetros **-dqdt** e **-qi** do comando **fteCreateTransfer**.

[“Exemplo: Dividindo um Arquivo de Texto em Várias Mensagens Usando um Delimitador de Expressão Regular” na página 273](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java. Para fazer isso, use o parâmetro **-dqdt** do comando **fteCreateTransfer**.

[“Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem” na página 277](#)

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem” na página 279](#)

Os metadados definidos pelo usuário são configurados como uma propriedade de mensagem do IBM MQ na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagens do IBM MQ permitem que um aplicativo selecione mensagens para processar ou recuperar informações sobre uma mensagem sem acessar os cabeçalhos IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Iniciando uma Nova Transferência de Arquivos” na página 220](#)

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Referências relacionadas

[“Falha de uma transferência de arquivo para mensagem” na página 282](#)

Se uma transferência de arquivo para mensagem falhar após o agente ter iniciado a gravação de dados do arquivo na fila de destino, o agente gravará uma mensagem na fila para indicar a um aplicativo que está consumindo as mensagens que ocorreu uma falha.

Exemplo: Dividindo um Único Arquivo em Várias Mensagens por Comprimento

É possível dividir um arquivo em várias mensagens do IBM MQ usando o parâmetro **-qs** do comando **fteCreateTransfer**. O arquivo é dividido em seções de comprimento fixo, cada uma delas é gravada em uma mensagem individual.

Sobre esta tarefa

O arquivo de origem é chamado `/tmp/source.file` e tem 36 KB de tamanho. O arquivo de origem está localizado no mesmo sistema do agente de origem `AGENT_NEPTUNE`. O agente de origem, `AGENT_NEPTUNE`, conecta-se ao gerenciador de filas `QM_NEPTUNE`. O agente de destino é `AGENT_MERCURY`, que se conecta ao gerenciador de filas `QM_MERCURY`. A fila de destino, `RECEIVING_QUEUE`, também está localizada no gerenciador de filas `QM_MERCURY`. A transferência divide o arquivo de origem em seções com 1 KB de tamanho e grava cada uma dessas seções em uma mensagem em `RECEIVING_QUEUE`.

Procedimento

Digite o seguinte comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

O agente de origem, `AGENT_NEPTUNE`, lê os dados do arquivo `/tmp/source.file` e transfere esses dados para o agente de destino, `AGENT_MERCURY`. O agente de destino, `AGENT_MERCURY`, grava os dados em trinta e seis mensagens persistentes de 1 KB na fila `RECEIVING_QUEUE@QM_MERCURY`. Todas essas mensagens têm o mesmo ID do grupo do IBM MQ e a última mensagem no grupo tem a sinalização `LAST_MSG_IN_GROUP` do IBM MQ configurada.

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens” na página 267](#)

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Tarefas relacionadas

[“Configurando um Agente para Executar Transferências de Arquivo para Mensagem” na página 269](#)

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente `enableQueueInputOutput` como `true`. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente `enableClusterQueueInputOutput` como `true`.

[“Exemplo: Transferindo um Único Arquivo para uma Única Mensagem” na página 270](#)

É possível especificar uma fila como o destino de uma transferência de arquivo usando o parâmetro **-dq** com o comando **fteCreateTransfer**. O arquivo de origem deve ser menor que o comprimento máximo da mensagem definido na fila de destino. A fila de destino não precisa estar no mesmo gerenciador de filas que o gerenciador de filas ao qual o agente de destino se conecta, mas estes dois gerenciadores de filas devem poder comunicar-se.

[“Exemplo: Dividindo um Arquivo de Texto com um Delimitador de Expressão Regular e Incluindo o Delimitador nas Mensagens” na página 275](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java e inclua a correspondência de expressão regular nas mensagens resultantes. Para fazer isso, use os parâmetros **-dqdt** e **-qi** do comando **fteCreateTransfer**.

[“Exemplo: Dividindo um Arquivo de Texto em Várias Mensagens Usando um Delimitador de Expressão Regular” na página 273](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java. Para fazer isso, use o parâmetro **-dqdt** do comando **fteCreateTransfer**.

[“Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem” na página 277](#)

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem” na página 279](#)

Os metadados definidos pelo usuário são configurados como uma propriedade de mensagem do IBM MQ na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagens do IBM MQ permitem que um aplicativo selecione mensagens para processar ou recuperar informações sobre uma mensagem sem acessar os cabeçalhos IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Iniciando uma Nova Transferência de Arquivos” na página 220](#)

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Referências relacionadas

[“Falha de uma transferência de arquivo para mensagem” na página 282](#)

Se uma transferência de arquivo para mensagem falhar após o agente ter iniciado a gravação de dados do arquivo na fila de destino, o agente gravará uma mensagem na fila para indicar a um aplicativo que está consumindo as mensagens que ocorreu uma falha.

Exemplo: Dividindo um Arquivo de Texto em Várias Mensagens Usando um Delimitador de Expressão Regular

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java. Para fazer isso, use o parâmetro **-dqdt** do comando **fteCreateTransfer**.

Sobre esta tarefa

O arquivo é dividido em seções de comprimento variável, cada uma delas é gravada em uma mensagem individual. O arquivo de texto é dividido em cada ponto no qual o texto no arquivo corresponde a uma determinada expressão regular. O arquivo de origem é chamado `/tmp/names.text` e tem o seguinte conteúdo:

```
Jenny Jones,John Smith,Jane Brown
```

A expressão regular que especifica onde dividir o arquivo é o caractere vírgula (`,`).

O arquivo de origem está localizado no mesmo sistema que o agente de origem `AGENT_NEPTUNE`, que se conecta ao gerenciador de filas `QM_NEPTUNE`. A fila de destino, `RECEIVING_QUEUE`, está localizada no gerenciador de filas `QM_MERCURY`. `QM_MERCURY` também é o gerenciador de filas usado pelo agente de destino `AGENT_MERCURY`. A transferência divide o arquivo de origem em seções e grava cada uma dessas seções em uma mensagem em `RECEIVING_QUEUE`.

Procedimento

Digite o seguinte comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

O agente de origem, `AGENT_NEPTUNE`, lê os dados do arquivo `/tmp/names.text` e transfere esses dados para o agente de destino, `AGENT_MERCURY`. O agente de destino, `AGENT_MERCURY`, grava os dados em três mensagens persistentes na fila `RECEIVING_QUEUE`. Todas essas mensagens têm o mesmo

ID do grupo do IBM MQ e a última mensagem no grupo tem a sinalização LAST_MSG_IN_GROUP do IBM MQ configurada.

Os dados nas mensagens são os seguintes.

- Primeira mensagem:

Jenny Jones

- Segunda mensagem:

John Smith

- Terceira mensagem:

Jane Brown

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens” na página 267](#)

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Tarefas relacionadas

[“Configurando um Agente para Executar Transferências de Arquivo para Mensagem” na página 269](#)

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente `enableQueueInputOutput` como `true`. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente `enableClusterQueueInputOutput` como `true`.

[“Exemplo: Transferindo um Único Arquivo para uma Única Mensagem” na página 270](#)

É possível especificar uma fila como o destino de uma transferência de arquivo usando o parâmetro **-dq** com o comando **fteCreateTransfer**. O arquivo de origem deve ser menor que o comprimento máximo da mensagem definido na fila de destino. A fila de destino não precisa estar no mesmo gerenciador de filas que o gerenciador de filas ao qual o agente de destino se conecta, mas estes dois gerenciadores de filas devem poder comunicar-se.

[“Exemplo: Dividindo um Único Arquivo em Várias Mensagens por Comprimento” na página 272](#)

É possível dividir um arquivo em várias mensagens do IBM MQ usando o parâmetro **-qs** do comando **fteCreateTransfer**. O arquivo é dividido em seções de comprimento fixo, cada uma delas é gravada em uma mensagem individual.

[“Exemplo: Dividindo um Arquivo de Texto com um Delimitador de Expressão Regular e Incluindo o Delimitador nas Mensagens” na página 275](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java e inclua a correspondência de expressão regular nas mensagens resultantes. Para fazer isso, use os parâmetros **-dqdt** e **-qi** do comando **fteCreateTransfer**.

[“Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem” na página 277](#)

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem” na página 279](#)

Os metadados definidos pelo usuário são configurados como uma propriedade de mensagem do IBM MQ na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagens

do IBM MQ permitem que um aplicativo selecione mensagens para processar ou recuperar informações sobre uma mensagem sem acessar os cabeçalhos IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Iniciando uma Nova Transferência de Arquivos” na página 220](#)

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Referências relacionadas

[“Falha de uma transferência de arquivo para mensagem” na página 282](#)

Se uma transferência de arquivo para mensagem falhar após o agente ter iniciado a gravação de dados do arquivo na fila de destino, o agente gravará uma mensagem na fila para indicar a um aplicativo que está consumindo as mensagens que ocorreu uma falha.

[Expressões regulares usadas pelo MFT](#)

Exemplo: Dividindo um Arquivo de Texto com um Delimitador de Expressão Regular e Incluindo o Delimitador nas Mensagens

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java e inclua a correspondência de expressão regular nas mensagens resultantes. Para fazer isso, use os parâmetros **-dqdt** e **-qi** do comando **fteCreateTransfer**.

Sobre esta tarefa

Transfira um único arquivo de texto para várias mensagens em uma fila. O arquivo é dividido em seções de comprimento variável, cada uma delas é gravada em uma mensagem individual. O arquivo de texto é dividido em cada ponto no qual o texto no arquivo corresponde a uma determinada expressão regular. O arquivo de origem é chamado `/tmp/customers.text` e tem o seguinte conteúdo:

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314

Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42

Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

A expressão regular que especifica onde dividir o arquivo é `Customer\number: \s\d+`, que corresponde ao texto "Número do cliente: " seguido por qualquer número de dígitos. Expressões regulares especificadas na linha de comandos devem estar entre aspas duplas para evitar que o shell de comando avalie a expressão regular. A expressão regular é avaliada como uma expressão regular de Java. Para obter mais informações, consulte [Expressões regulares usadas pelo MFT](#).

Por padrão, o número de caracteres ao qual uma expressão regular pode corresponder é definido para cinco. A expressão regular usada nesse exemplo corresponde a cadeias que são maiores que cinco caracteres. Para ativar correspondências maiores que cinco caracteres, edite o arquivo de propriedades para incluir a propriedade **maxDelimiterMatchLength**.

Por padrão, o texto que corresponde à expressão regular não é incluído nas mensagens. Para incluir o texto que corresponde à expressão regular nas mensagens, como nesse exemplo, use o parâmetro **-qi**. O arquivo de origem está localizado no mesmo sistema que o agente de origem `AGENT_NEPTUNE`, que se conecta ao gerenciador de filas `QM_NEPTUNE`. A fila de destino, `RECEIVING_QUEUE`, está localizada no gerenciador de filas `QM_MERCURY`. `QM_MERCURY` também é o gerenciador de filas usado pelo agente de destino `AGENT_MERCURY`. A transferência divide o arquivo de origem em seções e grava cada uma dessas seções em uma mensagem em `RECEIVING_QUEUE`.

Procedimento

1. Pare o agente do destino usando o seguinte comando:

```
fteStopAgent AGENT_MERCURY
```

2. Inclua a seguinte linha no arquivo de propriedades do agente para AGENT_MERCURY:

```
maxDelimiterMatchLength=25
```

Nota: Aumentar o valor de **maxDelimiterMatchLength** pode diminuir o desempenho.

3. Inicie o agente do destino usando o seguinte comando:

```
fteStartAgent AGENT_MERCURY
```

4. Digite o seguinte comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE  
text -dqdt "Customer\snumber:\s\d+" -qi -dqdp postfix /tmp/customers.text
```

O agente de origem, AGENT_NEPTUNE, lê os dados do arquivo /tmp/customers.text e transfere esses dados para o agente de destino, AGENT_MERCURY. O agente de destino, AGENT_MERCURY, grava os dados em três mensagens persistentes na fila RECEIVING_QUEUE. Todas essas mensagens têm o mesmo ID do grupo do IBM MQ e a última mensagem no grupo tem a sinalização LAST_MSG_IN_GROUP do IBM MQ configurada.

Os dados nas mensagens são os seguintes.

- Primeira mensagem:

```
Customer name: John Smith  
Customer contact details: john@example.net  
Customer number: 314
```

- Segunda mensagem:

```
Customer name: Jane Brown  
Customer contact details: jane@example.com  
Customer number: 42
```

- Terceira mensagem:

```
Customer name: James Jones  
Customer contact details: jjones@example.net  
Customer number: 26
```

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens” na página 267](#)

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Tarefas relacionadas

[“Configurando um Agente para Executar Transferências de Arquivo para Mensagem” na página 269](#)

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente enableQueueInputOutput como true. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente enableClusterQueueInputOutput como true.

[“Exemplo: Transferindo um Único Arquivo para uma Única Mensagem” na página 270](#)

É possível especificar uma fila como o destino de uma transferência de arquivo usando o parâmetro **-dq** com o comando **fteCreateTransfer**. O arquivo de origem deve ser menor que o comprimento máximo da mensagem definido na fila de destino. A fila de destino não precisa estar no mesmo gerenciador de filas que o gerenciador de filas ao qual o agente de destino se conecta, mas estes dois gerenciadores de filas devem poder comunicar-se.

“Exemplo: Dividindo um Único Arquivo em Várias Mensagens por Comprimento” na página 272

É possível dividir um arquivo em várias mensagens do IBM MQ usando o parâmetro **-qs** do comando **fteCreateTransfer**. O arquivo é dividido em seções de comprimento fixo, cada uma delas é gravada em uma mensagem individual.

“Exemplo: Dividindo um Arquivo de Texto em Várias Mensagens Usando um Delimitador de Expressão Regular” na página 273

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java. Para fazer isso, use o parâmetro **-dqdt** do comando **fteCreateTransfer**.

“Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem” na página 277

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

“Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem” na página 279

Os metadados definidos pelo usuário são configurados como uma propriedade de mensagem do IBM MQ na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagens do IBM MQ permitem que um aplicativo selecione mensagens para processar ou recuperar informações sobre uma mensagem sem acessar os cabeçalhos IBM MQ Message Descriptor (MQMD) ou MQRFH2.

“Iniciando uma Nova Transferência de Arquivos” na página 220

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Referências relacionadas

[O arquivo MFT agent.properties](#)

[Expressões regulares usadas pelo MFT](#)

Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

Sobre esta tarefa

Inclua o parâmetro **-qmp true** no comando **fteCreateTransfer**. Nesse exemplo, o ID do usuário MQMD do usuário que envia o comando `larmer`.

Procedimento

Digite o seguinte comando:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true
-t text /tmp/source_file.txt
```

As propriedades de mensagem do IBM MQ da primeira mensagem gravada pelo agente de destino AGENT_SATURN, na fila MY_QUEUE, no gerenciador de filas MyQM, são configuradas para estes valores:

```
usr.WMQFTETransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd
usr.WMQFTETransferMode=text
usr.WMQFTESourceAgent=AGENT_JUPITER
usr.WMQFTEDestinationAgent=AGENT_SATURN
usr.WMQFTEFileName=source_file.txt
usr.WMQFTEFileSize=1024
usr.WMQFTEFileLastModified=1273740879040
usr.WMQFTEFileIndex=0
usr.WMQFTEMqmdUser=larmer
```

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens” na página 267](#)

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Tarefas relacionadas

[“Configurando um Agente para Executar Transferências de Arquivo para Mensagem” na página 269](#)

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente `enableQueueInputOutput` como `true`. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente `enableClusterQueueInputOutput` como `true`.

[“Exemplo: Transferindo um Único Arquivo para uma Única Mensagem” na página 270](#)

É possível especificar uma fila como o destino de uma transferência de arquivo usando o parâmetro **-dq** com o comando **fteCreateTransfer**. O arquivo de origem deve ser menor que o comprimento máximo da mensagem definido na fila de destino. A fila de destino não precisa estar no mesmo gerenciador de filas que o gerenciador de filas ao qual o agente de destino se conecta, mas estes dois gerenciadores de filas devem poder comunicar-se.

[“Exemplo: Dividindo um Único Arquivo em Várias Mensagens por Comprimento” na página 272](#)

É possível dividir um arquivo em várias mensagens do IBM MQ usando o parâmetro **-qs** do comando **fteCreateTransfer**. O arquivo é dividido em seções de comprimento fixo, cada uma delas é gravada em uma mensagem individual.

[“Exemplo: Dividindo um Arquivo de Texto com um Delimitador de Expressão Regular e Incluindo o Delimitador nas Mensagens” na página 275](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java e inclua a correspondência de expressão regular nas mensagens resultantes. Para fazer isso, use os parâmetros **-dqdt** e **-qi** do comando **fteCreateTransfer**.

[“Exemplo: Dividindo um Arquivo de Texto em Várias Mensagens Usando um Delimitador de Expressão Regular” na página 273](#)

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java. Para fazer isso, use o parâmetro **-dqdt** do comando **fteCreateTransfer**.

[“Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem” na página 279](#)

Os metadados definidos pelo usuário são configurados como uma propriedade de mensagem do IBM MQ na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagens do IBM MQ permitem que um aplicativo selecione mensagens para processar ou recuperar informações sobre uma mensagem sem acessar os cabeçalhos IBM MQ Message Descriptor (MQMD) ou MQRFH2.

[“Iniciando uma Nova Transferência de Arquivos” na página 220](#)

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Referências relacionadas

[“Falha de uma transferência de arquivo para mensagem” na página 282](#)

Se uma transferência de arquivo para mensagem falhar após o agente ter iniciado a gravação de dados do arquivo na fila de destino, o agente gravará uma mensagem na fila para indicar a um aplicativo que está consumindo as mensagens que ocorreu uma falha.

Propriedades de mensagem do MQ configuradas pelo MFT em mensagens gravadas nas filas de destino

Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem

Os metadados definidos pelo usuário são configurados como uma propriedade de mensagem do IBM MQ na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagens do IBM MQ permitem que um aplicativo selecione mensagens para processar ou recuperar informações sobre uma mensagem sem acessar os cabeçalhos IBM MQ Message Descriptor (MQMD) ou MQRFH2.

Sobre esta tarefa

Inclua os parâmetros `-qmp true` e `-md account=123456` no comando **fteCreateTransfer** para configurar a propriedade `usr.account` como 123456 no cabeçalho RFH2.

Procedimento

Digite o seguinte comando:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM  
-qmp true -md account=123456 /tmp/source_file.txt
```

Além do conjunto padrão de propriedades de mensagem do IBM MQ, a propriedade definida pelo usuário é configurada no cabeçalho da mensagem da primeira mensagem gravada pelo agente de destino AGENT_SATURN, na fila MY_QUEUE, no gerenciador de filas MyQM. O cabeçalho é configurado com o seguinte valor:

```
usr.account=123456
```

O prefixo `usr` é incluído no início do nome dos metadados definidos pelo usuário.

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens” na página 267](#)

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Tarefas relacionadas

[“Configurando um Agente para Executar Transferências de Arquivo para Mensagem” na página 269](#)

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente `enableQueueInputOutput` como `true`. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente `enableClusterQueueInputOutput` como `true`.

[“Exemplo: Transferindo um Único Arquivo para uma Única Mensagem” na página 270](#)

É possível especificar uma fila como o destino de uma transferência de arquivo usando o parâmetro **-dq** com o comando **fteCreateTransfer**. O arquivo de origem deve ser menor que o comprimento máximo da mensagem definido na fila de destino. A fila de destino não precisa estar no mesmo gerenciador de filas que o gerenciador de filas ao qual o agente de destino se conecta, mas estes dois gerenciadores de filas devem poder comunicar-se.

[“Exemplo: Dividindo um Único Arquivo em Várias Mensagens por Comprimento” na página 272](#)

É possível dividir um arquivo em várias mensagens do IBM MQ usando o parâmetro **-qs** do comando **fteCreateTransfer**. O arquivo é dividido em seções de comprimento fixo, cada uma delas é gravada em uma mensagem individual.

“Exemplo: Dividindo um Arquivo de Texto com um Delimitador de Expressão Regular e Incluindo o Delimitador nas Mensagens” na página 275

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java e inclua a correspondência de expressão regular nas mensagens resultantes. Para fazer isso, use os parâmetros **-dqdt** e **-qi** do comando **fteCreateTransfer**.

“Exemplo: Dividindo um Arquivo de Texto em Várias Mensagens Usando um Delimitador de Expressão Regular” na página 273

Transfira um único arquivo de texto para múltiplas mensagens, dividindo o arquivo em cada correspondência de uma determinada expressão regular de Java. Para fazer isso, use o parâmetro **-dqdt** do comando **fteCreateTransfer**.

“Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem” na página 277

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

“Iniciando uma Nova Transferência de Arquivos” na página 220

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Referências relacionadas

Propriedades de mensagem do MQ configuradas pelo MFT em mensagens gravadas nas filas de destino

Exemplo: Incluindo uma Propriedade de Mensagem Definida pelo Usuário para uma Transferência de Arquivo para Mensagem

Se estiver usando o Managed File Transfer para transferências gerenciadas de mensagem para arquivo, é possível incluir uma propriedade de mensagem definida pelo usuário para a mensagem resultante.

Sobre esta tarefa

É possível usar qualquer um dos seguintes métodos para definir uma propriedade de mensagem customizada:

- Especifique o parâmetro **-md** na solicitação de transferência. Para obter mais informações, consulte “Exemplo: Configurando Propriedades Definidas pelo Usuário em uma Transferência de Arquivo para Mensagem” na página 279.
- Use uma tarefa do Ant. É possível usar `fte:filecopy` ou `fte:filemove`. O exemplo a seguir é uma tarefa `fte:filecopy`:

```
<project xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs" default="complete">
<!-- Initialise the properties used in this script.-->

<target name="init" description="initialise task properties">
    <property name="src.file" value="/home/user/file1.bin"/>
    <property name="dst.queue" value="TEST.QUEUE@qm2"/>
    <fte:uuid property="job.name" length="8"
prefix="copyjob#"/>
</target>
<target name="step1" depends="init" description="transfer file">

<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result">

<fte:metadata>
<fte:entry name="fileName" value="{FileName}"/>
</fte:metadata>

<fte:filespec srcfilespec="{src.file}" dstqueue="{dst.queue}"
dstmsgprops="true"/>
```



```

</fte:filecopy>
</target>
</project>

```

- Use um monitor de recurso e uma substituição de variável. O exemplo a seguir mostra algum XML de tarefa de transferência:

```

<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
  <name>METADATA</name>
  <pollInterval units="minutes">5</pollInterval>
  <batch maxSize="5"/>
  <agent>AGENT1</agent>
  <resources>
    <directory recursionLevel="0">e:\temp</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <allOf>
        <condition>
          <fileMatch>
            <pattern>*.txt</pattern>
          </fileMatch>
        </condition>
      </allOf>
    </conditions>
  </triggerMatch>
  <tasks>
    <task>
      <name/>
      <transfer>
        <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xmlns:noNamespaceSchemaLocation="FileTransfer.xsd">
          <managedTransfer>
            <originator>
              <hostName>mqjason.raleigh.ibm.com.</hostName>
              <userID>administrator</userID>
            </originator>
            <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
            <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
            <transferSet priority="0">
              <metaDataSet>
                <metaData key="FileName">${FileName}</metaData>
              </metaDataSet>
              <item checksumMethod="MD5" mode="text">
                <source disposition="delete" recursive="false">
                  <file>${FilePath}</file>
                </source>
                <destination type="queue">
                  <queue persistent="true">
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
                </destination>
              </item>
            </transferSet>
          </managedTransfer>
        </request>
      </transfer>
    </task>
  </tasks>
  <originator>
    <hostName>mqjason.raleigh.ibm.com.</hostName>
    <userID>administrator</userID>
  </originator>
</monitor:monitor>

```

Tarefas relacionadas

[“Exemplo: Configurando propriedades de mensagem do IBM MQ em uma transferência de arquivo para mensagem” na página 277](#)

É possível usar o parâmetro **-qmp** no comando **fteCreateTransfer** para especificar se as propriedades de mensagem do IBM MQ são configuradas na primeira mensagem gravada na fila de destino pela transferência. As propriedades de mensagem do IBM MQ permitem que um aplicativo selecione mensagens para serem processadas ou recupere informações sobre uma mensagem sem acessar os cabeçalhos do IBM MQ Message Descriptor (MQMD) ou MQRFH2.

Referências relacionadas

[Tarefa fte:filecopy Ant](#)

[Tarefa fte:filemove Ant](#)

Falha de uma transferência de arquivo para mensagem

Se uma transferência de arquivo para mensagem falhar após o agente ter iniciado a gravação de dados do arquivo na fila de destino, o agente gravará uma mensagem na fila para indicar a um aplicativo que está consumindo as mensagens que ocorreu uma falha.

A mensagem gravada na fila de destino se uma falha ocorrer:

- estará em branco
- terá o mesmo ID do grupo do IBM MQ que a mensagem anterior gravada na fila de destino pelo agente
- terá a sinalização LAST_MSG_IN_GROUP do IBM MQ configurada
- conterá propriedades de mensagem adicionais do IBM MQ, se as propriedades de mensagem estiverem ativadas. Para obter mais informações, veja o tópico [Propriedades de mensagem do MQ configuradas pelo MFT em mensagens gravadas em filas de destino](#).

exemplo

Uma transferência é requisitada pela execução do seguinte comando:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE  
-qmp true -qs 1K /tmp/source1.txt
```

O arquivo `source1.txt` é 48 KB. A transferência divide esse arquivo em mensagens de 1 KB e grava essas mensagens na fila de destino `RECEIVING_QUEUE`.

Enquanto a transferência está em andamento, após o agente ter gravado 16 mensagens para `RECEIVING_QUEUE`, uma falha ocorre no agente de origem.

O agente grava uma mensagem em branco para `RECEIVING_QUEUE`. Além do conjunto de propriedades de mensagem padrão, a mensagem em branco tem o seguinte conjunto de propriedades de mensagem:

```
usr.WMQFTEResultCode = 40  
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

Em IBM MQ 9.3.0, quando uma transferência de um arquivo falha, devido a um erro de verificação de tamanho do delimitador, apenas uma mensagem vazia é enviada. Além disso, as propriedades do usuário são incluídas nesta mensagem se a falha de transferência foi devido ao delimitador exceder o tamanho de configuração no agente de destino.

Conceitos relacionados

[“Transferindo dados de arquivos para mensagens” na página 267](#)

É possível usar um recurso arquivo para mensagem do Managed File Transfer para transferir os dados de um arquivo para uma única mensagem ou várias mensagens, em uma fila do IBM MQ.

Tarefas relacionadas

[“Configurando um Agente para Executar Transferências de Arquivo para Mensagem” na página 269](#)

Por padrão, os agentes não podem executar transferências de arquivo para mensagem ou de mensagem para arquivo. Para ativar essa função, deve-se configurar a propriedade do agente `enableQueueInputOutput` como `true`. Para ativar a gravação nas filas em cluster do IBM MQ, deve-se também configurar a propriedade do agente `enableClusterQueueInputOutput` como `true`.

[“Iniciando uma Nova Transferência de Arquivos” na página 220](#)

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

Referências relacionadas

[O arquivo MFT `agent.properties`](#)

[Propriedades de mensagem do MQ configuradas pelo MFT em mensagens gravadas nas filas de destino](#)

Transferindo Dados de Mensagens para Arquivos

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

Para obter informações sobre transferências de arquivo para mensagem, consulte [“Transferindo dados de arquivos para mensagens” na página 267](#).



Atenção: O agente de origem para uma transferência de mensagem para arquivo não pode ser um agente de ponte de protocolo ou um agente de ponte `Connect:Direct`.

É possível transferir os dados da mensagem do IBM MQ para um arquivo. Os seguintes tipos de transferência de mensagem para arquivo são suportados:

- De uma única mensagem para um único arquivo
- De várias mensagens para um único arquivo
- De várias mensagens com o mesmo ID do grupo do IBM MQ para um único arquivo.
- De várias mensagens para um único arquivo, incluindo um delimitador de texto ou binário entre os dados de cada mensagem gravada no arquivo.

Se você estiver transferindo arquivos de mensagens grandes ou muitas mensagens pequenas, poderá ser necessário mudar algumas propriedades do IBM MQ ou do Managed File Transfer. Para obter mais informações, consulte [Orientação para configuração de atributos MQ e propriedades MFT associadas ao tamanho da mensagem](#).

Em uma transferência de mensagem para arquivo, o agente de origem procura as mensagens da fila de origem, ao contrário do GET destrutivo em versões anteriores do IBM MQ. As mensagens são removidas da fila de origem depois que todas as mensagens (em um grupo, se o agrupamento de mensagens for usado) são procuradas e os dados gravados no arquivo de destino. Isso permite que as mensagens permaneçam na fila de origem, caso uma transferência falhe ou seja cancelada. Em razão dessa mudança, a autoridade para PROCURAR também deve ser fornecida junto à autoridade GET para executar transferências de mensagem para arquivo.

Managed File Transfer compara o identificador de transferência e o valor do atributo `groupId` dentro da carga útil de XML da solicitação de transferência. Se esses dois identificadores são equivalentes, o agente de origem usa o identificador como uma opção de correspondência de identificador de mensagem (em oposição a uma opção de correspondência de identificador de grupo) para a primeira tentativa de MQGET que é feita na fila de entrada para a transferência de mensagem para arquivo.

Tarefas relacionadas

[“Exemplo: configurando um recurso do MFT” na página 243](#)

É possível especificar uma fila do IBM MQ como o recurso a ser monitorado por um monitor de recurso usando o parâmetro `-mq` com o comando `fteCreateMonitor`.

Referências relacionadas

[Propriedades de mensagem do MQ lidas pelo MFT de mensagens em filas de origem](#)

Configurando um agente para executar transferências de mensagem para arquivo

Por padrão, os agentes não podem executar transferências de mensagem para arquivo, ou de arquivo para mensagem. Para ativar essa função é necessário configurar a propriedade do agente `enableQueueInputOutput` para `true`.

Sobre esta tarefa

Se você tentar executar uma transferência de mensagem para arquivo de um agente de origem que não possui a propriedade `enableQueueInputOutput` configurada para `true`, a transferência falha. A mensagem de log da transferência que é publicada para o gerenciador de filas de coordenação contém a seguinte mensagem:

```
BFGI00197E: An attempt to read from a queue was rejected by the source agent.  
The agent must have enableQueueInputOutput=true set in the agent.properties file  
to support transferring from a queue.
```

Para habilitar o agente a gravar e ler a partir de filas, execute as seguintes etapas:

Procedimento

1. Pare o agente de origem usando o comando **fteStopAgent**.
2. Edite o arquivo `agent.properties` para incluir a linha `enableQueueInputOutput=true`.
O arquivo `agent.properties` está localizado no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/source_agent_name`.
3. Inicie o agente de origem usando o comando **fteStartAgent**.

Conceitos relacionados

[“Transferindo Dados de Mensagens para Arquivos”](#) na página 283

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

Tarefas relacionadas

[“Exemplo: Transferindo de uma Fila para um Único Arquivo”](#) na página 285

É possível especificar uma fila do IBM MQ como a origem de uma transferência de arquivos usando o parâmetro **-sq** com o comando **fteCreateTransfer**.

[“Exemplo: Transferindo um Grupo de Mensagens de uma Fila para uma Única Fila”](#) na página 286

É possível especificar um único grupo completo em uma fila do IBM MQ como a origem de uma transferência de arquivo usando os parâmetros **-sq** e **-sqgi** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador de Texto antes dos Dados a partir de cada Mensagem”](#) na página 287

Quando estiver transferindo em modo de texto de uma fila de origem para um arquivo, é possível especificar que um delimitador de texto seja inserido antes dos dados a partir de mensagens individuais, usando os parâmetros **-sq**, **-sqdt** e **-sqdp** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador Binário Após os Dados a Partir de Cada Mensagem”](#) na página 288

Ao transferir em modo binário de uma fila de origem para um arquivo, é possível especificar que um delimitador binário seja inserido após os dados de mensagens individuais, usando os parâmetros **-sq**, **-sqdb** e **-sqdp** com o comando **fteCreateTransfer**.

[“Monitorando uma Fila e Usando Substituição de Variável”](#) na página 249

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira

mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

[“Exemplo: falhando uma transferência de mensagem para arquivo usando as propriedades de mensagem do IBM MQ” na página 292](#)

É possível fazer com que uma transferência de mensagem para arquivo falhe, configurando a propriedade de mensagem `usr.UserReturnCode` IBM MQ como um valor diferente de zero. Também é possível especificar informações complementares sobre o motivo da falha, configurando a propriedade de mensagem `usr.UserSupplement` IBM MQ.

Referências relacionadas

[O arquivo MFT agent.properties](#)

Exemplo: Transferindo de uma Fila para um Único Arquivo

É possível especificar uma fila do IBM MQ como a origem de uma transferência de arquivos usando o parâmetro **-sq** com o comando **fteCreateTransfer**.

Sobre esta tarefa

Os dados de origem estão contidos em três mensagens na fila `START_QUEUE`. Essa fila deve estar no gerenciador de filas do agente de origem, `QM_NEPTUNE`.

Procedimento

Digite o seguinte comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE
                  -da AGENT_VENUS -df /out/three_to_one.txt
                  -sq START_QUEUE
```

Os dados nas mensagens na fila `START_QUEUE` são gravados no arquivo `/out/three_to_one.txt` no sistema no qual `AGENT_VENUS` está executando.

Conceitos relacionados

[“Transferindo Dados de Mensagens para Arquivos” na página 283](#)

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

Tarefas relacionadas

[“Configurando um agente para executar transferências de mensagem para arquivo” na página 284](#)

Por padrão, os agentes não podem executar transferências de mensagem para arquivo, ou de arquivo para mensagem. Para ativar essa função é necessário configurar a propriedade do agente `enableQueueInputOutput` para `true`.

[“Exemplo: Transferindo um Grupo de Mensagens de uma Fila para uma Única Fila” na página 286](#)

É possível especificar um único grupo completo em uma fila do IBM MQ como a origem de uma transferência de arquivo usando os parâmetros **-sq** e **-sqgi** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador de Texto antes dos Dados a partir de cada Mensagem” na página 287](#)

Quando estiver transferindo em modo de texto de uma fila de origem para um arquivo, é possível especificar que um delimitador de texto seja inserido antes dos dados a partir de mensagens individuais, usando os parâmetros **-sq**, **-sqdt** e **-sqdp** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador Binário Após os Dados a Partir de Cada Mensagem” na página 288](#)

Por padrão, os agentes não podem executar transferências de mensagem para arquivo, ou de arquivo para mensagem. Para ativar essa função é necessário configurar a propriedade do agente `enableQueueInputOutput` para `true`.

[“Exemplo: Transferindo de uma Fila para um Único Arquivo” na página 285](#)

É possível especificar uma fila do IBM MQ como a origem de uma transferência de arquivos usando o parâmetro **-sq** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador de Texto antes dos Dados a partir de cada Mensagem” na página 287](#)

Quando estiver transferindo em modo de texto de uma fila de origem para um arquivo, é possível especificar que um delimitador de texto seja inserido antes dos dados a partir de mensagens individuais, usando os parâmetros **-sq**, **-sqdt** e **-sqdp** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador Binário Após os Dados a Partir de Cada Mensagem” na página 288](#)

Ao transferir em modo binário de uma fila de origem para um arquivo, é possível especificar que um delimitador binário seja inserido após os dados de mensagens individuais, usando os parâmetros **-sq**, **-sqdb** e **-sqdp** com o comando **fteCreateTransfer**.

[“Monitorando uma Fila e Usando Substituição de Variável” na página 249](#)

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

[“Exemplo: falhando uma transferência de mensagem para arquivo usando as propriedades de mensagem do IBM MQ” na página 292](#)

É possível fazer com que uma transferência de mensagem para arquivo falhe, configurando a propriedade de mensagem `usr.UserReturnCode` IBM MQ como um valor diferente de zero. Também é possível especificar informações complementares sobre o motivo da falha, configurando a propriedade de mensagem `usr.UserSupplement` IBM MQ.

Referências relacionadas

[fteCreateTransfer](#): iniciar uma nova transferência de arquivos

Exemplo: Inserindo um Delimitador de Texto antes dos Dados a partir de cada Mensagem

Quando estiver transferindo em modo de texto de uma fila de origem para um arquivo, é possível especificar que um delimitador de texto seja inserido antes dos dados a partir de mensagens individuais, usando os parâmetros **-sq**, **-sqdt** e **-sqdp** com o comando **fteCreateTransfer**.

Sobre esta tarefa

Nesse exemplo, existem quatro mensagens na fila `START_QUEUE`. Essa fila está no gerenciador de filas do agente de origem, `QM_NEPTUNE`. O delimitador de texto a ser inserido antes dos dados de cada mensagem pode ser expresso como uma sequência de caracteres literal Java, por exemplo: `\n\u002D\u002D\u002D\n`.

Procedimento

Digite o seguinte comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/output.txt
-t text -sqdt "\n\u002D\u002D\u002D\n" -sqdp prefix -sq START_QUEUE
```

O delimitador de texto é incluído no início dos dados a partir de cada uma das quatro mensagens em `START_QUEUE` pelo agente de origem, `AGENT_NEPTUNE`. Esses dados são gravados no arquivo de destino, `/out/output.txt`.

Conceitos relacionados

[“Transferindo Dados de Mensagens para Arquivos” na página 283](#)

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

Tarefas relacionadas

[“Configurando um agente para executar transferências de mensagem para arquivo” na página 284](#)

Por padrão, os agentes não podem executar transferências de mensagem para arquivo, ou de arquivo para mensagem. Para ativar essa função é necessário configurar a propriedade do agente `enableQueueInputOutput` para `true`.

[“Exemplo: Transferindo de uma Fila para um Único Arquivo” na página 285](#)

É possível especificar uma fila do IBM MQ como a origem de uma transferência de arquivos usando o parâmetro **-sq** com o comando **fteCreateTransfer**.

[“Exemplo: Transferindo um Grupo de Mensagens de uma Fila para uma Única Fila” na página 286](#)

É possível especificar um único grupo completo em uma fila do IBM MQ como a origem de uma transferência de arquivo usando os parâmetros **-sq** e **-sqgi** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador Binário Após os Dados a Partir de Cada Mensagem” na página 288](#)

Ao transferir em modo binário de uma fila de origem para um arquivo, é possível especificar que um delimitador binário seja inserido após os dados de mensagens individuais, usando os parâmetros **-sq**, **-sqdb** e **-sqdp** com o comando **fteCreateTransfer**.

[“Monitorando uma Fila e Usando Substituição de Variável” na página 249](#)

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

[“Exemplo: falhando uma transferência de mensagem para arquivo usando as propriedades de mensagem do IBM MQ” na página 292](#)

É possível fazer com que uma transferência de mensagem para arquivo falhe, configurando a propriedade de mensagem `usr.UserReturnCode` IBM MQ como um valor diferente de zero. Também é possível especificar informações complementares sobre o motivo da falha, configurando a propriedade de mensagem `usr.UserSupplement` IBM MQ.

Referências relacionadas

fteCreateTransfer: [iniciar uma nova transferência de arquivos](#)

Exemplo: Inserindo um Delimitador Binário Após os Dados a Partir de Cada Mensagem

Ao transferir em modo binário de uma fila de origem para um arquivo, é possível especificar que um delimitador binário seja inserido após os dados de mensagens individuais, usando os parâmetros **-sq**, **-sqdb** e **-sqdp** com o comando **fteCreateTransfer**.

Sobre esta tarefa

Nesse exemplo, existem três mensagens na fila `START_QUEUE`. Essa fila está no gerenciador de filas do agente de origem, `QM_NEPTUNE`. O delimitador binário a ser inserido após os dados a partir de cada mensagem deve ser expresso como uma lista separada por vírgula de bytes hexadecimais, por exemplo: `x34, xE7, xAE`.

Procedimento

Digite o seguinte comando:


```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file  
-sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

O delimitador binário é anexado aos dados a partir de cada uma das três mensagens em START_QUEUE pelo agente de origem, AGENT_NEPTUNE. Esses dados são gravados no arquivo de destino, /out/binary.file.

Conceitos relacionados

[“Transferindo Dados de Mensagens para Arquivos” na página 283](#)

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

Tarefas relacionadas

[“Configurando um agente para executar transferências de mensagem para arquivo” na página 284](#)

Por padrão, os agentes não podem executar transferências de mensagem para arquivo, ou de arquivo para mensagem. Para ativar essa função é necessário configurar a propriedade do agente enableQueueInputOutput para true.

[“Exemplo: Transferindo de uma Fila para um Único Arquivo” na página 285](#)

É possível especificar uma fila do IBM MQ como a origem de uma transferência de arquivos usando o parâmetro **-sq** com o comando **fteCreateTransfer**.

[“Exemplo: Transferindo um Grupo de Mensagens de uma Fila para uma Única Fila” na página 286](#)

É possível especificar um único grupo completo em uma fila do IBM MQ como a origem de uma transferência de arquivo usando os parâmetros **-sq** e **-sqgi** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador de Texto antes dos Dados a partir de cada Mensagem” na página 287](#)

Quando estiver transferindo em modo de texto de uma fila de origem para um arquivo, é possível especificar que um delimitador de texto seja inserido antes dos dados a partir de mensagens individuais, usando os parâmetros **-sq**, **-sqdt** e **-sqdp** com o comando **fteCreateTransfer**.

[“Monitorando uma Fila e Usando Substituição de Variável” na página 249](#)

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

[“Exemplo: falhando uma transferência de mensagem para arquivo usando as propriedades de mensagem do IBM MQ” na página 292](#)

É possível fazer com que uma transferência de mensagem para arquivo falhe, configurando a propriedade de mensagem `usr.UserReturnCode` IBM MQ como um valor diferente de zero. Também é possível especificar informações complementares sobre o motivo da falha, configurando a propriedade de mensagem `usr.UserSupplement` IBM MQ.

Referências relacionadas

[**fteCreateTransfer**: iniciar uma nova transferência de arquivos](#)

Monitorando uma Fila e Usando Substituição de Variável

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

Sobre esta tarefa

Nesse exemplo, o agente de origem é chamado de AGENT_VENUS, que se conecta ao QM_VENUS. A fila que o AGENT_VENUS monitora é chamada de START_QUEUE e está localizada no QM_VENUS. O agente sonda a fila a cada 30 minutos.

Quando um grupo completo de mensagens é gravado na fila, o monitor de tarefa envia o grupo de mensagens para um arquivo em um número de agentes de destinos, todos se conectando ao gerenciador de filas QM_MARS. O nome do arquivo para o qual o grupo de mensagens é transferido é definido pela IBM MQ propriedade de mensagens `usr.fileName` na primeira mensagem no grupo. O nome do agente para o qual o grupo de mensagens é enviado é definido pela IBM MQ propriedade de mensagem `usr.toAgent` na primeira mensagem no grupo. Se o cabeçalho `usr.toAgent` não estiver configurado, o valor padrão a ser usado para o agente de destino será AGENT_MAGENTA.

Ao especificar `useGroups="true"`, se você também não especificar `groupId="{GROUPID}"`, a transferência obterá somente a primeira mensagem na fila. Por exemplo, se estiver usando a substituição de variável para gerar o `fileName`, será possível que o conteúdo de `a.txt` não esteja correto. Isto ocorre porque `fileName` é gerado pelo monitor, mas a transferência realmente obtém uma mensagem que não é a que deve gerar o arquivo chamado `fileName`.

Procedimento

1. Crie o XML de tarefa que define a tarefa que o monitor executa quando for acionado.

```
<?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="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="{toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/{fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

As variáveis que são substituídas com os valores dos cabeçalhos da mensagem do IBM MQ são destacados em **negrito**. Esse XML de tarefa é salvo no arquivo `/home/USER1/task.xml`

2. Crie um monitor de recursos para monitorar a fila START_QUEUE.

Execute o seguinte comando:

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. Um usuário ou programa grava um grupo de mensagens na fila START_QUEUE.

A primeira mensagem nesse grupo tem o conjunto de propriedades de mensagens do IBM MQ a seguir:

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. O monitor é acionado quando o grupo completo é gravado. O agente substitui as propriedades de mensagem do IBM MQ no XML de tarefa.

Isso resulta na transformação da tarefa XML to:

```
<?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="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Resultados

A transferência definida pela tarefa XML é executada. O grupo completo de mensagens que são lidos a partir da START_QUEUE por AGENT_VENUS é gravado em um arquivo chamado /reports/larmer.rpt no sistema em que o AGENT_VIOLET está em execução.

Como proceder a seguir

Transferindo cada mensagem para um arquivo separado

Se desejar monitorar uma fila e fazer com que cada mensagem seja transferida para um arquivo separado, será possível usar uma técnica semelhante àquela descrita anteriormente neste tópico.

1. Crie o monitor conforme descrito anteriormente, especificando o parâmetro **-tr completeGroups** no comando **fteCreateMonitor ..**
2. No XML da tarefa, especifique o seguinte:

```
<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
```

No entanto, ao colocar as mensagens na fila de origem, não as coloque em um grupo do IBM MQ. Inclua propriedades de mensagem do IBM MQ em cada mensagem. Por exemplo, especifique a propriedade `usr.filename` com um valor de nome do arquivo exclusivo para cada mensagem. Isso efetivamente faz com que o Managed File Transfer Agent trate cada mensagem na fila de origem como um grupo separado.

Conceitos relacionados

[“Transferindo Dados de Mensagens para Arquivos” na página 283](#)

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

[“Customizando as tarefas do monitor de recurso do MFT com substituição de variável.” na página 244](#)

Quando as condições acionadoras de um monitor de recurso ativo são satisfeitas, a tarefa definida é chamada. Além de chamar a tarefa de transferência ou de comando sempre com o mesmo agente de destino ou o mesmo nome do arquivo de destino, também é possível modificar a definição da tarefa no tempo de execução. Faça isso inserindo nomes de variáveis no XML de definição de tarefa. Quando

o monitor determina que as condições do acionador são satisfeitas e que a definição da tarefa contém nomes de variáveis, ele substitui os nomes das variáveis pelos valores das variáveis e, em seguida, chama a tarefa.

O que Fazer se os Arquivos de Destino Criados por uma Transferência Iniciada por um Monitor de Recurso de Fila Contiverem Dados Incorretos

Tarefas relacionadas

“Configurando tarefas de monitor do MFT para iniciar comandos e scripts” na página 237

Monitores de recursos não estão limitados a executar transferências de arquivo como suas tarefas associadas. É também possível configurar o monitor para chamar outros comandos a partir do agente de monitoramento, incluindo programas executáveis, scripts Ant ou tarefas JCL. Para chamar comandos, edite o XML de definição de tarefa de monitoramento para incluir um ou mais elementos de comando nos parâmetros de chamada de comando correspondentes, como argumentos e propriedades.

“Exemplo: configurando um recurso do MFT” na página 243

É possível especificar uma fila do IBM MQ como o recurso a ser monitorado por um monitor de recurso usando o parâmetro `-mq` com o comando `fteCreateMonitor`.

Referências relacionadas

`fteCreateMonitor`: criar um monitor de recurso do MFT

Propriedades de mensagem do MQ lidas pelo MFT de mensagens em filas de origem

Exemplo: falhando uma transferência de mensagem para arquivo usando as propriedades de mensagem do IBM MQ

É possível fazer com que uma transferência de mensagem para arquivo falhe, configurando a propriedade de mensagem `usr.UserReturnCode` IBM MQ como um valor diferente de zero. Também é possível especificar informações complementares sobre o motivo da falha, configurando a propriedade de mensagem `usr.UserSupplement` IBM MQ.

Sobre esta tarefa

Neste exemplo, uma transferência está em andamento entre a fila `INPUT_QUEUE` e o arquivo `/home/user/output.file`.

Um usuário está criando mensagens e colocando-as na fila `INPUT_QUEUE`. O agente de origem está consumindo mensagens a partir da fila `INPUT_QUEUE` e enviando os dados de transferência para o agente de destino. O agente de destino está gravando esses dados no arquivo `/home/user/output.file`.

O usuário que grava mensagens na fila `INPUT_QUEUE` deseja parar a transferência que está em andamento e excluir quaisquer dados que já foram gravados na fila de destino.

Procedimento

1. O usuário grava uma mensagem na fila `INPUT_QUEUE` que tem o conjunto de propriedades de mensagem do IBM MQ a seguir:

```
usr.UserReturnCode=1
usr.UserSupplement="Cancelling transfer - sent wrong data."
```

2. O agente de origem lê as propriedades de mensagem do IBM MQ e para o processamento das mensagens na fila. O agente de destino exclui quaisquer dados de arquivo que foram gravados no diretório de destino.
3. O agente de origem envia uma mensagem de log de transferência para o gerenciador de filas de coordenação relatando a falha da transferência.

A mensagem contém as seguintes informações:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d312020202020202020207e970d4920008702" agentRole="sourceAgent"
```

```

        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="TransferLog.xsd"
        xmlns="">
<action time="2008-11-02T21:28:09.593Z">progress</action>
<sourceAgent agent="FTEAGENT" QMgr="QM1">
  <systemInfo architecture="x86" name="Windows 7"
    version="6.1 build 7601 Service Pack 1" />
</sourceAgent>
<destinationAgent agent="FTEAGENT" QMgr="QM1">
  <systemInfo architecture="x86" name="Windows 7"
    version="6.1 build 7601 Service Pack 1" />
</destinationAgent>
<originator>
  <hostName>reportserver.com</hostName>
  <userID>USER1</userID>
  <mqmdUserID>USER1 </mqmdUserID>
</originator>
<transferSet index="0" size="1"
  startTime="2008-11-02T21:28:09.281Z"
  total="1">
  <item mode="binary">
    <source>
      <queue>INPUT_QUEUE@QM1</queue>
    </source>
    <destination exist="error">
      <file>/home/user/output.file</file>
    </destination>
    <status resultCode="1">
      <supplement>Cancelling transfer - sent wrong data.</supplement>
    </status>
  </item>
</transferSet>
</transaction>

```

Conceitos relacionados

[“Transferindo Dados de Mensagens para Arquivos”](#) na página 283

O recurso de mensagem para arquivo do Managed File Transfer permite transferir dados de uma ou mais mensagens em uma fila do IBM MQ para um arquivo, um conjunto de dados (no z/OS) ou um espaço no arquivo de usuário. Se você tiver um aplicativo que cria ou processa mensagens do IBM MQ, será possível usar o recurso mensagem para arquivo do Managed File Transfer para transferir essas mensagens para um arquivo em qualquer sistema em sua rede do Managed File Transfer.

Tarefas relacionadas

[“Configurando um agente para executar transferências de mensagem para arquivo”](#) na página 284

Por padrão, os agentes não podem executar transferências de mensagem para arquivo, ou de arquivo para mensagem. Para ativar essa função é necessário configurar a propriedade do agente `enableQueueInputOutput` para `true`.

[“Exemplo: Transferindo de uma Fila para um Único Arquivo”](#) na página 285

É possível especificar uma fila do IBM MQ como a origem de uma transferência de arquivos usando o parâmetro **-sq** com o comando **fteCreateTransfer**.

[“Exemplo: Transferindo um Grupo de Mensagens de uma Fila para uma Única Fila”](#) na página 286

É possível especificar um único grupo completo em uma fila do IBM MQ como a origem de uma transferência de arquivo usando os parâmetros **-sq** e **-sqgi** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador de Texto antes dos Dados a partir de cada Mensagem”](#) na página 287

Quando estiver transferindo em modo de texto de uma fila de origem para um arquivo, é possível especificar que um delimitador de texto seja inserido antes dos dados a partir de mensagens individuais, usando os parâmetros **-sq**, **-sqdt** e **-sqdp** com o comando **fteCreateTransfer**.

[“Exemplo: Inserindo um Delimitador Binário Após os Dados a Partir de Cada Mensagem”](#) na página 288

Ao transferir em modo binário de uma fila de origem para um arquivo, é possível especificar que um delimitador binário seja inserido após os dados de mensagens individuais, usando os parâmetros **-sq**, **-sqdb** e **-sqdp** com o comando **fteCreateTransfer**.

[“Monitorando uma Fila e Usando Substituição de Variável”](#) na página 249

É possível monitorar uma fila e transferir as mensagens da fila monitorada para um arquivo usando o comando **fteCreateMonitor**. O valor de qualquer propriedade de mensagem do IBM MQ na primeira

mensagem a ser lida a partir da fila monitorada pode ser substituído na definição XML de tarefa e usado para definir o comportamento da transferência.

Referências relacionadas

[Propriedades de mensagem do MQ lidas pelo MFT de mensagens em filas de origem](#)

A ponte de protocolo

A ponte de protocolo permite que a rede do Managed File Transfer (MFT) acesse arquivos armazenados em um servidor de arquivos fora da rede do MFT, em seu domínio local ou um local remoto. Este servidor de arquivos pode usar os protocolos de rede FTP, FTPS ou SFTP. Cada servidor de arquivos precisa de pelo menos um agente dedicado. O agente dedicado é conhecido como o agente de ponte de protocolo. Um agente de ponte pode interagir com múltiplos servidores de arquivos.

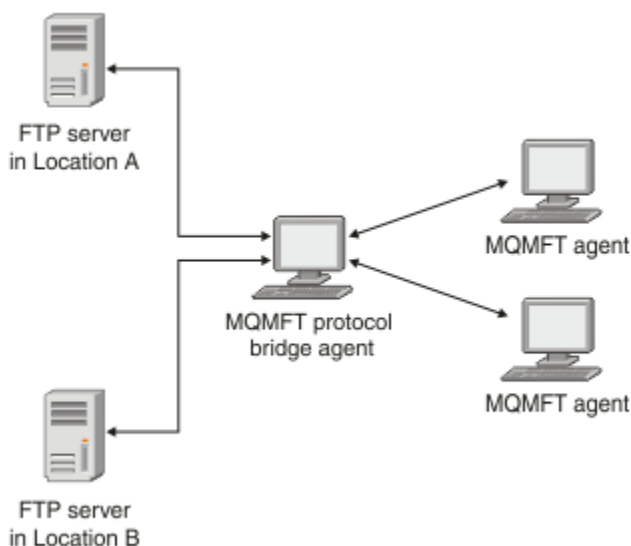
A ponte de protocolo está disponível como parte do componente Service do Managed File Transfer. É possível ter diversos agentes dedicados em um único sistema que execute o MFT que se conecta a servidores de arquivos diferentes.

É possível usar um agente de ponte de protocolo para transferir arquivos para diversos terminais simultaneamente. O MFT fornece um arquivo chamado `ProtocolBridgeProperties.xml` possível de editar para definir os diferentes servidores de arquivos de protocolo para os quais você deseja transferir arquivos. O comando **fteCreateBridgeAgent** inclui os detalhes do servidor de arquivos de protocolo padrão no `ProtocolBridgeProperties.xml`. Esse arquivo é descrito em [Formato de arquivo de propriedades da ponte de protocolo](#).

É possível usar o agente de ponte de protocolo para executar as seguintes ações:

- Fazer upload de arquivos da rede do MFT para um servidor remoto utilizando o FTP, FTPS ou SFTP.
- Fazer download de arquivos de um servidor remoto usando FTP, FTPS ou SFTP para a rede do MFT

Nota: O agente de ponte de protocolo apenas pode suportar servidores FTP, FTPS ou SFTP que permitem que os arquivos sejam acessados pelo seu caminho de arquivo absoluto. Se um caminho de arquivo relativo estiver especificado em uma solicitação de transferência, o agente de ponte de protocolo tentará converter o caminho relativo em um caminho de arquivo absoluto com base no diretório inicial usado para efetuar login no servidor de protocolo. Esses servidores de protocolo que permitem acesso a arquivos baseados apenas no diretório atual não são suportados pelo agente de ponte de protocolo.



O diagrama mostra dois servidores de FTP, em locais diferentes. Os servidores FTP estão sendo utilizados para a troca de arquivos com os agentes do Managed File Transfer. O agente de ponte de protocolo está entre os servidores FTP e o restante da rede do MFT e é configurado para se comunicar com os dois servidores FTP.

Certifique-se de que tenha outro agente na rede do MFT, além do agente de ponte de protocolo. O agente de ponte de protocolo é uma ponte apenas para o servidor FTP, FTPS ou SFTP e não grava arquivos transferidos para o disco local. Se desejar transferir arquivos para ou a partir do servidor FTP, FTPS ou SFTP, deve-se usar o agente de ponte de protocolo como o destino ou origem para a transferência de arquivos (representando o servidor FTP, FTPS ou SFTP) e outro agente padrão como a origem ou destino correspondente.

Quando você transferir arquivos usando a ponte de protocolo, a ponte deverá ter permissão de leitura no diretório de origem ou de destino que contém os arquivos que você deseja transferir. Por exemplo, se você deseja transferir arquivos do diretório `/home/fte/bridge` que possui permissões de execução (`d-x-x-x-x`), todas as transferências que você tentar deste diretório falharão com a seguinte mensagem de erro:

```
BFGBR0032E: Attempt to read filename from the protocol file server
has failed with server error 550. Failed to open file.
```

Configurando um Agente de Ponte de Protocolo

Um agente de ponte de protocolo é semelhante a um agente padrão do MFT. Crie um agente de ponte de protocolo usando o comando **fteCreateBridgeAgent**. É possível configurar um agente de ponte de protocolo usando o arquivo `ProtocolBridgeProperties.xml`, que é descrito em [Formato do arquivo de propriedades da ponte de protocolo](#). Se você estiver usando uma versão anterior, configure o agente usando as propriedades de ponte de protocolo específicas descritas em [Propriedades avançadas do agente: ponte de protocolo](#) e [Propriedades avançadas do agente: registro do agente de ponte de protocolo](#). Para todas as versões, também é possível configurar um mapeamento de credencial conforme descrito em [“Mapeando Credenciais para um Servidor de Arquivos”](#) na página 303. Depois de configurar um agente de ponte de protocolo para um determinado servidor de arquivos de protocolo, será possível então usar esse agente apenas para esse propósito.

Recuperação da ponte do protocolo

Se o agente de ponte do protocolo não puder se conectar ao servidor de arquivos por ele estar indisponível, todos os pedidos de transferência de arquivos serão enfileirados até que o servidor de arquivos se torne disponível. Se o agente de ponte do protocolo não conseguir se conectar ao servidor de arquivos por estar usando as credenciais incorretas, a transferência falhará e a mensagem de log de transferência refletirá esse erro. Se o agente de ponte de protocolo for encerrado por qualquer razão, todas as transferências de arquivos solicitadas serão retidas e continuarão quando a ponte de protocolo for reiniciada.

Durante a transferência do arquivo, os arquivos são tipicamente gravados como arquivos temporários no destino e são, em seguida, renomeados quando a transferência é concluída. Entretanto, se o destino da transferência for um servidor de arquivos do protocolo que esteja configurado como somente gravação (ou seja, os usuários podem fazer upload de arquivos para o servidor de arquivos do protocolo, mas não podem alterar esses arquivos transferidos por upload de maneira alguma; efetivamente, os usuário podem somente gravar uma vez), os arquivos transferidos são gravados diretamente no destino. Isto significa que se ocorrer um problema durante a transferência, os arquivos parcialmente gravados permanecerão no servidor de arquivos de protocolo de destino e o Managed File Transfer não poderá excluir ou editar estes arquivos. Neste caso, a transferência falhará.

Tarefas relacionadas

[“Exemplo: como configurar um agente de ponte de protocolo para usar credenciais de chave privada com um servidor UNIX SFTP”](#) na página 308

Este exemplo demonstra como é possível gerar e configurar o arquivo `ProtocolBridgeCredentials.xml`. Este é um exemplo típico e os detalhes podem variar de acordo com a plataforma, mas os princípios permanecem os mesmos.

[“Definindo propriedades para servidores de arquivos de protocolo usando o arquivo ProtocolBridgeProperties.xml”](#) na página 296

Defina as propriedades de um ou mais servidores de arquivos de protocolo para os quais e dos quais você deseja transferir arquivos usando o arquivo `ProtocolBridgeProperties.xml`, que é fornecido pelo Managed File Transfer no diretório de configuração do agente.

Referências relacionadas

[fteCreateBridgeAgent \(criar e configurar um agente de ponte de protocolo do MFT\)](#)

[“Mapeando Credenciais para um Servidor de Arquivos” na página 303](#)

Mapeie credenciais do usuário no Managed File Transfer para credenciais do usuário no servidor de arquivos usando a função de mapeamento de credencial padrão do agente de ponte de protocolo ou gravando sua própria saída de usuário. O Managed File Transfer fornece uma saída de usuário de amostra que executa o mapeamento de credencial do usuário.

[Interface ProtocolBridgeCredentialExit.java](#)

[Saída do usuário da credencial de ponte do protocolo de amostra](#)

[Suporte de Servidor FTPS pela Ponte de Protocolo](#)

Definindo propriedades para servidores de arquivos de protocolo usando o arquivo `ProtocolBridgeProperties.xml`

Defina as propriedades de um ou mais servidores de arquivos de protocolo para os quais e dos quais você deseja transferir arquivos usando o arquivo `ProtocolBridgeProperties.xml`, que é fornecido pelo Managed File Transfer no diretório de configuração do agente.

Sobre esta tarefa

O comando **`fteCreateBridgeAgent`** cria o arquivo `ProtocolBridgeProperties.xml` no diretório de configuração do agente `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`. O comando também cria uma entrada no arquivo para o servidor de arquivos de protocolo padrão, se um padrão tiver sido especificado quando o comando foi executado.

A mensagem BFGCL0392I dá a localização do arquivo `ProtocolBridgeProperties.xml`.

```
<?xml version="1.0" encoding="IBM-1047"?>
<!--
This ProtocolBridgeProperties.xml file determines the protocol servers that will be accessed by
the
MQMFT protocol bridge agent.

Each protocol server is defined using either a <tns:ftpServer>, <tns:ftpsServer>, or
<tns:sftpServer>
element - depending on the protocol used to communicate with the server. When the protocol
bridge agent participates in a managed file transfer it will determine which server to used
based on
the prefix (if any) present on the file path. For example a file path of 'server1:/home/user/
file.txt' would
be interpreted as a request to transfer /home/user/file.txt using 'server1'. The server name
is compared
to the 'name' attribute of each <tns:ftpServer>, <tns:ftpsServer> or <tns:sftpServer> element
in this
XML document and the first match is used to determine which protocol server the protocol bridge
agent will connect to. If no match is found then the managed file transfer operation will fail.

If a file path is not prefixed with a server name, for example '/home/user/file.txt' then this
XML
document can specify a default server to use for the managed file transfer. To specify a
default server use the <tns:defaultServer> element as the first element inside the
<tns:serverProperties>
element. The default server will be used whenever the protocol bridge agent participates in
a managed file transfer for file names which do not specify a prefix.

An optional <tns:limits> element can be specified within each server definition. This element
contains
attributes that govern the amount of resources used by each defined server.

An optional <tns:credentialsFile> element can be specified within each serverProperties
definition. This
element contains a path to a file containing credentials to be used when connecting to defined
servers.
```

An example `ProtocolBridgeProperties.xml` file is as follows:


```

<?xml version="1.0" encoding="UTF-8"?>
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
ProtocolBridgeProperties.xsd">

  <tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />

  <tns:defaultServer name="myFTPserver" />

  <tns:ftpServer name="myFTPserver" host="windows.hursley.ibm.com" port="1234"
platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  listFormat="unix" limitedWrite="false">

    <tns:limits maxListFileNames="100" maxListDirectoryLevels="999999999"
      maxReconnectRetry="2" reconnectWaitPeriod="10"
      maxSessions="60" socketTimeout="30" />

  </tns:ftpServer>

  <tns:ftpsServer name="myFTPSserver" host="unix.hursley.ibm.com" platform="unix"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false" ftpsType="explicit"
  trustStore="C:\FTE\keystores\myFTPSserver\FTPSKeyStore.jks"
trustStorePassword="password">

    <tns:limits maxReconnectRetry="10" connectionTimeout="10"/>

  </tns:ftpsServer>

  <tns:sftpServer name="mySFTPserver" host="windows.hursley.ibm.com" platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  limitedWrite="false">

    <tns:limits connectionTimeout="60"/>

  </tns:sftpServer>
</tns:serverProperties>

```

This example shows the outermost <tns:serverProperties> element which must exist for the document to be valid, an optional <tns:defaultServer> element, as well as definitions for an FTP, FTPS and SFTP server.

The attributes of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements determine the characteristics of the connection established to the server. These attributes correspond to the command line parameters for the 'fteCreateBridgeAgent' command.

The following attributes are valid for all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: name, host, port, platform, fileEncoding, limitedWrite and controlEncoding.

The following attributes are valid for the <tns:ftpServer> and <tns:ftpsServer> elements: timeZone, locale, listFormat, listFileRecentDateFormat, listFileOldDateFormat, and monthShortNames.

The following attributes are valid for the <tns:ftpServer> element only: passiveMode

The following attributes are valid for the <tns:ftpsServer> element only: ftpsType, trustStore, trustStorePassword, trustStoreType, keyStore, keyStorePassword, keyStoreType, ccc, protFirst, auth, and connectTimeout.

The following attributes are valid for the <tns:limits> element within all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: maxListFileNames, maxListDirectoryLevels, maxReconnectRetry, reconnectWaitPeriod, maxSessions and socketTimeout

```

-->
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
ProtocolBridgeProperties.xsd">

```

<!-- By default the location of the credentials file is in the home directory of the user that started the -->

```

    <!-- protocol bridge agent. If you wish to specify a different location use the
credentialsFile element to -->
    <!-- do this. For
example:
    <!-- <tns:credentialsFile path="/test/
ProtocolBridgeCredentials.xml"/>
-->

    <tns:defaultServer name="WINMVSCA.HURSLEY.IBM.COM" />
    <tns:ftpServer name="WINMVSCA.HURSLEY.IBM.COM" host="WINMVSCA.HURSLEY.IBM.COM"
platform="unix"
        timeZone="Europe/London" locale="en-GB" fileEncoding="US-ASCII"
        listFormat="unix" limitedWrite="false" />

    <!-- Define servers here -->
</tns:serverProperties>

```

O comando pode produzir a mensagem a seguir: BFGCL0532I:

Para que este agente funcione, é necessário criar manualmente um arquivo de credenciais adicional. Por padrão, este arquivo é chamado ProtocolBridgeCredentials.xml e está localizado no diretório inicial do usuário que inicia o agente. Por exemplo, se esse usuário iniciasse o agente, o local seria: \$HOME/ProtocolBridgeCredentials.xml

Se você usar um arquivo de credenciais:

1. Veja o texto a seguir para obter informações adicionais sobre como criar um.
2. O arquivo de credenciais deve estar em um diretório com permissões restritas. Por exemplo, não deve haver acesso de leitura para outros usuários.
3. Especifique o local do diretório para o arquivo de credenciais na variável de ambiente \$HOME para o ID do usuário do agente iniciado ou edite o arquivo ProtocolBridgeProperties.xml e especifique o local em:

```
<tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml"/>
```

Se desejar incluir servidores de protocolo não padrão adicionais, edite este arquivo para definir suas propriedades. Este exemplo inclui um servidor FTP adicional.

Nota: O agente de ponte de protocolo não suporta bloqueio de arquivo. Isso se deve ao fato de que o Managed File Transfer não suporta o mecanismo de bloqueio de arquivo em um servidor de arquivos.

Procedimento

1. Defina um servidor de arquivos de protocolo, inserindo as seguintes linhas no arquivo como um elemento filho de <tns:serverProperties>:

```

<tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234"
platform="windows"
        timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
        listFormat="unix" limitedWrite="false" >
<tns:limits maxListFileNames="10" maxListDirectoryLevels="500"/>

```

2. Em seguida, altere o valor dos atributos:

- name é o nome do servidor de arquivos de protocolo
- host é o nome do host ou endereço IP do servidor de arquivos de protocolo
- port é o número da porta do servidor de arquivos de protocolo
- platform é a plataforma na qual o servidor de arquivos de protocolo é executado
- timeZone é o fuso horário no qual o servidor de arquivos de protocolo é executado
- locale é o idioma usado no servidor de arquivos de protocolo
- fileEncoding é a codificação de caracteres do servidor de arquivos de protocolo
- listFormat é o formato da listagem de arquivos retornado do servidor de arquivos de protocolo
- limitedWrite determina se será seguido o modo padrão ao gravar em um servidor de arquivos, que deve criar um arquivo temporário e, em seguida, renomear esse arquivo quando a transferência

estiver concluída. Para um servidor de arquivos configurado como somente gravação, o arquivo é criado diretamente com seu nome final. O valor desta propriedade pode ser true ou false. O atributo `limitedWrite` e a propriedade de agente `doNotUseTempOutputFile` são usados juntos no caso de agentes de ponte de protocolo. Se deseja utilizar arquivos temporários, então você não deve configurar o valor de `doNotUseTempOutputFile` e deve configurar o valor de `limitedWrite` para false. Qualquer outra combinação de configurações significa que os arquivos temporários não serão usados.

- `maxListFileNames` é o número máximo de nomes coletados ao varrer um diretório no servidor de arquivos de protocolo em busca de nomes de arquivos.
- `maxListDirectoryLevels` é o número máximo de níveis de diretório para recursão ao varrer um diretório no servidor de arquivos de protocolo em busca de nomes de arquivos.

Para obter mais detalhes sobre esses atributos, incluindo se eles são obrigatórios ou opcionais e seus valores padrão, veja [Formato de arquivo de propriedades de ponte de protocolo](#).

Referências relacionadas

[Formato de Arquivo de Propriedades de Ponte de Protocolo](#)

[Expressões regulares usadas pelo MFT](#)

Consultando as propriedades do servidor de arquivos de protocolo: `ProtocolBridgePropertiesExit2`

Se tiver um grande número de servidores de arquivos de protocolo, poderá implementar a interface com `ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` para consultar propriedades de servidor de arquivos de protocolo que sejam referenciadas em transferências. É possível implementar essa interface na preferência para manter um arquivo do `ProtocolBridgeProperties.xml`.

Sobre esta tarefa

O Managed File Transfer fornece uma saída de usuário de amostra que consulta propriedades do servidor de arquivos de protocolo. Para obter mais informações, consulte [“Usando a saída de usuário de amostra para consultar as propriedades do servidor de arquivos de protocolo”](#) na página 300.

Qualquer saída de usuário que consulte propriedades de ponte de protocolo deve implementar a interface com `ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2`. Para obter mais informações, veja [Interface ProtocolBridgePropertiesExit.java](#).

É possível encadear diversas saídas de propriedades do servidor de protocolo juntas de uma maneira semelhante às outras saídas de usuário. As saídas são chamadas na ordem em que são especificadas usando a propriedade `protocolBridgePropertiesExitClasses` no arquivo de propriedades do agente. Todos os métodos de inicialização retornam separadamente se um ou mais retornar um valor de false, o agente não inicia. O erro é relatado no log de eventos do agente.

Apenas um resultado geral é retornado para os métodos `getProtocolServerProperties` de todas as saídas. Se o método retornar um objeto de propriedades como o código de resultado, este valor será o resultado retornado e os métodos `getProtocolServerProperties` das saídas subsequentes não serão chamados. Se o método retornar um valor nulo como o código de resultado, o método `getProtocolServerProperties` da próxima saída será chamado. Se não houver nenhuma saída subsequente, o resultado nulo será retornado. Um código de resultado geral de nulo é considerado como uma falha de consulta pelo agente de ponte de protocolo.

É recomendado usar a interface `ProtocolBridgePropertiesExit2.java`, mas para obter informações sobre a interface `ProtocolBridgePropertiesExit.java`, consulte [“Consultando as propriedades do servidor de arquivos de protocolo: ProtocolBridgePropertiesExit”](#) na página 301.

Para executar sua saída, conclua as seguintes etapas:

Procedimento

1. Compile a saída de usuário de propriedades do servidor de protocolo.
2. Crie um arquivo Java archive (JAR) que contenha a saída compilada e sua estrutura de pacotes.
3. Coloque o arquivo JAR contendo a classe de saída no diretório `exits` do agente de ponte de protocolo. Este diretório é encontrado no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.
4. Edite o arquivo de propriedades do agente de ponte de protocolo para incluir a propriedade `protocolBridgePropertiesExitClasses`. Para o valor desta propriedade, especifique uma lista separada por vírgula de classes que implementam uma saída de usuário de propriedades do servidor de ponte de protocolo. As classes de saída são chamadas na ordem em que estiverem especificadas nesta lista. Para obter mais informações, consulte [O arquivo MFT agent.properties](#).
5. Opcionalmente, é possível especificar a propriedade `protocolBridgePropertiesConfiguration`. O valor especificado para esta propriedade é transmitido como uma Sequência para o método `initialize()` das classes de saída especificadas por `protocolBridgePropertiesExitClasses`. Para obter mais informações, consulte [O arquivo MFT agent.properties](#).

Usando a saída de usuário de amostra para consultar as propriedades do servidor de arquivos de protocolo

O Managed File Transfer fornece uma saída de usuário de amostra que consulta propriedades do servidor de arquivos de protocolo.

Sobre esta tarefa

Uma saída de usuário de amostra que consulta as propriedades da ponte de protocolo é fornecida no diretório `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` e no tópico [Saída do usuário da ponte do protocolo de amostra](#).

A saída `SamplePropertiesExit2.java` lê um arquivo de propriedades que contém propriedades para servidores de protocolo. O formato de cada entrada no arquivo de propriedades é o seguinte:

```
serverName=type://host:port
```

O local do arquivo de propriedades é obtido da propriedade do agente de ponte de protocolo `protocolBridgePropertiesConfiguration`.

Para executar a saída de usuário de amostra, conclua as seguintes etapas:

Procedimento

1. Compile o arquivo `SamplePropertiesExit2.java`.
2. Crie um arquivo JAR que contenha a saída compilada e sua estrutura de pacotes.
3. Coloque o arquivo JAR no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits`.
4. Edite o arquivo `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` para conter a linha:

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. Crie um arquivo de propriedades da ponte de protocolo, por exemplo `protocol_bridge_properties.properties`, no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent`. Edite este arquivo para incluir entradas no formato:

```
serverName=type://host:port
```

6. Edite o arquivo `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` para conter a linha:

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

Deve-se usar o caminho absoluto para o arquivo `protocol_bridge_properties.properties`.

7. Inicie o agente de ponte de protocolo usando o comando **`fteStartAgent`**.

Conceitos relacionados

[“A ponte de protocolo” na página 294](#)

A ponte de protocolo permite que a rede do Managed File Transfer (MFT) acesse arquivos armazenados em um servidor de arquivos fora da rede do MFT, em seu domínio local ou um local remoto. Este servidor de arquivos pode usar os protocolos de rede FTP, FTPS ou SFTP. Cada servidor de arquivos precisa de pelo menos um agente dedicado. O agente dedicado é conhecido como o agente de ponte de protocolo. Um agente de ponte pode interagir com múltiplos servidores de arquivos.

Referências relacionadas

[Interface ProtocolBridgePropertiesExit.java](#)

[Saída de Usuário de Propriedades da Ponte de Protocolo de Amostra](#)

[O arquivo MFT `agent.properties`](#)

[`fteCreateBridgeAgent` \(criar e configurar um agente de ponte de protocolo do MFT\)](#)

Consultando as propriedades do servidor de arquivos de protocolo:

ProtocolBridgePropertiesExit

Se tiver um grande número de servidores de arquivos de protocolo, será possível implementar a interface com `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` para consultar propriedades de servidor de arquivos de protocolo que sejam referenciadas nas transferências.

Sobre esta tarefa

É possível implementar a interface

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` em vez de manter um arquivo `ProtocolBridgeProperties.xml`. Use a interface do `ProtocolBridgePropertiesExit2.java`. O método **`getCredentialLocation`** em `ProtocolBridgePropertiesExit2.java` usa o local padrão do arquivo `ProtocolBridgeCredentials.xml`, que é seu diretório inicial.

Qualquer saída de usuário que procure as propriedades da ponte de protocolo deve implementar a interface com `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit`:

```
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 ProtocolBridgePropertiesExit {

    /**
     * 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);

/**
 * 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);
}

```

É possível encadear diversas saídas de propriedades do servidor de protocolo juntas de uma maneira semelhante às outras saídas de usuário. As saídas são chamadas na ordem em que são especificadas usando a propriedade `protocolBridgePropertiesExitClasses` no arquivo de propriedades do agente. Todos os métodos de inicialização retornam separadamente se um ou mais retornar um valor de `false`, o agente não inicia. O erro é relatado no log de eventos do agente.

Apenas um resultado geral é retornado para os métodos `getProtocolServerProperties` de todas as saídas. Se o método retornar um objeto de propriedades como o código de resultado, este valor será o resultado retornado e os métodos `getProtocolServerProperties` das saídas subsequentes não serão chamados. Se o método retornar um valor nulo como o código de resultado, o método `getProtocolServerProperties` da próxima saída será chamado. Se não houver nenhuma saída subsequente, o resultado nulo será retornado. Um código de resultado geral de nulo é considerado como uma falha de consulta pelo agente de ponte de protocolo.

Procedimento

Para executar sua saída, conclua as seguintes etapas:

1. Compile a saída de usuário de propriedades do servidor de protocolo.
2. Crie um arquivo Java archive (JAR) que contenha a saída compilada e sua estrutura de pacotes.
3. Coloque o arquivo JAR contendo a classe de saída no diretório `exits` do agente de ponte de protocolo.

Este diretório é encontrado no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.

4. Edite o arquivo de propriedades do agente de ponte de protocolo para incluir a propriedade `protocolBridgePropertiesExitClasses`.

Para o valor desta propriedade, especifique uma lista separada por vírgula de classes que implementam uma saída de usuário de propriedades do servidor de ponte de protocolo. As classes de saída são chamadas na ordem em que estiverem especificadas nesta lista. Para obter mais informações, consulte [O arquivo MFT agent.properties](#).

5. Opcionalmente, é possível especificar a propriedade `protocolBridgePropertiesConfiguration`.

O valor especificado para esta propriedade é transmitido como uma Sequência para o método `initialize()` das classes de saída especificadas por `protocolBridgePropertiesExitClasses`. Para obter mais informações, consulte [O arquivo MFT agent.properties](#).

Mapeando Credenciais para um Servidor de Arquivos

Mapeie credenciais do usuário no Managed File Transfer para credenciais do usuário no servidor de arquivos usando a função de mapeamento de credencial padrão do agente de ponte de protocolo ou gravando sua própria saída de usuário. O Managed File Transfer fornece uma saída de usuário de amostra que executa o mapeamento de credencial do usuário.

Conceitos relacionados

[“A ponte de protocolo” na página 294](#)

A ponte de protocolo permite que a rede do Managed File Transfer (MFT) acesse arquivos armazenados em um servidor de arquivos fora da rede do MFT, em seu domínio local ou um local remoto. Este servidor de arquivos pode usar os protocolos de rede FTP, FTPS ou SFTP. Cada servidor de arquivos precisa de pelo menos um agente dedicado. O agente dedicado é conhecido como o agente de ponte de protocolo. Um agente de ponte pode interagir com múltiplos servidores de arquivos.

Tarefas relacionadas

[“Mapeando credenciais para um servidor de arquivos usando o arquivo ProtocolBridgeCredentials.xml” na página 303](#)

Mapeie credenciais do usuário no Managed File Transfer para credenciais do usuário no servidor de arquivos usando a função de mapeamento de credencial padrão do agente de ponte de protocolo. O Managed File Transfer fornece um arquivo XML possível de editar para incluir as informações de credenciais.

[“Mapeando Credenciais para um Servidor de Arquivos Usando Classes de Saída” na página 305](#)

Se não desejar usar a função de mapeamento de credencial padrão do agente de ponte de protocolo, será possível mapear credenciais do usuário no Managed File Transfer para usar credenciais no servidor de arquivos, gravando sua própria saída de usuário. Se você configurar saídas de usuário de mapeamento de credencial, elas assumirão o lugar da função de mapeamento de credencial padrão.

[“Exemplo: como configurar um agente de ponte de protocolo para usar credenciais de chave privada com um servidor UNIX SFTP” na página 308](#)

Este exemplo demonstra como é possível gerar e configurar o arquivo `ProtocolBridgeCredentials.xml`. Este é um exemplo típico e os detalhes podem variar de acordo com a plataforma, mas os princípios permanecem os mesmos.

Referências relacionadas

[Interface ProtocolBridgeCredentialExit.java](#)

[Saída do usuário da credencial de ponte do protocolo de amostra](#)

[O arquivo MFT agent.properties](#)

Mapeando credenciais para um servidor de arquivos usando o arquivo ProtocolBridgeCredentials.xml

Mapeie credenciais do usuário no Managed File Transfer para credenciais do usuário no servidor de arquivos usando a função de mapeamento de credencial padrão do agente de ponte de protocolo. O Managed File Transfer fornece um arquivo XML possível de editar para incluir as informações de credenciais.

Sobre esta tarefa

O arquivo `ProtocolBridgeCredentials.xml` deve ser criado manualmente pelo usuário. Por padrão, o local desse arquivo é o diretório inicial do usuário que iniciou o agente de ponte de protocolo, mas isso pode ser armazenado em qualquer lugar no sistema de arquivos acessível pelo agente. Para especificar

um local diferente, inclua o elemento `<credentialsFile>` no arquivo `ProtocolBridgeProperties.xml`. Por exemplo,

```
<tns:credentialsFile path="/example/path/to/ProtocolBridgeCredentials.xml"/>
```

Antes de usar um agente de ponte de protocolo, configure o mapeamento de credencial editando este arquivo para incluir informações do host, do usuário e de credencial. Para obter mais informações e amostras, veja [Formato de arquivo de credenciais de ponte de protocolo](#).

Procedimento

1. Edite a linha `<tns:server name="server name">` para mudar o valor do atributo de nome para o nome do servidor no arquivo `ProtocolBridgeProperties.xml`.

É possível usar o atributo de padrão para especificar que você usou um nome do servidor que contém curingas ou expressões regulares. Por exemplo,

```
<tns:server name="serverA*" pattern="wildcard">
```

2. Insira informações de ID e de credenciais do usuário no arquivo como elementos filhos de `<tns:server>`.

É possível inserir um ou muitos dos seguintes elementos no arquivo:

- Se o servidor de arquivos de protocolo for um servidor FTP, FTPS ou SFTP, será possível usar senhas para autenticar o usuário que solicita a transferência. Inclua as seguintes linhas no arquivo:

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  serverPassword="Server Password">
</tns:user>
```

Depois, mude o valor dos atributos.

- `name` é uma expressão regular Java para corresponder o ID do usuário MQMD associado à solicitação de transferência do MFT
- `serverUserId` é o valor que é passado ao servidor de arquivos de protocolo como o ID do usuário de login. Se o atributo `serverUserId` não for especificado, o ID do usuário MQMD associado à solicitação de transferência do MFT será usado como alternativa
- `serverPassword` é a senha associada ao `serverUserId`.

O atributo `name` pode conter uma expressão regular Java. O mapeador de credenciais tenta corresponder o ID do usuário MQMD da solicitação de transferência do MFT a essa expressão regular. O agente de ponte de protocolo tenta corresponder o ID do usuário do MQMD à expressão regular no atributo de nome dos elementos `<tns:user>` na ordem em que o elemento existe no arquivo. Quando uma correspondência é localizada, o agente de ponte de protocolo não procura mais correspondências. Se for localizada uma correspondência, os valores `serverUserId` e `serverPassword` correspondentes serão transmitidos para o servidor de arquivos de protocolo como o ID do usuário e senha de login. As correspondências do ID do usuário MQMD fazem distinção entre maiúsculas e minúsculas.

- Se um servidor de arquivos de protocolo for um servidor SFTP, será possível usar chaves públicas e privadas para autenticar a solicitação do usuário da transferência. Insira as seguintes linhas no arquivo e altere o valor dos atributos. O elemento `<tns:user>` pode conter um ou vários elementos `<tns:privateKey>`.

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  hostKey="Host Key">
  <tns:privateKey associationName="association"
    keyPassword="Private key password">
    Private key file text
```




```
</tns:privateKey>  
</tns:user>
```

- name é uma expressão regular Java para corresponder o ID do usuário MQMD associado à solicitação de transferência do MFT
- serverUserId é o valor que é passado ao servidor de arquivos de protocolo como o ID do usuário de login. Se o atributo serverUserId não for especificado, o ID do usuário MQMD associado à solicitação de transferência do MFT será usado como alternativa
- hostKey é a chave esperada retornada pelo servidor ao efetuar logon
- key é a chave privada do serverUserId
- keyPassword é a senha da chave para gerar chaves públicas
- associationName é um valor usado para identificação para propósitos de rastreamento e criação de logs

O atributo name pode conter uma expressão regular Java. O mapeador de credenciais tenta corresponder o ID do usuário MQMD da solicitação de transferência do MFT a essa expressão regular. O agente de ponte de protocolo tenta corresponder o ID do usuário do MQMD à expressão regular no atributo de nome dos elementos <tns:user> na ordem em que o elemento existe no arquivo. Quando uma correspondência é localizada, o agente de ponte de protocolo não procura mais correspondências. Se for localizada uma correspondência, os valores serverUserId e key correspondentes serão usados para autenticar o usuário do MFT com o servidor de arquivos de protocolo. As correspondências do ID do usuário MQMD fazem distinção entre maiúsculas e minúsculas.

Para obter informações adicionais sobre como usar chaves privadas com um agente de ponte de protocolo, consulte [“Exemplo: como configurar um agente de ponte de protocolo para usar credenciais de chave privada com um servidor UNIX SFTP”](#) na página 308.

Nota: 

Quando a solicitação de transferência for gravada na fila de comandos, o ID do usuário MQMD poderá ser convertido em maiúsculas, se a fila de comandos do agente de origem estiver em um sistema z/OS ou IBM i. Como resultado, o ID de usuário do MQMD para o mesmo usuário de origem deverá chegar na saída de credenciais igual ao original ou convertido para letras maiúsculas, dependendo do agente de origem especificado na solicitação de transferência. A saída do mapeamento de credencial padrão executa correspondências com distinção entre maiúsculas e minúsculas em relação ao ID do usuário MQMD, que você pode precisar permitir no arquivo de mapeamento.

Referências relacionadas

[Formato de Arquivo de Credenciais de Ponte de Protocolo](#)

[Formato de Arquivo de Propriedades de Ponte de Protocolo](#)

[Expressões regulares usadas pelo MFT](#)

Mapeando Credenciais para um Servidor de Arquivos Usando Classes de Saída

Se não desejar usar a função de mapeamento de credencial padrão do agente de ponte de protocolo, será possível mapear credenciais do usuário no Managed File Transfer para usar credenciais no servidor de arquivos, gravando sua própria saída de usuário. Se você configurar saídas de usuário de mapeamento de credencial, elas assumirão o lugar da função de mapeamento de credencial padrão.

Sobre esta tarefa

O Managed File Transfer fornece uma saída de usuário de amostra que executa o mapeamento de credencial do usuário. Para obter mais informações, consulte [“Usando a saída de usuário da credencial de ponte do protocolo de amostra”](#) na página 307.

Uma saída de usuário para mapeamento de credenciais de ponte de protocolo deve implementar uma das seguintes interfaces:

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`, que permite que um agente de ponte de protocolo transfira arquivos para e a partir de um servidor de arquivos de protocolo padrão
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`, que permite transferir arquivos para e a partir de diversos terminais.

A interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` contém a mesma função que `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` e também inclui função estendida. Para obter mais informações, veja [ProtocolBridgeCredentialExit.java interface](#) e [ProtocolBridgeCredentialExit2.java interface](#).

As saídas de credenciais podem ser encadeadas juntas de maneira semelhante a outras saídas de usuário. As saídas são chamadas na ordem em que são especificadas usando a propriedade `protocolBridgeCredentialConfiguration` no arquivo de propriedades do agente. Todos os métodos de inicialização retornam separadamente se um ou mais retornar um valor de `false`, o agente não inicia. O erro é relatado no log de eventos do agente.

Apenas um resultado geral é retornado para os métodos `mapMQUserId` de todas as saídas, da seguinte forma:

- Se o método retornar um valor de `USER_SUCCESSFULLY_MAPPED` ou `USER_DENIED_ACCESS` como o código de resultado, este valor será o resultado retornado e os métodos `mapMQUserId` das saídas subsequentes não serão chamados.
- Se o método retornar um valor de `NO_MAPPING_FOUND` como o código do resultado, o método `mqMQUserId` da próxima saída é chamado.
- Se não existir nenhuma saída subsequente, o resultado `NO_MAPPING_FOUND` é retornado.
- Um código de resultado geral igual a `USER_DENIED_ACCESS` ou `NO_MAPPING_FOUND` é considerado como uma falha na transferência pelo agente de ponte.

Para executar sua saída, conclua as seguintes etapas:

Procedimento

1. Compile a saída de usuário da credencial de ponte do protocolo.
2. Crie um arquivo Java archive (JAR) que contenha a saída compilada e sua estrutura de pacotes.
3. Coloque o arquivo JAR que contém a classe de saída no diretório `exits` do agente da ponte. O diretório está no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.
4. Edite o arquivo de propriedades do agente de ponte de protocolo para incluir a propriedade `protocolBridgeCredentialExitClasses`. Para o valor desta propriedade, especifique uma lista separada por vírgula de classes que implementam uma rotina de saída de credencial de ponte de protocolo. As classes de saída são chamadas na ordem em que estiverem especificadas nesta lista. Para obter mais informações, consulte [O arquivo MFT agent.properties](#).
5. Edite o arquivo de propriedades do agente de ponte de protocolo para incluir:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

O arquivo `agent.properties` para um agente está em seu diretório `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name`.

Se você alterar o arquivo `agent.properties`, deve-se reiniciar o agente para captar as alterações.

6. É possível, como opção, especificar a propriedade `protocolBridgeCredentialConfiguration`. O valor especificado para esta propriedade é transmitido como um objeto de Sequência para o método `initialize()` das classes de saída especificadas por `protocolBridgeCredentialExitClasses`. Para obter mais informações, consulte [O arquivo MFT agent.properties](#).
7. Inicie o agente de ponte de protocolo com o comando **`fteStartAgent`**.

Usando a saída de usuário da credencial de ponte do protocolo de amostra

O Managed File Transfer fornece uma saída de usuário de amostra que executa o mapeamento de credencial do usuário.

Sobre esta tarefa

Uma saída de credencial de ponte de protocolo de amostra é fornecida no diretório `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` e no tópico [Saída do usuário credencial da ponte de protocolo de amostra](#). Essa amostra se baseia na interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`.

A saída `SampleCredentialExit.java` lê um arquivo de propriedades que mapeia os IDs de usuários do MQMD associados a solicitações de transferência para IDs de usuários do servidor e senhas de servidores. O local do arquivo de propriedades é obtido da propriedade do agente de ponte de protocolo `protocolBridgeCredentialConfiguration`.

Para executar a saída de usuário de amostra, conclua as seguintes etapas:

Procedimento

1. Compile o arquivo `SampleCredentialExit.java`.
2. Crie um arquivo JAR que contenha a saída compilada e sua estrutura de pacotes.
3. Coloque o arquivo JAR no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits`.
4. Edite o arquivo `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` para conter a linha:

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

5. Edite o arquivo de propriedades do agente de ponte de protocolo para incluir:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_n
ame\exits\SampleCredentialExit.jar
```

O arquivo `agent.properties` para um agente está em seu diretório `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name`.

Se você alterar o arquivo `agent.properties`, deve-se reiniciar o agente para captar as alterações.

6. Crie um arquivo de propriedades de credencial (`credentials.properties`) no diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` e edite-o para incluir entradas no formato:

```
mqUserId=serverUserId,serverPassword
```

7. Edite o arquivo `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` para conter a linha:

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/
config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

Deve-se usar o caminho absoluto para o arquivo `credentials.properties`.

8. Inicie o agente de ponte de protocolo usando o comando **fteStartAgent**.

Conceitos relacionados

[“A ponte de protocolo” na página 294](#)

A ponte de protocolo permite que a rede do Managed File Transfer (MFT) acesse arquivos armazenados em um servidor de arquivos fora da rede do MFT, em seu domínio local ou um local remoto. Este servidor

de arquivos pode usar os protocolos de rede FTP, FTPS ou SFTP. Cada servidor de arquivos precisa de pelo menos um agente dedicado. O agente dedicado é conhecido como o agente de ponte de protocolo. Um agente de ponte pode interagir com múltiplos servidores de arquivos.

Referências relacionadas

[Interface ProtocolBridgeCredentialExit.java](#)

[Interface ProtocolBridgeCredentialExit2.java](#)

[Saída do usuário da credencial de ponte do protocolo de amostra](#)

O arquivo MFT `agent.properties`

`fteCreateBridgeAgent` (criar e configurar um agente de ponte de protocolo do MFT)

Exemplo: como configurar um agente de ponte de protocolo para usar credenciais de chave privada com um servidor UNIX SFTP

Este exemplo demonstra como é possível gerar e configurar o arquivo `ProtocolBridgeCredentials.xml`. Este é um exemplo típico e os detalhes podem variar de acordo com a plataforma, mas os princípios permanecem os mesmos.

Sobre esta tarefa

Procedimento

1. Gere uma chave pública e privada para ser usada para autenticar com o servidor SFTP.

Por exemplo, em um sistema host do Linux, é possível usar a ferramenta **ssh-keygen**, fornecida como parte do pacote 'openssh', para criar o par de chaves pública / privada

Por padrão, sem argumentos, o comando **ssh-keygen** solicita uma localização e passphrase para os dois arquivos de chaves, cujo padrão é os nomes:

```
id_rsa      <-- Private key
id_rsa.pub  <-- Public key
```



Atenção: Se você estiver usando o comando **ssh-keygen** de uma versão recente do OpenSSH, como a fornecida com o RHEL 8, o formato de chave usado não será compatível com o agente de ponte de protocolo e as tentativas de transferência para o servidor SFTP falharão com a mensagem:

```
BFGBR0216E: Authentication to protocol server 'sftp.host.address' failed
because of invalid private key.
```

Para criar uma chave privada compatível com essas versões mais recentes do OpenSSH, especifique o formato de chave com o seguinte argumento para o comando **ssh-keygen**:

```
ssh-keygen -m PEM
```

O conteúdo da chave privada `id_rsa` tem a primeira e a última linha de:

```
-----BEGIN RSA PRIVATE KEY-----
.....
-----END RSA PRIVATE KEY-----
```

que é compatível com o agente de ponte de protocolo.

2. Copie todo o conteúdo do arquivo `id_rsa.pub` para o arquivo `~/.ssh/authorized_keys` do usuário SFTP no servidor SFTP.

Certifique-se de que as permissões de arquivo neste arquivo e no diretório `~/ .ssh` estejam configuradas apropriadamente para que o servidor SFTP permita a autenticação de chave. Essas permissões são tipicamente:

```
~/ .ssh           Mode 700
~/ .ssh/authorized_keys Mode 600
```

3. O Managed File Transfer requer uma impressão digital ssh do host gerada usando o algoritmo MD5. Execute um dos comandos a seguir para obter a impressão digital ssh do host do servidor SFTP.

- Para o Red Hat® Enterprise Linux versão 6.x e abaixo e o Linux Ubuntu 14.04, execute o comando a seguir:

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
```

- A partir do Red Hat Enterprise Linux 7.x, Linux Ubuntu 16.04 e SuSE Linux 12.4, o comando `ssh-keygen` gera, por padrão, a impressão digital SSH utilizando o algoritmo SHA256. Para gerar a impressão digital ssh usando o algoritmo MD5, execute o comando a seguir:

```
ssh-keygen -l -E MD5 -f /etc/ssh/ssh_host_rsa_key.pub
```

A saída do comando será semelhante ao exemplo a seguir:

```
2048 MD5:64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 no comment (RSA)
```

Extraia a porção hexadecimal apenas da saída para usar como o `hostKey` no arquivo `ProtocolBridgeCredentials.xml` (veja a etapa “4” na página 309). Portanto, neste exemplo, você extrairia `64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27`.

4. No sistema de agente de ponte de protocolo, edite o arquivo `ProtocolBridgeCredentials.xml`. Substitua os valores mostrados em *itálico* do exemplo mostrado pelos seus próprios valores:

```
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
ProtocolBridgeCredentials.xsd">
  <tns:agent name="Agent_name">
  <tns:server name="SFTP_name">
  <tns:user name="mq_User_ID" serverUserId="SFTP_user_ID"
    hostKey="ssh_host_finger">
  <tns:privateKey associationName="name" keyPassword="pass_phrase">
    Complete contents of the id_rsa file including the entries
    -----BEGIN RSA PRIVATE KEY-----
    -----END RSA PRIVATE KEY-----
  </tns:privateKey>
</tns:user>
</tns:server>
</tns:agent>
</tns:credentials>
```

em que:

- *Agent_name* é o nome do agente de ponte de protocolo.
- *SFTP_host_name* é o nome do servidor SFTP conforme mostrado no arquivo `ProtocolBridgeProperties.xml`.
- *mq_User_ID* é o ID do usuário do MQMD associado à solicitação de transferência.
- *SFTP_user_ID* é o ID do usuário SFTP como usado na etapa 2. Ele é o valor transmitido para o servidor SFTP como ID do usuário de login.
- *ssh_host_finger* é a impressão digital coletada na etapa 3.

- *name* é o nome que pode ser especificado para ser usado para propósitos de rastreamento e de criação de log.
- *pass_phrase* é o passphrase fornecido no ssh-keygen na etapa 1.
- *Conteúdo completo do arquivo id_rsa* é o conteúdo completo do arquivo `id_rsa` gerado da etapa 1. Para evitar um erro de conexão, certifique-se de incluir as duas entradas a seguir:

```
-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----
```

É possível incluir chaves adicionais duplicando o elemento `<tns:privatekey>`.

5. Inicie o agente de ponte de protocolo se o agente ainda não estiver iniciado. Alternativamente, o agente da ponte de protocolo pesquisa periodicamente o arquivo `ProtocolBridgeCredentials.xml` e captar as alterações.

Referências relacionadas

[Formato de Arquivo de Credenciais de Ponte de Protocolo](#)

[fteCreateBridgeAgent \(criar e configurar um agente de ponte de protocolo do MFT\)](#)

[O arquivo MFT `agent.properties`](#)

[“Mapeando Credenciais para um Servidor de Arquivos” na página 303](#)

Mapeie credenciais do usuário no Managed File Transfer para credenciais do usuário no servidor de arquivos usando a função de mapeamento de credencial padrão do agente de ponte de protocolo ou gravando sua própria saída de usuário. O Managed File Transfer fornece uma saída de usuário de amostra que executa o mapeamento de credencial do usuário.

Configurando uma Ponte de Protocolo para um Servidor FTPS

Configure um servidor FTPS de maneira semelhante à configuração de um servidor FTP: crie um agente de ponte para o servidor, defina as propriedades de servidor e mapeie credenciais do usuário.

Sobre esta tarefa

Para configurar um servidor FTPS, conclua as seguintes etapas:

Procedimento

1. Crie um agente de ponte de protocolo para o servidor FTPS usando o comando **`fteCreateBridgeAgent`**. Os parâmetros aplicáveis ao FTP também são aplicáveis ao FTPS, mas há também três parâmetros necessários específicos do FTPS:
 - a) O parâmetro **`-bt`**. Especifique FTPS como o valor deste parâmetro.
 - b) O parâmetro **`-bts`** para o arquivo de armazenamento confiável. O comando supõe que somente a autenticação de servidor é necessária e que deve-se especificar o local do arquivo de armazenamento confiável.

O formato explícito do protocolo FTPS é configurado pelo comando **`fteCreateBridgeAgent`** por padrão, mas é possível configurar o formato implícito alterando o arquivo de propriedades da ponte de protocolo. A ponte de protocolo sempre se conecta a servidores FTPS em modo passivo.

Para obter mais informações sobre o comando **`fteCreateBridgeAgent`**, veja [fteCreateBridgeAgent \(criar e configurar um agente de ponte de protocolo do MFT\)](#).

Se precisar de instruções sobre como criar arquivos de armazenamento confiável, consulte as informações sobre a [keytool](#) na documentação da [keytool Oracle](#).

2. Defina as propriedades do servidor FTPS dentro de um elemento `<ftpsServer>` no arquivo de propriedades da ponte de protocolo: `ProtocolBridgeProperties.xml`. Para obter mais informações, consulte [“Definindo propriedades para servidores de arquivos de protocolo usando o arquivo ProtocolBridgeProperties.xml” na página 296](#). Também é possível ativar a autenticação de

cliente editando o arquivo de propriedades da ponte de protocolo. Para obter detalhes de todas as opções de configuração, veja [Formato de arquivo de propriedades da ponte de protocolo](#).

3. Mapeie credenciais de usuário no Managed File Transfer para credenciais de usuário no servidor FTPS usando a função de mapeamento de credencial padrão do agente de ponte de protocolo ou gravando sua própria saída de usuário. Para obter mais informações, consulte [“Mapeando Credenciais para um Servidor de Arquivos”](#) na página 303.
4. Por padrão, o arquivo de armazenamento confiável é configurado como tendo o formato JKS; se desejar alterar o formato, edite o arquivo de propriedades da ponte de protocolo.

Exemplo

Uma entrada de exemplo para um servidor FTPS no arquivo de propriedades de ponte de protocolo é mostrada como a seguir:

```
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
  <tns:defaultServer name="ftpserver.mycompany.com" />

  <tns:ftpsServer name="ftpserver.mycompany.com" host="ftpserver.mycompany.com" port="990"
  platform="windows"
  timeZone="Europe/London" locale="en_US" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false"
  trustStore="c:\mydirec\truststore.jks" />

  <!-- Define servers here -->
</tns:serverProperties>
```

Como proceder a seguir

Para obter informações sobre as partes do protocolo FTPS que são suportadas e que não são suportadas, veja [Suporte do servidor FTPS pela ponte de protocolo](#).

Conceitos relacionados

[“A ponte de protocolo”](#) na página 294

A ponte de protocolo permite que a rede do Managed File Transfer (MFT) acesse arquivos armazenados em um servidor de arquivos fora da rede do MFT, em seu domínio local ou um local remoto. Este servidor de arquivos pode usar os protocolos de rede FTP, FTPS ou SFTP. Cada servidor de arquivos precisa de pelo menos um agente dedicado. O agente dedicado é conhecido como o agente de ponte de protocolo. Um agente de ponte pode interagir com múltiplos servidores de arquivos.

Tarefas relacionadas

[“Mapeando credenciais para um servidor de arquivos usando o arquivo ProtocolBridgeCredentials.xml”](#) na página 303

Mapeie credenciais do usuário no Managed File Transfer para credenciais do usuário no servidor de arquivos usando a função de mapeamento de credencial padrão do agente de ponte de protocolo. O Managed File Transfer fornece um arquivo XML possível de editar para incluir as informações de credenciais.

[“Definindo propriedades para servidores de arquivos de protocolo usando o arquivo ProtocolBridgeProperties.xml”](#) na página 296

Defina as propriedades de um ou mais servidores de arquivos de protocolo para os quais e dos quais você deseja transferir arquivos usando o arquivo `ProtocolBridgeProperties.xml`, que é fornecido pelo Managed File Transfer no diretório de configuração do agente.

Referências relacionadas

[fteCreateBridgeAgent](#) (criar e configurar um agente de ponte de protocolo do MFT)

[Formato de Arquivo de Credenciais de Ponte de Protocolo](#)

[Formato de Arquivo de Propriedades de Ponte de Protocolo](#)

[Suporte de Servidor FTPS pela Ponte de Protocolo](#)

Cenários e exemplos para limitar o número de transferências de arquivos para servidores de arquivos individuais

Como o agente de ponte de protocolo revisado trabalha com os atributos **maxActiveDestinationTransfers** e **failTransferWhenCapacityReached**, juntamente com alguns exemplos.

Cenários que mostram o funcionamento do agente de ponte de protocolo com base no valor de **maxActiveDestinationTransfers**

Cenário 1

O arquivo `ProtocolBridgeProperties.xml` para um agente de ponte de protocolo contém duas definições de servidores de arquivos:

- Você não configurou o atributo global **maxActiveDestinationTransfers**.
- Você não configurou o atributo **maxActiveDestinationTransfers** em `fileServerA` e `fileServerB`.
- Você configurou o atributo **maxDestinationTransfers** do agente de ponte de protocolo como o valor padrão.

Se você tiver configurado o atributo **maxDestinationTransfers** do agente de ponte de protocolo com o valor padrão de 25, então:

- O agente de destino começa a processar duas transferências gerenciadas para `fileServerA`.
- Ambas as transferências são concluídas.

Neste momento, o cliente percebe que `fileServerA` falhou e configura os valores a seguir para `fileServerA` no arquivo `ProtocolBridgeProperties.xml`:

```
maxActiveDestinationTransfers = 0  
failTransferWhenCapacityReached =true
```

- Outra transferência chega para `fileServerA` e alguns para `fileServerB`:

Com base nas propriedades configuradas na etapa anterior, a transferência gerenciada para `fileServerA` é rejeitada e marcada como com falha, enquanto as transferências para `fileServerB` são manipuladas no fluxo existente padrão.

- Após algum tempo, o cliente descobre que o `fileServerA` está em execução novamente, portanto, o cliente remove ou comenta o valor incluído anteriormente no `ProtocolBridgeProperties.xml`. Uma nova transferência gerenciada chega para `fileServerA` e é manipulada no fluxo existente padrão.

Cenário 2

- Você configurou o atributo **maxActiveDestinationTransfers** para um servidor de arquivos e não configurou o atributo **failTransferWhenCapacityReached**.
- O agente de ponte de protocolo está agindo como o agente de destino para esse número de transferências gerenciadas para o servidor de arquivos.
- O valor do atributo **maxActiveDestinationTransfers** é reduzido em 1.

O agente de ponte de protocolo atualiza dinamicamente sua configuração e configura **maxActiveDestinationTransfers** com o novo valor enquanto ele ainda está ativo. As transferências gerenciadas em andamento não são afetadas por essa atualização e têm permissão para serem concluídas.

Cenário 3

O arquivo ProtocolBridgeProperties.xml para um agente de ponte de protocolo contém duas definições de servidor de arquivo:

- Você não configurou o atributo global **maxActiveDestinationTransfers**.
- Você não configurou o atributo **failTransferWhenCapacityReached**
- Você configurou o **maxActiveDestinationTransfers** para 1 em fileServerA.
- Você não configurou o atributo **maxActiveDestinationTransfers** em fileServerB

Se o agente de ponte de protocolo tiver o atributo **maxDestinationTransfers** configurado como 5:

- O número máximo de transferências de destino ativas do agente de ponte de protocolo para o fileServerA é 1 (embora o agente de destino tenha cinco slots de transferência de destino, apenas um pode ser usado para transferências gerenciadas para fileServerA).

Isso é útil quando fileServerA falha. Quando fileServerA estiver em execução novamente, o valor de **maxActiveDestinationTransfers** poderá ser aumentado para 5 para permitir a capacidade total das transferências de destino permitidas.

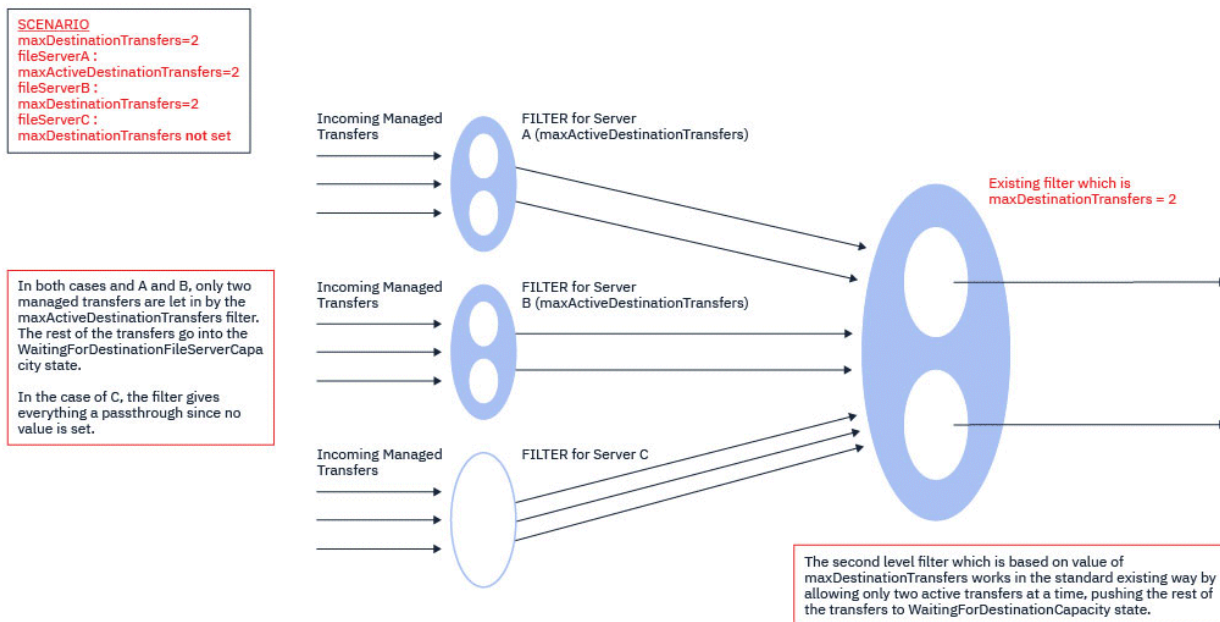
- O número máximo de transferências de destino ativas do agente de ponte de protocolo para fileServerB é 5.

Como o **maxActiveDestinationTransfers** não está configurado para esse servidor de arquivos, o agente de ponte de protocolo pode usar todos os seus cinco slots de transferência de destino para transferências gerenciadas nele.

Cenário 4

No diagrama a seguir:

- Você configurou o atributo **maxDestinationTransfers** como 2 no arquivo agent.properties
- Você configurou o **maxActiveDestinationTransfers** como 2 em fileServerA.
- Você configurou o atributo **maxActiveDestinationTransfers** como 2 em fileServerB
- Você não configurou o atributo **maxActiveDestinationTransfers** em fileServerC



Como mostra o diagrama, os atributos **maxActiveDestinationTransfers** e **maxDestinationTransfers** são independentes entre si.

Os valores para **maxActiveDestinationTransfers** para cada um dos servidores são verificados. Com base nesse valor, as transferências têm permissão para continuar adiante ou são enviadas por push para o estado **WaitingForDestinationFileServerCapacity**.

As transferências que são permitidas, então, passam pelo fluxo padrão existente de verificação com relação a **maxDestinationTransfers**.

Cenário 5



Atenção: É necessário ter cuidado ao configurar os valores dos atributos **maxActiveDestinationTransfers**, pois deve-se manter em mente o valor do atributo **maxDestinationTransfers**.

Se você não fizer isso, uma situação conforme descrito no texto a seguir poderá ocorrer:

- Você não configurou um valor para o atributo global **maxActiveDestinationTransfers**
- Você configurou um valor de **maxDestinationTransfers= 2** no arquivo `agent.properties`.
- Você configurou um valor de **maxActiveDestinationTransfers= 2** em `fileServerA`.
- Você não configurou um valor para **maxActiveDestinationTransfers** on `fileServerB`.

Suponha que a sequência de eventos a seguir ocorra:

- O agente de ponte de protocolo recebe uma solicitação para transferir um arquivo para `fileServerA`. O agente de ponte de protocolo não está fazendo nada atualmente, portanto, ele aceita essa solicitação de transferência gerenciada.

Os slots de transferência agora se parecem com isto:

- Destination Transfers: 1
- Destination Transfers for `fileServerA`: 1
- Destination transfers for `fileServerB`: 0

- Agora, o agente de ponte de protocolo recebe outra solicitação para agir como o agente de destino para uma transferência gerenciada envolvendo `fileServerA`. Mais uma vez, ele aceita essa solicitação e, por isso, os slots de transferência são semelhantes a isto:

- Destination Transfers: 2
- Destination Transfers for `fileServerA`: 2
- Destination transfers for `fileServerB`: 0

Os dois slots `Destination Transfer` no agente agora estão ocupados e, portanto, o agente não pode participar de mais nenhuma transferência gerenciada até que uma das transferências para `fileServerA` seja concluída.

- Um curto tempo depois, `fileServerA` falha, o que faz com que as duas transferências gerenciadas entrem em recuperação. Os slots `Destination transfer` que essas transferências gerenciadas estão usando permanecem em uso durante esse período.
- Em seguida, o agente de ponte de protocolo recebe uma solicitação para transferir um arquivo para `fileServerB`. Há um espaço para essa transferência nos slots `Destination Transfers for fileServerB`, no entanto, todos os slots `Destination Transfer` para o agente estão sendo usados e, portanto, a transferência é colocada na lista não processada para que possa ser tentada novamente mais tarde.

Como resultado, a transferência para `fileServerB` é bloqueada até que pelo menos uma das transferências para `fileServerA` tenha concluído e liberado seu slot `Destination Transfer`.

Para evitar que esta situação aconteça:

- Configure o valor de **maxActiveDestinationTransfers** nos servidores de arquivos para ser menor que o valor **maxDestinationTransfers**, para que os slots livres permaneçam.
- Ou distribua igualmente o valor do atributo **maxActiveDestinationTransfers** entre todos os servidores de terminal.

Comportamento do agente de ponte de protocolo com base nos valores do atributo `maxActiveDestinationTransfers`

Nota: Em todos os casos de erro listados na tabela a seguir, se o atributo `maxActiveDestinationTransfers` for configurado com um valor que não é válido, o agente de ponte de protocolo assumirá que esse atributo não está configurado.

<code>maxActiveDestinationTransfers</code>	Valor de amostra	Descrição
Não especificado	Não especificado	As transferências ocorrem como de costume. Não há limite definido sobre o número de transferências para o terminal <code>*ftp*</code> .
Especificado	0	Nenhuma transferência permitida para esse terminal <code>*ftp*</code> específico.
Valor negativo	-1	Erro registrado em <code>output0.log</code> . O valor -1 não é válido para um número inteiro não negativo. O agente de ponte de protocolo assume que o atributo não está configurado.
Valor de número não inteiro	abc	Erro registrado em <code>output0.log</code> . O valor abc não é válido para um número inteiro. O agente de ponte de protocolo assume que o atributo não está configurado.
Vazio	""	O valor '' do atributo <code>maxActiveDestinationTransfers</code> não é válido para um número inteiro não negativo.
Especificado	5	Permite que apenas cinco transferências ativas estejam em execução a qualquer momento para este terminal <code>*ftp*</code> . As transferências excessivas são tentadas novamente ou rejeitadas, com base no valor do atributo <code>failTransferWhenCapacityReached</code> .

Comportamento do agente de ponte de protocolo para a combinação dos atributos `maxActiveDestinationTransfers` e `failTransferWhenCapacityReached`

Valor de <code>failTransferWhenCapacityReached</code>	Valor de <code>maxActiveDestinationTransfers</code>	Resultado
Não	3	Três transferências ativas são permitidas para este servidor de terminal. Quaisquer transferências adicionais são tentadas novamente.
True	3	Três transferências ativas são permitidas para este servidor de terminal. Quaisquer transferências adicionais são rejeitadas e marcadas como com falha.

Valor de failTransferWhenCapacityReached	Valor de maxActiveDestinationTransfers	Resultado
Não especificado	3	O valor padrão false para failTransferWhenCapacityReached é considerado. O resultado é que três transferências ativas são permitidas para este servidor de terminal. Quaisquer transferências adicionais são tentadas novamente.
Valores diferentes do valor booleano	Especificado	Erro registrado em output.log. O valor especificado para failTransferWhenCapacityReached não é um valor booleano. O valor padrão para failTransferWhenCapacityReached é considerado.

Comportamento do agente de ponte de protocolo para a combinação dos atributos maxDestinationTransfers e failTransferWhenCapacityReached

Valor de failTransferWhenCapacityReached	Valor de maxDestinationTransfers	Resultado
True	10	Quando o número de transferências ativas simultâneas atinge 10, a 11ª transferência gerenciada falha pelo agente de ponte de protocolo.
Não	10	Comportamento existente. Quando o número de transferências ativas simultâneas atinge 10, a 11ª transferência gerenciada é enfileirada aguardando que um slot seja liberado.
Não especificado	10	Comportamento existente

Mensagens de Erro

Mensagem existente:

BFGS0082I

É registrada no arquivo output0.log do agente de origem quando o agente de ponte de protocolo rejeita a transferência, quando o agente de ponte de protocolo já está executando o número máximo de transferências definidas no atributo **maxDestinationTransfers**.

Novas mensagens:

BFGSS0085I

É registrada no arquivo output0.log do agente de origem quando o agente de ponte de protocolo rejeita e tenta novamente uma transferência gerenciada.

BFGSS0086I

É registrada no arquivo output0.log do agente de origem quando o agente de ponte de protocolo rejeita e tenta novamente uma transferência gerenciada e o item de destino não inclui o nome do servidor de arquivos

BFGSS0084E

É registrada no Explorer e no arquivo audit.xml quando o agente de ponte de protocolo rejeita, por exceder o número máximo de transferências simultâneas especificadas no atributo **maxActiveDestinationTransfers**, e marca uma transferência gerenciada como com falha.

BFGSS0087E

É registrada no Explorer e no arquivo audit.xml quando o agente de ponte de protocolo rejeita, por exceder o número máximo de transferências de destino especificado no atributo **maxActiveDestinationTransfers**, e marca uma transferência gerenciada como com falha.

BFGSS0088W

É registrada no output0.log, quando o valor do atributo **maxActiveDestinationTransfers** excede o valor do atributo **maxDestinationTransfers**.

BFGSS0089I

É registrada no arquivo output0.log do agente de ponte de protocolo de destino, quando ele está trabalhando um agente de origem que não está no IBM MQ 9.3.0 ou mais recente.

Conceitos relacionados

[“A ponte de protocolo” na página 294](#)

A ponte de protocolo permite que a rede do Managed File Transfer (MFT) acesse arquivos armazenados em um servidor de arquivos fora da rede do MFT, em seu domínio local ou um local remoto. Este servidor de arquivos pode usar os protocolos de rede FTP, FTPS ou SFTP. Cada servidor de arquivos precisa de pelo menos um agente dedicado. O agente dedicado é conhecido como o agente de ponte de protocolo. Um agente de ponte pode interagir com múltiplos servidores de arquivos.

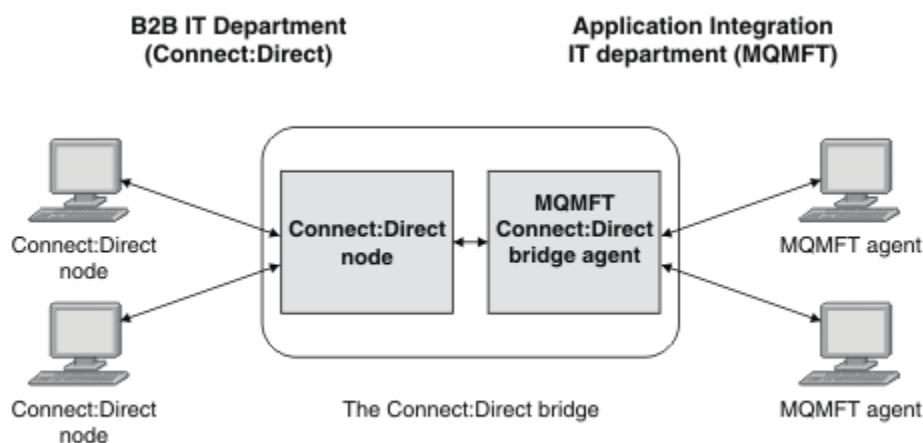
Tarefas relacionadas

[“Definindo propriedades para servidores de arquivos de protocolo usando o arquivo ProtocolBridgeProperties.xml” na página 296](#)

Defina as propriedades de um ou mais servidores de arquivos de protocolo para os quais e dos quais você deseja transferir arquivos usando o arquivo `ProtocolBridgeProperties.xml`, que é fornecido pelo Managed File Transfer no diretório de configuração do agente.

A Ponte Connect:Direct

É possível transferir arquivos de e para uma rede existente do IBM Sterling Connect:Direct. Use a ponte do Connect:Direct, que é um componente do Managed File Transfer, para transferir arquivos entre o MFT e o IBM Sterling Connect:Direct.



O diagrama mostra uma ponte do MFT Connect:Direct entre dois departamentos, o departamento de TI B2B e o departamento de TI Integração de aplicações. O departamento de TI B2B usa o Connect:Direct para transferir arquivos para e a partir dos parceiros de negócios da empresa. O departamento de TI de Integração de Aplicativos usa o IBM MQ como sua infraestrutura do sistema de mensagens e, portanto, escolheu recentemente o Managed File Transfer como sua solução de transferência de arquivos.

Usando a ponte do MFT Connect:Direct, os dois departamentos podem transferir arquivos entre a rede do Connect:Direct no departamento de TI B2B e a rede do MFT no departamento de TI Integração de Aplicativos. A ponte do Connect:Direct é um componente do Managed File Transfer, que inclui um agente do MFT que se comunica com um nó do Connect:Direct. O agente do MFT é dedicado a transferências com o nó Connect:Direct e é conhecido como o agente de ponte do Connect:Direct.

A ponte do Connect:Direct está disponível como parte dos componentes Service e Agent do Managed File Transfer e pode ser usada para as tarefas a seguir:

1. Use os comandos do Managed File Transfer para iniciar uma transferência de um arquivo ou de múltiplos arquivos, de um agente do MFT para um nó do Connect:Direct.
2. Use os comandos do Managed File Transfer para iniciar uma transferência de um arquivo ou de múltiplos arquivos, de um nó do Connect:Direct para um agente do MFT.
3. Use os comandos do Managed File Transfer para iniciar uma transferência de arquivos que inicia um processo do Connect:Direct definido pelo usuário.
4. Use o processo do Connect:Direct para enviar uma solicitação de transferência de arquivos do MFT.

Uma ponte Connect:Direct pode transferir arquivos apenas para ou a partir de nós Connect:Direct. A ponte Connect:Direct pode transferir arquivos para e de um sistema de arquivos local somente como parte de uma transferência enviada por um processo Connect:Direct.

▶ **z/OS** É possível usar a ponte do Connect:Direct para transferir de ou para um conjunto de dados que está localizado em um nó do Connect:Direct em um sistema z/OS. Há algumas diferenças no comportamento comparado com transferências do conjunto de dados que envolvem agentes do Managed File Transfer. Para obter mais informações, consulte ▶ **z/OS** [Transferindo conjuntos de dados para e a partir de nós Connect:Direct.](#)

Plataformas suportadas

A ponte do Connect:Direct é feita de um agente de ponte do MFT Connect:Direct e um nó do Connect:Direct. O agente é suportado no Windows e no Linux for x86-64. O nó é suportado nas plataformas que são suportadas para o IBM Sterling Connect:Direct for Windows e IBM Sterling Connect:Direct for UNIX. Para obter instruções sobre como criar o agente de ponte Connect:Direct e configurar um nó Connect:Direct para o agente para se comunicar, veja [Configurando a ponte Connect:Direct.](#)

A ponte do Connect:Direct pode transferir arquivos de e para os nós do Connect:Direct que estão em execução como parte de uma instalação do Serviço Connect:Direct for Windows ou Connect:Direct for UNIX ou Connect:Direct for z/OS. Para obter detalhes sobre as versões suportadas do Connect:Direct, consulte a página da web [Requisitos do sistema para o IBM MQ](#).

O agente e o nó que formam a ponte do Connect:Direct devem estar no mesmo sistema ou ter acesso ao mesmo sistema de arquivos, por exemplo, por meio de uma montagem NFS compartilhada. Este sistema de arquivos é usado para armazenar temporariamente arquivos durante transferências de arquivos que envolvem a ponte Connect:Direct, em um diretório definido pelo parâmetro **cdTmpDir**. O agente de ponte do Connect:Direct e o nó de ponte do Connect:Direct devem poder endereçar este diretório usando o mesmo nome do caminho. Por exemplo, se o agente e o nó estiverem em sistemas separados do Windows, os sistemas deverão usar a mesma letra da unidade para montar o sistema de arquivo compartilhado. As configurações a seguir permitem que o agente e o nó usem o mesmo nome do caminho:

- O agente e o nó estão no mesmo sistema, que está executando o Windows ou o Linux for x86-64
- O agente está no Linux for x86-64 e o nó está no AIX
- O agente está em um sistema Windows e o nó está em outro sistema Windows

As configurações a seguir não permitem que o agente e o nó usem o mesmo nome do caminho:

- O agente está no Linux for x86-64 e o nó está no Windows
- O agente está no Windows e o nó está no UNIX

Considere esta restrição ao planejar a instalação da ponte do Connect:Direct.

Conceitos relacionados

[“Recuperação e Reinicialização para Transferências para e a partir de Nós Connect:Direct” na página 326](#)
O Managed File Transfer talvez não possa conectar-se ao nó do IBM Sterling Connect:Direct durante uma transferência; por exemplo, se o nó ficar indisponível. O Managed File Transfer tentará recuperar a transferência ou a transferência falhará e será produzida uma mensagem de erro.

[“Enviando um processo do Connect:Direct definido pelo usuário de uma solicitação de transferência de arquivos” na página 328](#)

É possível submeter uma solicitação de transferência para uma transferência que passa pelo agente ponte Connect:Direct que chama um processo Connect:Direct definido pelo usuário como parte da transferência de arquivos.

[“Utilizando processos Connect:Direct para submeter solicitações de transferência Managed File Transfer” na página 332](#)

É possível enviar uma solicitação de transferência para o agente ponte Connect:Direct a partir de um processo Connect:Direct. O Managed File Transfer fornece comandos que podem ser chamados a partir de uma instrução **RUN TASK** em um processo Connect:Direct.

Tarefas relacionadas

[Configurando a Ponte Connect:Direct](#)

[“Transferindo um Arquivo para um Nó Connect:Direct” na página 320](#)

É possível transferir um arquivo de um agente Managed File Transfer para um nó Connect:Direct usando a ponte Connect:Direct. Especifique um nó do Connect:Direct como destino da transferência, especificando o agente de ponte do Connect:Direct como o agente de destino e especificando o arquivo de destino no formulário `connect_direct_node_name:file_path`.

[“Transferindo um Arquivo de um Nó Connect:Direct” na página 321](#)

É possível transferir um arquivo de um nó Connect:Direct para um Managed File Transfer Agent usando a ponte Connect:Direct. É possível especificar um nó do Connect:Direct como a origem da transferência especificando o agente de ponte do Connect:Direct como o agente de origem e especificando a especificação de origem no formulário `connect_direct_node_name:file_path`.

[“Transferindo Vários Arquivos para um Nó Connect:Direct” na página 323](#)

É possível transferir múltiplos arquivos de um agente Managed File Transfer Agent para um nó Connect:Direct usando a ponte Connect:Direct. Para usar um nó do Connect:Direct como destino da transferência de vários arquivos, especifique o agente da ponte do

Connect:Direct como o agente de destino e especifique o diretório de destino no formulário `connect_direct_node_name:directory_path`.

[“Transferring multiple files from a Connect:Direct node” na página 324](#)

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

[“Transferindo Vários Arquivos para Connect:Direct Usando Curingas” na página 325](#)

Para transferir múltiplos arquivos de um agente Managed File Transfer para um nó Connect:Direct, use a ponte Connect:Direct. É possível usar caracteres curinga na especificação de origem fornecida para o comando **fteCreateTransfer**. Assim como todas as transferências do Managed File Transfer que envolvem curingas, apenas a última parte do caminho de arquivo pode conter caracteres curinga. Por exemplo, `/abc/def*` é um caminho de arquivo válido e `/abc*/def` não é válido.

[Resolução de Problemas da Ponte Connect:Direct](#)

Referências relacionadas

[fteCreateCDAgent: criar um agente de ponte Connect:Direct](#)

[Restrição de Conexão:Agente de ponte direta](#)

Transferindo um Arquivo para um Nó Connect:Direct

É possível transferir um arquivo de um agente Managed File Transfer para um nó Connect:Direct usando a ponte Connect:Direct. Especifique um nó do Connect:Direct como destino da transferência, especificando o agente de ponte do Connect:Direct como o agente de destino e especificando o arquivo de destino no formulário `connect_direct_node_name:file_path`.

Antes de começar

Antes de transferir um arquivo, deve-se configurar a ponte Connect:Direct, que é um componente do Managed File Transfer. Para obter mais informações, consulte [Configurando a ponte Connect:Direct](#).

Sobre esta tarefa

Neste exemplo, o agente de ponte Connect:Direct é chamado CD_BRIDGE. O agente de origem é chamado FTE_AGENT e pode estar em qualquer versão de WMQFTE. O nó Connect:Direct de destino é chamado CD_NODE1. O arquivo a ser transferido está localizado no caminho do arquivo `/home/helen/file.log` no sistema em que o FTE_AGENT está localizado. O arquivo é transferido para o caminho de arquivo `/files/data.log` no sistema em que CD_NODE1 está em execução.

Procedimento

1. Use o comando `fteCreateTransfer` com o valor para o parâmetro **-df** (arquivo de destino) no formato `connect_direct_node_name:file_path` e o valor do parâmetro **-da** (agente de destino) especificado como o nome do agente de ponte Connect:Direct.

Nota: O nó Connect:Direct especificado por `connect_direct_node_name` é o nó para o qual você deseja que o arquivo seja transferido, não o nó Connect:Direct que opera como parte da ponte do Connect:Direct.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
-df CD_NODE1:/files/data.log /home/helen/file.log
```

Para obter mais informações, consulte [fteCreateTransfer: iniciar uma nova transferência de arquivos](#).

2. O agente de origem FTE_AGENT transfere o arquivo para o agente de ponte Connect:Direct CD_BRIDGE. O arquivo é armazenado temporariamente no sistema no qual o agente de ponte Connect:Direct está em execução, no local definido pela propriedade do agente `cdTmpDir`. O agente de ponte Connect:Direct transfere o arquivo para o nó Connect:Direct CD_NODE1.

Conceitos relacionados

[“A Ponte Connect:Direct” na página 317](#)

É possível transferir arquivos de e para uma rede existente do IBM Sterling Connect:Direct. Use a ponte do Connect:Direct, que é um componente do Managed File Transfer, para transferir arquivos entre o MFT e o IBM Sterling Connect:Direct.

Tarefas relacionadas

[“Transferindo um Arquivo de um Nó Connect:Direct” na página 321](#)

É possível transferir um arquivo de um nó Connect:Direct para um Managed File Transfer Agent usando a ponte Connect:Direct. É possível especificar um nó do Connect:Direct como a origem da transferência especificando o agente de ponte do Connect:Direct como o agente de origem e especificando a especificação de origem no formulário `connect_direct_node_name:file_path`.

Referências relacionadas

O arquivo `MFT agent.properties`

Transferindo um Arquivo de um Nó Connect:Direct

É possível transferir um arquivo de um nó Connect:Direct para um Managed File Transfer Agent usando a ponte Connect:Direct. É possível especificar um nó do Connect:Direct como a origem da transferência especificando o agente de ponte do Connect:Direct como o agente de origem e especificando a especificação de origem no formulário `connect_direct_node_name:file_path`.

Antes de começar

Antes de transferir um arquivo, deve-se configurar a ponte Connect:Direct, que é um componente do Managed File Transfer. Consulte [Configurando a ponte Connect:Direct](#).

Sobre esta tarefa

Neste exemplo, o agente de ponte Connect:Direct é chamado `CD_BRIDGE`. O agente de destino é chamado `FTE_AGENT` e pode ser qualquer versão do Managed File Transfer. O nó Connect:Direct de origem é chamado `CD_NODE1`. O arquivo a ser transferido está localizado no caminho do arquivo `/home/brian/in.file` no sistema em que `CD_NODE1` está localizado. O arquivo é transferido para o caminho de arquivo `/files/out.file` no sistema em que o `FTE_AGENT` está em execução.

Procedimento

Use o comando `fteCreateTransfer` com o valor da especificação de origem no formulário `connect_direct_node_name:file_path` e o valor do parâmetro `-sa` especificado como o nome do agente ponte Connect:Direct.

Nota: O nó Connect:Direct especificado por `connect_direct_node_name` é o nó do qual você deseja que o arquivo seja transferido, não o nó Connect:Direct que opera como parte da ponte do Connect:Direct. Por exemplo:

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT
                 -df /files/out.file CD_NODE1:/home/brian/in.file
```

Para obter mais informações, consulte [fteCreateTransfer: iniciar uma nova transferência de arquivos](#).

Resultados

O agente de ponte Connect:Direct `CD_BRIDGE` solicita o arquivo do nó Connect:Direct `CD_NODE1`. O nó Connect:Direct envia o arquivo para a ponte Connect:Direct. Enquanto o arquivo está sendo transferido do nó Connect:Direct, a ponte Connect:Direct armazena o arquivo temporariamente no local definido pela propriedade do agente `cdTmpDir`. Quando o arquivo tiver terminado de transferir do nó Connect:Direct para a ponte Connect:Direct, a ponte Connect:Direct então envia o arquivo para o agente de destino `FTE_AGENT` e exclui o arquivo do local temporário.

Conceitos relacionados

[“A Ponte Connect:Direct” na página 317](#)

É possível transferir arquivos de e para uma rede existente do IBM Sterling Connect:Direct. Use a ponte do Connect:Direct, que é um componente do Managed File Transfer, para transferir arquivos entre o MFT e o IBM Sterling Connect:Direct.

Referências relacionadas

O arquivo `MFT.agent.properties`

Transferindo um conjunto de dados para um nó do Connect:Direct no z/OS

É possível transferir um conjunto de dados de um agente Managed File Transfer no z/OS para um nó Connect:Direct no z/OS usando uma ponte Connect:Direct que está localizada em um sistema Windows ou Linux.

Antes de começar

Antes de transferir um arquivo, deve-se configurar a ponte Connect:Direct, que é um componente do Managed File Transfer. Consulte [Configurando a ponte Connect:Direct](#).

Sobre esta tarefa

Nesse exemplo, o parâmetro **-df** é usado para especificar o destino da transferência. O parâmetro **-df** é válido para uso quando o agente de origem da transferência é de qualquer versão do Managed File Transfer. É possível usar o parâmetro **-ds** no lugar. O agente de origem chama-se FTE_ZOS1 e é um agente Managed File Transfer. O agente de ponte do Connect:Direct é chamado CD_BRIDGE e está localizado em um sistema Linux. O nó Connect:Direct de destino é chamado CD_ZOS2. Tanto o agente de origem quanto o nó de destino do Connect:Direct estão localizados em sistemas z/OS. O conjunto de dados a ser transferido está localizado em `//FTEUSER.SOURCE.LIB` no sistema onde o FTE_ZOS1 está localizado. O conjunto de dados é transferido para o conjunto de dados `//CDUSER.DEST.LIB` no sistema onde o CD_ZOS2 está localizado.

Procedimento

1. Utilize o comando `fteCreateTransfer` com o valor para o parâmetro **-df** no formulário: `connect_direct_node_name:data_set_name;attributes` e o valor do parâmetro **-da** (agente de destino) especificado como o nome do agente ponte Connect:Direct.

O nó Connect:Direct especificado por `connect_direct_node_name` é o nó que você deseja que o conjunto de dados seja transferido para, não o nó Connect:Direct que opera como parte da ponte Connect:Direct.

O nome do conjunto de dados dados por `data_set_name` deve ser absoluto, não relativo. Connect:Direct não inclui prefixo no nome do conjunto de dados com o nome do usuário.

```
fteCreateTransfer -sa FTE_ZOS1 -sm QM_ZOS
                  -da CD_BRIDGE -dm QM_BRIDGE
                  -df CD_ZOS2://'CDUSER.DEST.LIB;BLKSIZE(8000);LRECL(80)'  
                  //'FTEUSER.SOURCE.LIB'
```

Para obter mais informações, consulte [fteCreateTransfer: iniciar uma nova transferência de arquivos](#).

2. O agente de origem FTE_ZOS1 transfere os dados no conjunto de dados para o agente ponte Connect:Direct CD_BRIDGE. Os dados são temporariamente armazenados como um arquivo simples no sistema onde o agente ponte Connect:Direct está executando, no local definido pela propriedade de agente `cdTmpDir`. O agente ponte Connect:Direct transfere os dados para o nó Connect:Direct CD_ZOS2. Quando a transferência está concluída, o arquivo simples é excluído do sistema onde o agente ponte Connect:Direct está executando.

Conceitos relacionados

[“A Ponte Connect:Direct” na página 317](#)

É possível transferir arquivos de e para uma rede existente do IBM Sterling Connect:Direct. Use a ponte do Connect:Direct, que é um componente do Managed File Transfer, para transferir arquivos entre o MFT e o IBM Sterling Connect:Direct.

Tarefas relacionadas

[Transferindo conjuntos de dados para e de nós Connect:Direct](#)

Referências relacionadas

[Propriedades BPXWDYN que não se deve usar com o MFT](#)

Transferindo Vários Arquivos para um Nó Connect:Direct

É possível transferir múltiplos arquivos de um agente Managed File Transfer Agent para um nó Connect:Direct usando a ponte Connect:Direct. Para usar um nó do Connect:Direct como destino da transferência de vários arquivos, especifique o agente da ponte do Connect:Direct como o agente de destino e especifique o diretório de destino no formulário `connect_direct_node_name:directory_path`.

Antes de começar

Antes de transferir arquivos, deve-se configurar a ponte Connect:Direct, que é um componente do Managed File Transfer. Consulte [Configurando a ponte Connect:Direct](#).

Sobre esta tarefa

Neste exemplo, o agente de origem é chamado FTE_AGENT. O agente de ponte Connect:Direct é chamado CD_BRIDGE. O nó Connect:Direct de destino é chamado CD_NODE1. Os arquivos a serem transferidos são `/home/jack/data.log`, `/logs/log1.txt` e `/results/latest` no sistema em que FTE_AGENT está localizado. Os arquivos são transferidos para o diretório `/in/files` no sistema em que CD_NODE1 está executando.

Procedimento

Use o comando `fteCreateTransfer` com o valor para o parâmetro **-dd** (diretório de destino) no formato `connect_direct_node_name:directory_path`. Especifique o valor do parâmetro **-da** (agente de destino) como o nome do agente de ponte Connect:Direct.

Nota: O nó Connect:Direct especificado pelo `connect_direct_node_name` é o nó para o qual você deseja que os arquivos sejam transferidos, não o nó Connect:Direct que opera como parte da ponte do Connect:Direct.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

Para obter mais informações, consulte [fteCreateTransfer](#): iniciar uma nova transferência de arquivos.

Resultados

O agente de origem FTE_AGENT transfere o primeiro arquivo para o agente de ponte Connect:Direct CD_BRIDGE. O agente de ponte do Connect:Direct armazena temporariamente o arquivo no local definido pela propriedade `cdTmpDir`. Quando o arquivo tiver sido totalmente transferido do agente de origem para a ponte Connect:Direct, o agente de ponte Connect:Direct enviará o arquivo ao nó Connect:Direct definido pela propriedade do agente `cdNode`. Este nó envia o arquivo para o nó Connect:Direct de destino CD_NODE1. O agente ponte Connect:Direct exclui o arquivo do local temporário quando a transferência entre os dois nós Connect:Direct estiver concluída. Este processo é repetido para cada arquivo de origem especificado.

Conceitos relacionados

[“A Ponte Connect:Direct” na página 317](#)

É possível transferir arquivos de e para uma rede existente do IBM Sterling Connect:Direct. Use a ponte do Connect:Direct, que é um componente do Managed File Transfer, para transferir arquivos entre o MFT e o IBM Sterling Connect:Direct.

Tarefas relacionadas

[“Transferindo um Arquivo para um Nó Connect:Direct” na página 320](#)

É possível transferir um arquivo de um agente Managed File Transfer para um nó Connect:Direct usando a ponte Connect:Direct. Especifique um nó do Connect:Direct como destino da transferência, especificando o agente de ponte do Connect:Direct como o agente de destino e especificando o arquivo de destino no formulário `connect_direct_node_name:file_path`.

[“Transferindo Vários Arquivos para Connect:Direct Usando Curingas” na página 325](#)

Para transferir múltiplos arquivos de um agente Managed File Transfer para um nó Connect:Direct, use a ponte Connect:Direct. É possível usar caracteres curinga na especificação de origem fornecida para o comando **fteCreateTransfer**. Assim como todas as transferências do Managed File Transfer que envolvem curingas, apenas a última parte do caminho de arquivo pode conter caracteres curinga. Por exemplo, `/abc/def*` é um caminho de arquivo válido e `/abc*/def` não é válido.

[“Transferindo um Arquivo de um Nó Connect:Direct” na página 321](#)

É possível transferir um arquivo de um nó Connect:Direct para um Managed File Transfer Agent usando a ponte Connect:Direct. É possível especificar um nó do Connect:Direct como a origem da transferência especificando o agente de ponte do Connect:Direct como o agente de origem e especificando a especificação de origem no formulário `connect_direct_node_name:file_path`.

[“Transferring multiple files from a Connect:Direct node” na página 324](#)

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

Referências relacionadas

O arquivo MFT `agent.properties`

Transferring multiple files from a Connect:Direct node

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

Before you begin

Before transferring a file, you must configure the Connect:Direct bridge, which is a component of Managed File Transfer. See [Configuring the Connect:Direct bridge](#).

About this task

In this example, the Connect:Direct bridge agent is called CD_BRIDGE. The destination agent is called FTE_Z, and is running on a z/OS system. The source Connect:Direct node is called CD_NODE1. The files to be transferred are located at the file paths `/in/file1`, `/in/file2`, and `/in/file3` on the system where CD_NODE1 is located. The files are transferred to the partitioned data set `//OBJECT.LIB` on the system where FTE_Z is running.

Procedure

Use the `fteCreateTransfer` command with the values for the source specifications in the form `connect_direct_node_name:file_path` and the value of the **-sa** parameter specified as the name of the Connect:Direct bridge agent.

Note: The Connect:Direct node specified by `connect_direct_node_name` is the node that you want the files to be transferred from, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z
                  -dp '//OBJECT.LIB' CD_NODE1:/in/file1
                  CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

For more information, see [fteCreateTransfer](#): start a new file transfer.

Results

The Connect:Direct bridge agent CD_BRIDGE requests the first file from the Connect:Direct node CD_NODE1. The Connect:Direct node sends the file to the Connect:Direct bridge. While the file is being transferred from the Connect:Direct node, the Connect:Direct bridge stores the file temporarily in the location defined by the cdTmpDir agent property. When the file has finished transferring from the Connect:Direct node to the Connect:Direct bridge, the Connect:Direct bridge sends the file to the destination agent FTE_Z and then deletes the file from the temporary location. This process is repeated for each specified source file.

Related concepts

[“A Ponte Connect:Direct” on page 317](#)

É possível transferir arquivos de e para uma rede existente do IBM Sterling Connect:Direct. Use a ponte do Connect:Direct, que é um componente do Managed File Transfer, para transferir arquivos entre o MFT e o IBM Sterling Connect:Direct.

Related reference

[The MFT agent.properties file](#)

Transferindo Vários Arquivos para Connect:Direct Usando Curingas

Para transferir múltiplos arquivos de um agente Managed File Transfer para um nó Connect:Direct, use a ponte Connect:Direct. É possível usar caracteres curinga na especificação de origem fornecida para o comando **fteCreateTransfer**. Assim como todas as transferências do Managed File Transfer que envolvem curingas, apenas a última parte do caminho de arquivo pode conter caracteres curinga. Por exemplo, /abc/def* é um caminho de arquivo válido e /abc*/def não é válido.

Antes de começar

Antes de transferir um arquivo, deve-se configurar a ponte Connect:Direct, que é um componente do Managed File Transfer. Para obter mais informações, consulte [Configurando a ponte Connect:Direct](#).

Sobre esta tarefa

Neste exemplo, o agente de origem é chamado FTE_AGENT e o agente de ponte Connect:Direct é chamado CD_BRIDGE. O nó Connect:Direct de destino é chamado CD_NODE1. Os arquivos a serem transferidos estão localizados no diretório /reports no sistema em que o FTE_AGENT está localizado. Apenas os arquivos com nomes que começam com report, seguidos por dois caracteres e o sufixo .log, são transferidos. Por exemplo, o arquivo /reports/report01.log é transferido, mas o arquivo /reports/report1.log não é transferido. Os arquivos são transferidos para o diretório /home/fred no sistema em que CD_NODE1 está executando.

Procedimento

1. Use o comando fteCreateTransfer com o valor para o parâmetro **-dd** (diretório de destino) no formato *connect_direct_node_name:directory_path*. Para o parâmetro **-da** (agente de destino), especifique o agente de ponte Connect:Direct.

Nota: O nó Connect:Direct especificado pelo *connect_direct_node_name* é o nó para o qual você deseja que os arquivos sejam transferidos, não o nó Connect:Direct que opera como parte da ponte do Connect:Direct.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/home/fred "/reports/report?? .log"
```

Para obter mais informações, consulte [fteCreateTransfer](#): iniciar uma nova transferência de arquivos.

2. O agente de origem FTE_AGENT transfere o primeiro arquivo que corresponde ao padrão /reports/report?? .log para o agente de ponte do Connect:Direct CD_BRIDGE. O agente de ponte do Connect:Direct armazena temporariamente o arquivo no local definido pela propriedade cdTmpDir.

Quando o arquivo tiver sido totalmente transferido do agente de origem para a ponte Connect:Direct, o agente de ponte Connect:Direct enviará o arquivo ao nó Connect:Direct definido pela propriedade do agente cdNode. Este nó envia o arquivo para o nó Connect:Direct de destino CD_NODE1. O agente ponte Connect:Direct exclui o arquivo do local temporário quando a transferência entre os dois nós Connect:Direct estiver concluída. Este processo é repetido para cada arquivo de origem que corresponde ao padrão de carne curinga `/reports/report??.log`.

Nota: A lista de arquivos que correspondem ao padrão de `/reports/report??.log` varia dependendo do sistema operacional do sistema em que o agente de origem FTE_AGENT está localizado.

- Se o agente de origem estiver localizado em um sistema com um sistema operacional Windows, a correspondência de padrões não fará distinção entre maiúsculas e minúsculas. O padrão corresponde a todos os arquivos no diretório `/reports` com um nome de arquivo no formato `report` seguido por dois caracteres e um sufixo de `.log`, independentemente do caso em que as letras estejam. Por exemplo, `Report99.log` é uma correspondência.
- Se o agente de origem estiver localizado em um sistema com um sistema operacional Linux ou UNIX, a correspondência de padrões fará distinção entre maiúsculas e minúsculas. O padrão corresponde apenas a esses arquivos no diretório `/reports` com um nome de arquivo do formato `report` seguido por dois caracteres e um sufixo de `.log`. Por exemplo, `reportAB.log` é uma correspondência, mas `reportAB.LOG` e `Report99.log` não são correspondências.

Conceitos relacionados

[“A Ponte Connect:Direct” na página 317](#)

É possível transferir arquivos de e para uma rede existente do IBM Sterling Connect:Direct. Use a ponte do Connect:Direct, que é um componente do Managed File Transfer, para transferir arquivos entre o MFT e o IBM Sterling Connect:Direct.

Tarefas relacionadas

[Usando caracteres curinga com o MFT](#)

[“Transferindo um Arquivo para um Nó Connect:Direct” na página 320](#)

É possível transferir um arquivo de um agente Managed File Transfer para um nó Connect:Direct usando a ponte Connect:Direct. Especifique um nó do Connect:Direct como destino da transferência, especificando o agente de ponte do Connect:Direct como o agente de destino e especificando o arquivo de destino no formulário `connect_direct_node_name:file_path`.

[“Transferindo Vários Arquivos para um Nó Connect:Direct” na página 323](#)

É possível transferir múltiplos arquivos de um agente Managed File Transfer Agent para um nó Connect:Direct usando a ponte Connect:Direct. Para usar um nó do Connect:Direct como destino da transferência de vários arquivos, especifique o agente da ponte do Connect:Direct como o agente de destino e especifique o diretório de destino no formulário `connect_direct_node_name:directory_path`.

[“Transferring multiple files from a Connect:Direct node” na página 324](#)

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

Referências relacionadas

[O arquivo MFT agent.properties](#)

Recuperação e Reinicialização para Transferências para e a partir de Nós Connect:Direct

O Managed File Transfer talvez não possa conectar-se ao nó do IBM Sterling Connect:Direct durante uma transferência; por exemplo, se o nó ficar indisponível. O Managed File Transfer tentará recuperar a transferência ou a transferência falhará e será produzida uma mensagem de erro.

Se o Nó Connect:Direct Ficar Indisponível

Se o nó do Connect:Direct ficar indisponível; por exemplo, devido a uma indisponibilidade de rede ou de energia, o Managed File Transfer recuperará uma transferência de arquivos das seguintes maneiras:

- Se o Managed File Transfer não tiver se conectado anteriormente com êxito ao nó Connect:Direct como parte dessa solicitação de transferência, a transferência será tentada novamente por um período de tempo determinado pelos valores do **cdMaxConnectionRetries** e do **recoverableTransferRetryInterval** properties. Essas propriedades são especificadas no arquivo `agent.properties` para o agente de ponte Connect:Direct. A transferência falha e uma mensagem de erro é produzida após o número de tentativas com falha atingir o valor do **cdMaxConnectionRetries** property. Por padrão, a transferência é tentada indefinidamente, com intervalos de 60 segundos.
- Se, anteriormente, o Managed File Transfer tiver se conectado com êxito ao nó do Connect:Direct como parte desta solicitação de transferência, a transferência será tentada novamente por um período de tempo determinado pelos valores das propriedades **cdMaxPartialWorkConnectionRetries** e **recoverableTransferRetryInterval**. A transferência falha e é produzida uma mensagem de erro, após o número de tentativas com falha atingir o valor da propriedade **cdMaxPartialWorkConnectionRetries**. Por padrão, a transferência é tentada indefinidamente, com intervalos de 60 segundos.
- Para determinados tipos de falha do nó Connect:Direct, por exemplo, o nó que está sendo interrompido à força, os processos do Connect:Direct entram no status `HEld Due to Error` (HE) quando o nó é recuperado. Após a recuperação do nó, o Managed File Transfer retomará automaticamente os processos do Connect:Direct que estão relacionados à transferência de arquivos e têm um status de HE.
- Se a transferência falhar, os arquivos temporários relacionados à transferência serão excluídos do sistema que hospeda a ponte Connect:Direct. O local desses arquivos temporários é definido pela propriedade **cdTmpDir**.
- Se a transferência for do Managed File Transfer para o Connect:Direct e uma disposição de origem de exclusão for especificada, os arquivos de origem não serão excluídos se a transferência falhar.

Se as Credenciais do Usuário do Nó Connect:Direct Forem Inválidas

Se o Managed File Transfer falhar ao conectar-se ao nó do Connect:Direct porque as credenciais do usuário são rejeitadas pelo nó, a transferência falhará e será produzida uma mensagem de erro. Neste caso, verifique se você forneceu as credenciais do usuário corretas para o nó Connect:Direct. Para obter mais informações, consulte [Mapeando credenciais para o Connect:Direct](#).

Se o Agente de Ponte Connect:Direct Ficar Indisponível

Se o agente de ponte do Connect:Direct ficar indisponível, as transferências de arquivos em andamento serão recuperadas da mesma maneira que as transferências padrão do Managed File Transfer. Para obter informações adicionais, consulte [“Recuperação e reinicialização do MFT”](#) na página 334.

Conceitos relacionados

[“A Ponte Connect:Direct”](#) na página 317

É possível transferir arquivos de e para uma rede existente do IBM Sterling Connect:Direct. Use a ponte do Connect:Direct, que é um componente do Managed File Transfer, para transferir arquivos entre o MFT e o IBM Sterling Connect:Direct.

[“Recuperação e reinicialização do MFT”](#) na página 334

Se seu agente ou gerenciador de filas estiver indisponível por alguma razão, por exemplo, devido a uma falha de energia ou de rede, o Managed File Transfer se recuperará da seguinte forma nestes cenários:

Tarefas relacionadas

[Configurando a Ponte Connect:Direct](#)

Referências relacionadas

[O arquivo MFT `agent.properties`](#)

Enviando um processo do Connect:Direct definido pelo usuário de uma solicitação de transferência de arquivos

É possível submeter uma solicitação de transferência para uma transferência que passa pelo agente ponte Connect:Direct que chama um processo Connect:Direct definido pelo usuário como parte da transferência de arquivos.

Por padrão, quando você submete uma solicitação de transferência de arquivos para uma transferência que passa pela ponte Connect:Direct, o agente ponte Connect:Direct gera o processo Connect:Direct que é usado para transferir o arquivo para ou de nós Connect:Direct remotos.

No entanto, é possível configurar o agente de ponte do Connect:Direct para em vez disso chamar um processo Connect:Direct definido pelo usuário usando o arquivo `ConnectDirectProcessDefinition.xml`.

O arquivo `ConnectDirectProcessDefinition.xml`

O comando `fteCreateCDAgent` cria o arquivo `ConnectDirectProcessDefinitions.xml` no diretório de configuração do agente `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`. Antes de poder chamar processos Connect:Direct definidos pelo usuário do agente ponte Connect:Direct, deve-se configurar definições de processo editando esse arquivo.

O arquivo define um ou mais conjuntos de processos que incluem o local de um ou mais processos Connect:Direct que são chamados como parte de uma transferência. Cada conjunto de processos inclui algumas condições. Se a transferência atender a todas as condições do conjunto de processos, o conjunto de processos será usado para especificar quais processos do Connect:Direct são chamados pela transferência. Para obter informações adicionais, consulte [“Especificando o processo Connect:Direct a ser iniciado usando o arquivo `ConnectDirectProcessDefinition.xml`” na página 329](#).

Variáveis Simbólicas Intrínsecas

É possível usar as variáveis simbólicas intrínsecas definidas pelo Managed File Transfer para substituir valores em processos definidos pelo usuário do Connect:Direct. Para seguir a convenção de nomenclatura de Connect:Direct, todas as variáveis simbólicas intrínsecas utilizadas pelo Managed File Transfer possuem formato `%FTE` seguido por cinco caracteres alfanuméricos maiúsculos.

Ao criar um processo para transferir arquivos de um nó do Connect:Direct para o sistema de ponte do Connect:Direct, deve-se usar a variável intrínseca `%FTETFILE` como o valor de TO FILE no processo do Connect:Direct. Ao criar um processo para transferir arquivos para um nó do Connect:Direct do sistema de ponte do Connect:Direct, deve-se usar a variável intrínseca `%FTEFFILE` como o valor de FROM FILE no processo do Connect:Direct. Estas variáveis contêm os caminhos de arquivos temporários usados pelo agente de ponte do Connect:Direct para transferências dentro e fora da rede do Managed File Transfer.

Para obter mais informações sobre variáveis simbólicas intrínsecas, consulte a documentação do produto Connect:Direct.

Processos de Amostra do Connect:Direct

O Managed File Transfer fornece processos de amostra do Connect:Direct. Essas amostras estão localizadas no seguinte diretório: `MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates`.

Tarefas relacionadas

[“Especificando o processo Connect:Direct a ser iniciado usando o arquivo `ConnectDirectProcessDefinition.xml`” na página 329](#)

Especifique qual processo Connect:Direct iniciar como parte de uma transferência do Managed File Transfer. O Managed File Transfer fornece um arquivo XML possível de editar para especificar definições de processo.

[“Usando Variáveis Simbólicas Intrínsecas em Processos do Connect:Direct que São Chamadas pelo Managed File Transfer”](#) na página 330

É possível instalar um processo definido pelo usuário do Connect:Direct a partir de uma transferência do Managed File Transfer e transmitir informações da transferência para o processo do Connect:Direct, usando variáveis simbólicas intrínsecas na definição de processo.

Referências relacionadas

Formato de arquivo de definição de processo Connect:Direct

[Variáveis de Substituição para Usar com Processos Connect:Direct Definidos pelo Usuário](#)

Especificando o processo Connect:Direct a ser iniciado usando o arquivo *ConnectDirectProcessDefinition.xml*

Especifique qual processo Connect:Direct iniciar como parte de uma transferência do Managed File Transfer. O Managed File Transfer fornece um arquivo XML possível de editar para especificar definições de processo.

Sobre esta tarefa

O comando **fteCreateCDAgent** cria o arquivo `ConnectDirectProcessDefinitions.xml` no diretório de configuração do agente `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`. Antes de poder chamar processos Connect:Direct definidos pelo usuário do agente ponte Connect:Direct, deve-se configurar definições de processo editando esse arquivo.

Para cada processo que você deseja especificar para chamar como parte de uma transferência pela ponte Connect:Direct, execute as seguintes etapas:

Procedimento

1. Defina o processo Connect:Direct que deseja que o agente ponte Connect:Direct chame como parte da transferência e salve o modelo do processo em um arquivo.
2. Abra o arquivo `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` em um editor de texto.
3. Crie um elemento `<processSet>`.
4. Dentro do elemento `<processSet>`, crie um elemento `<condition>`.
5. Dentro do elemento `<condition>`, crie um ou mais elementos que definam uma condição que a solicitação de transferência deve corresponder para chamar o processo Connect:Direct que você definiu na Etapa 1. Esses elementos podem ser tanto elementos `<match>` quanto elementos `<defined>`.
 - Use um elemento `<match>` para especificar que o valor de uma variável deve corresponder a um padrão. Crie o elemento `<match>` com os atributos a seguir:
 - `variable` - o nome da variável cujo valor é comparado. A variável é um símbolo intrínseco. Para obter mais informações, consulte [Variáveis de substituição para uso com processos do Connect:Direct definidos pelo usuário](#).
 - `value` - o padrão para comparar o valor da variável especificada.
 - Opcional: `pattern` - o tipo de padrão utilizado pelo valor do atributo `value`. Esse tipo de padrão pode ser `wildcard` ou `regex`. Esse atributo é opcional e o padrão é `wildcard`.
 - Use um elemento `<defined>` para especificar que uma variável deve ter um valor definido. Crie o elemento `<defined>` com o atributo a seguir:
 - `variable` - o nome da variável que deve ter um valor definido. A variável é um símbolo intrínseco. Para obter mais informações, consulte [Variáveis de substituição para uso com processos do Connect:Direct definidos pelo usuário](#).

As condições especificadas no elemento `<condition>` são combinadas com um AND lógico. Todas as condições devem ser atendidas para que o agente de ponte do Connect:Direct chame o processo

especificado por este elemento <processSet>. Se você não especificar um elemento <condition>, o conjunto de processos corresponderá a todas as transferências.

6. Dentro do elemento <processSet>, crie um elemento <process>.

7. Dentro do elemento <process>, crie um elemento <transfer>.

O elemento de transferência especifica o processo Connect:Direct que o agente ponte Connect:Direct chama como parte da transferência. Crie o elemento <transfer> com o atributo a seguir:

- process - o local do processo Connect:Direct definido na etapa 1. O local deste arquivo é especificado com um caminho absoluto ou relativo ao diretório `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`.

Resultados

Ao procurar por uma correspondência de condição, o agente de ponte do Connect:Direct procura desde o início do arquivo até o fim dele. A primeira correspondência localizada será usada.

Tarefas relacionadas

[Configurando a Ponte Connect:Direct](#)

Referências relacionadas

[Formato de arquivo de definição de processo Connect:Direct](#)

[fteCreateCDAgent: criar um agente de ponte Connect:Direct](#)

Usando Variáveis Simbólicas Intrínsecas em Processos do Connect:Direct que São Chamadas pelo Managed File Transfer

É possível instalar um processo definido pelo usuário do Connect:Direct a partir de uma transferência do Managed File Transfer e transmitir informações da transferência para o processo do Connect:Direct, usando variáveis simbólicas intrínsecas na definição de processo.

Sobre esta tarefa

Este exemplo usa variáveis simbólicas intrínsecas para transmitir informações de uma transferência do Managed File Transfer para um processo definido pelo usuário do Connect:Direct. Para obter mais informações sobre variáveis de substituição intrínsecas usadas pelo Managed File Transfer, veja [Variáveis de substituição para uso com processos do Connect:Direct definidos pelo usuário](#).

Neste exemplo, o arquivo é transferido de um Managed File Transfer Agent para um nó de ponte Connect:Direct. A primeira parte da transferência é executada pelo Managed File Transfer. A segunda parte da transferência é executada por um processo Connect:Direct definido pelo usuário.

Procedimento

1. Crie um processo Connect:Direct que utiliza variável simbólica intrínseca.

```
%FTEPNAME PROCESS
  SNODE=%FTESNODE
  PNODEID=(%FTEPUSER,%FTEPPASS)
  SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
  FROM (
    FILE=%FTEFFILE
    DISP=%FTEFDISP
  )
  TO (
    FILE=%FTETFILE
    DISP=%FTETDISP
  )
PEND
```

2. Salve este processo em um arquivo de texto no seguinte local: `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent/Example.cdp`

3. Edite o arquivo `ConnectDirectProcessDefinition.xml` para incluir uma regra que chama o processo Connect:Direct que você criou na Etapa 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/
  ConnectDirectProcessDefinitions ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="TOBERMORY" pattern="wildcard" />
    </tns:condition>
    <tns:process>
      <tns:transfer process="Example.cdp" />
    </tns:process>
  </tns:processSet>
</tns:cdprocess>
```

Neste exemplo, se uma solicitação de transferência for enviada ao agente de ponte do Connect:Direct que tem TOBERMORY como seu nó Connect:Direct de origem ou destino, o processo `Example.cdp` Connect:Direct é chamado.

4. Envie uma solicitação de transferência de arquivos que satisfaça as condições que você definiu no arquivo `ConnectDirectProcessDefinition.xml` na Etapa 3.

Por exemplo,

```
fteCreateTransfer -sa ORINOCO -da CD_BRIDGE
                 -sm QM_WIMBLEDON -dm QM_COMMON
                 -de overwrite -df TOBERMORY:/home/bulgaria/destination.txt
                 -sd leave c:\bungo\source.txt
```

Nesse exemplo, o nó Connect:Direct de destino é TOBERMORY. Esse nó é o nó secundário na transferência e o valor `%FTESNODE` é configurado para TOBERMORY. Este comando corresponde à condição que está configurada no arquivo `ConnectDirectProcessDefinition.xml`.

5. O Managed File Transfer transfere o arquivo de origem para um local temporário no mesmo sistema que o agente de ponte do Connect:Direct.
6. O agente ponte Connect:Direct configura os valores das variáveis simbólicas intrínsecas das informações na solicitação de transferência e informações de configuração.

As variáveis simbólicas intrínsecas são configuradas para os seguintes valores:

- `%FTEPNAME=process_name` - Esse valor é um nome de processo de 8 caracteres gerado pelo agente ponte Connect:Direct.
- `%FTESNODE=TOBERMORY` - Esse valor é um conjunto de parâmetro **-df** do comando **fteCreateTransfer**.
- `%FTEPUSER,=primary_node_user` - Esta informação é retirada do arquivo `ConnectDirectCredentials.xml`.
- `%FTEPPASS=primary_node_user_password` - Esta informação é retirada do arquivo `ConnectDirectCredentials.xml`.
- `%FTESUSER,=secondary_node_user` - Esta informação é retirada do arquivo `ConnectDirectCredentials.xml`.
- `%FTESPASS=secondary_node_user_password` - Esta informação é retirada do arquivo `ConnectDirectCredentials.xml`.
- `%FTEFFILE=temporary_location` - Esse valor é o local temporário do arquivo no mesmo sistema que o agente ponte Connect:Direct.
- `%FTEFDISP=leave` - Esse valor é o conjunto de parâmetro **-sd** do comando **fteCreateTransfer**.
- `%FTETFILE=/home/bulgaria/destination.txt` - Esse valor é um conjunto de parâmetro **-df** do comando **fteCreateTransfer**.
- `%FTETDISP=overwrite` - Esse valor é um conjunto de parâmetro **-de** do comando **fteCreateTransfer**.

7. O processo Connect:Direct é iniciado no nó ponte Connect:Direct. O Connect:Direct transfere o arquivo do local temporário no sistema de ponte Connect:Direct para o destino `/home/bulgaria/destination.txt` no sistema onde o nó Connect:Direct TOBERMORY está executando.

Conceitos relacionados

[“Enviando um processo do Connect:Direct definido pelo usuário de uma solicitação de transferência de arquivos” na página 328](#)

É possível submeter uma solicitação de transferência para uma transferência que passa pelo agente ponte Connect:Direct que chama um processo Connect:Direct definido pelo usuário como parte da transferência de arquivos.

Referências relacionadas

[Variáveis de Substituição para Usar com Processos Connect:Direct Definidos pelo Usuário](#)

Utilizando processos Connect:Direct para submeter solicitações de transferência Managed File Transfer

É possível enviar uma solicitação de transferência para o agente ponte Connect:Direct a partir de um processo Connect:Direct. O Managed File Transfer fornece comandos que podem ser chamados a partir de uma instrução **RUN TASK** em um processo Connect:Direct.

O Managed File Transfer fornece os seguintes comandos para usar com processos do Connect:Direct:

ftetag

Especifique esse comando em uma etapa que antecede o comando **ftebxfex** ou **ftecxfer** para criar as informações de auditoria necessárias para a transferência. Esse comando toma a especificação de origem da transferência como um parâmetro. Para obter informações sobre o formato da especificação de origem, consulte [fteCreateTransfer: iniciar uma nova transferência de arquivos](#)

ftebxfex

Especifique esse comando para criar uma solicitação de transferência de arquivos quando o gerenciador de fila para o qual a solicitação de transferência é enviada estiver no mesmo sistema que o nó Connect:Direct que envia o comando. Esse comando usa alguns parâmetros como o comando **fteCreateTransfer**. Para obter informações sobre esses parâmetros, veja [fteCreateTransfer: iniciar uma nova transferência de arquivos](#). Esse comando também possui um parâmetro extra:

-qmgrname

Obrigatório. O nome do gerenciador de fila para o qual enviar o comando.

ftecxfer

Especifique esse comando para criar uma solicitação de transferência do arquivo quando o gerenciador de fila ao qual a solicitação de transferência é enviada está localizado em um sistema diferente do nós Connect:Direct que submetem o comando. Esse comando usa alguns parâmetros como o comando **fteCreateTransfer**. Para obter informações sobre os parâmetros, veja [fteCreateTransfer: iniciar uma nova transferência de arquivos](#). Esse comando também possui três parâmetros adicionais:

-qmgrname

Obrigatório. O nome do gerenciador de fila para o qual enviar o comando.

-connname

Obrigatório. O host e a porta do gerenciador de fila aos quais o comando será enviado, especificados no formato CONNAME do IBM MQ. Por exemplo, `host.example.com(1337)`.

-channelname

Opcional. O nome do canal a usar para conectar-se ao gerenciador de fila ou ao qual enviar o comando. Se ele não for especificado, um padrão `SYSTEM.DEF.SVRCONN` será usado.

Tarefas relacionadas

[“Criando e Submetendo um Processo do Connect:Direct que Chama o Managed File Transfer Usando o Connect:Direct Requester” na página 333](#)

O Connect:Direct Requester é uma interface gráfica com o usuário possível de usar para criar e submeter um processo do Connect:Direct que chama o Managed File Transfer.

Referências relacionadas

Exemplo: [um arquivo de processo Connect:Direct que chama comandos do MFT](#)

Criando e Submetendo um Processo do Connect:Direct que Chama o Managed File Transfer Usando o Connect:Direct Requester

O Connect:Direct Requester é uma interface gráfica com o usuário possível de usar para criar e submeter um processo do Connect:Direct que chama o Managed File Transfer.

Sobre esta tarefa

Essa tarefa descreve como criar um processo do Connect:Direct que chama o comando Managed File Transfer **ftecxfer** ou o comando **ftebxfer**. Use o comando **ftecxfer** quando o gerenciador de filas para o qual a solicitação de transferência é enviada estiver localizado em um sistema diferente do que o nó Connect:Direct que envia o comando. Use o comando **ftebxfer** quando o gerenciador de filas para o qual a solicitação de transferência é enviada estiver localizado no mesmo sistema que o nó Connect:Direct que envia o comando. O comando **ftecxfer** estabelece uma conexão de cliente com o gerenciador de filas do agente de origem da transferência. Antes de chamar o comando **ftecxfer** é necessário chamar o comando **ftetag** e transmitir as informações de especificação de origem. Isto permite que o processo seja registrado e auditado da mesma forma que as transferências iniciadas do Managed File Transfer.

Procedimento

1. Inicie o Solicitante Connect:Direct.
2. Na guia **Nós** do painel, selecione o nó do Connect:Direct que seja usado como o nó primário do processo.
3. Selecione **Arquivo > Novo > Processo**. A janela de **Propriedades do processo** é aberta.
4. No campo **Nome:**, digite o nome do processo.
5. Selecione o nó secundário da lista **Snode > Nome:**.
6. Selecione o sistema operacional do nó secundário a partir da lista **Snode > Sistema Operacional:**.
7. Opcional: Complete todas as informações nessa janela que você precisa.
8. Clique em **OK**. A janela **Propriedades do Processo** é fechada.
9. Crie uma instrução que executa o comando Managed File Transfer **ftetag**.
 - a) Clique com o botão direito do mouse na janela **Processo** na instrução **End**.
 - b) Selecione **Inserir > Executar Tarefa**. A janela **Instrução para Executar Tarefa** é aberta.
 - c) No campo **Rótulo:**, digite Tag.
 - d) No campo **Parâmetros opcionais ou comandos**, digite `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)`. Para obter mais informações sobre o formato de *source_specification*, veja [fteCreateTransfer](#): iniciar uma nova transferência de arquivos.
 - e) Clique em **OK**. A janela **Instrução para Executar Tarefa** é fechada.
10. Crie uma instrução que execute o comando Managed File Transfer **ftecxfer** ou **ftebxfer**.
 - a) Clique com o botão direito do mouse na janela **Processo** na instrução **End**.
 - b) Selecione **Inserir > Executar Tarefa**. A janela **Instrução para Executar Tarefa** é aberta.
 - c) No campo **Rótulo:**, digite Transfer.
 - d) No campo **Parâmetros opcionais ou comandos**, digite `pgm(MQ_INSTALLATION_PATH/bin/ftecxfer) args(parameters)` ou `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)` dependendo de qual comando você escolher. Os parâmetros usados pelos comandos **ftecxfer** e **ftebxfer** são os mesmos que os parâmetros usados pelo comando **fteCreateTransfer**, mais alguns parâmetros adicionais específicos para o **ftecxfer** e o **ftebxfer**. Para obter mais informações, veja [fteCreateTransfer](#): iniciar uma nova

transferência de arquivos e [“Utilizando processos Connect:Direct para submeter solicitações de transferência Managed File Transfer”](#) na página 332.

- e) Clique em **OK**. A janela **Instrução para Executar Tarefa** é fechada.
11. Opcional: Crie qualquer instrução adicional que desejar.
12. Submeta o processo.
 - a) Clique com o botão direito na janela **Processo**.
 - b) Selecione **Enviar**. A janela **Connect:Direct Attach** é aberta.
 - c) Insira o nome de usuário e senha para usar ao executar o processo.
 - d) Clique em **OK**.

Conceitos relacionados

[“Utilizando processos Connect:Direct para submeter solicitações de transferência Managed File Transfer”](#) na página 332

É possível enviar uma solicitação de transferência para o agente ponte Connect:Direct a partir de um processo Connect:Direct. O Managed File Transfer fornece comandos que podem ser chamados a partir de uma instrução **RUN TASK** em um processo Connect:Direct.

Trabalhando com o MFT no IBM Integration Bus

É possível trabalhar com o Managed File Transfer a partir do IBM Integration Bus usando os nós FTEOutput e FTEInput.

- Use o nó FTEInput para transferir um arquivo através da rede usando o Managed File Transfer e, em seguida, processe esse arquivo como parte de um fluxo do Integration Bus.
- Use o nó FTEOutput para transferir um arquivo que foi produzido por um fluxo do Integration Bus para outro local na rede.

Os agentes que transferem arquivos para ou do agente broker podem estar em qualquer nível do Managed File Transfer.

Para obter mais informações, consulte a [documentação do produto IBM Integration Bus](#).

Recuperação e reinicialização do MFT

Se seu agente ou gerenciador de filas estiver indisponível por alguma razão, por exemplo, devido a uma falha de energia ou de rede, o Managed File Transfer se recuperará da seguinte forma nestes cenários:

- Geralmente, se houver um problema enquanto um arquivo está sendo transferido, o Managed File Transfer recuperará e reiniciará essa transferência de arquivos após o reparo do problema.
- Se um arquivo que estava no processo para ser transferido for excluído ou alterado enquanto o agente ou gerenciador de filas estiver indisponível, a transferência falhará e você receberá uma mensagem no log de transferência que fornece detalhes sobre a falha.
- Se um processo do agente falhar durante uma transferência de arquivos, a transferência continuará quando você reiniciar o agente.
- Se um agente perder a conexão com seu gerenciador de filas do agente, o agente esperará enquanto tenta reconectar-se ao gerenciador de filas. Quando o agente se reconectar com êxito a seu gerenciador de filas, a transferência atual continuará.
- Se o agente for parado por alguma razão, qualquer monitor do recurso associado com um agente para de pesquisar. Quando o agente é recuperado, os monitores também são reiniciados e a sondagem de recurso é retomada.
- Para uma transferência de arquivos com uma disposição de origem de `exclusão`, se uma recuperação ocorrer após todos os dados serem enviados a partir de um agente de origem para um agente de destino, o arquivo de origem estará desbloqueado antes da exclusão. Este desbloqueio significa que o

arquivo de origem pode ser modificado antes do arquivo ser excluído. Portanto, é considerado inseguro excluir o arquivo de origem e o seguinte aviso será exibido:

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file was modified after the source file was transferred.
```

Neste caso, verifique se o conteúdo do arquivo de origem não é modificado e, em seguida, exclua manualmente o arquivo de origem.

É possível verificar o status de suas transferências no IBM MQ Explorer. Se quaisquer transferências aparecerem como Stalled, poderá ser necessário tomar uma ação corretiva porque o status paralisado denota um problema com o agente ou entre os dois agentes envolvidos na transferência

Tarefas relacionadas

[“Configurando um tempo limite para recuperação de transferências paralisadas” na página 335](#)

É possível configurar um tempo limite de recuperação de transferência para transferências de arquivos paralisadas que se aplicam a todas as transferências de um agente de origem. Também é possível configurar um tempo limite de recuperação de transferência de uma transferência individual. Se você configurar um período de tempo específico, em segundos, durante o qual um agente de origem continue tentando recuperar uma transferência de arquivos paralisada e a transferência não for bem-sucedida quando o agente atingir o tempo limite, a transferência falhará.

Configurando um tempo limite para recuperação de transferências paralisadas

É possível configurar um tempo limite de recuperação de transferência para transferências de arquivos paralisadas que se aplicam a todas as transferências de um agente de origem. Também é possível configurar um tempo limite de recuperação de transferência de uma transferência individual. Se você configurar um período de tempo específico, em segundos, durante o qual um agente de origem continue tentando recuperar uma transferência de arquivos paralisada e a transferência não for bem-sucedida quando o agente atingir o tempo limite, a transferência falhará.

Sobre esta tarefa

É possível configurar um tempo limite de recuperação de transferência que se aplique a todas as transferências para um agente de origem, incluindo um parâmetro de tempo limite de recuperação de transferência no arquivo `agent.properties` do agente. Também é possível configurar um tempo limite de recuperação de transferência para uma transferência individual da linha de comandos, ou com IBM MQ Explorer, ou usando tarefas do Apache Ant. Se houver um valor de tempo limite de recuperação de transferência configurado no arquivo `agent.properties`, configurar o tempo limite de recuperação de transferência de uma transferência individual substituirá o valor no arquivo `agent.properties`.

Há três opções para o tempo limite de recuperação de transferência:

- O agente continuará tentando recuperar a transferência paralisada até que ela seja concluída com sucesso. Isso será o mesmo que o comportamento padrão do agente se o tempo limite de recuperação de transferência não for configurado.
- O agente marca a transferência como com falha imediatamente ao entrar em recuperação.
- O agente continua tentando novamente a transferência paralisada por um período de tempo especificado antes que a transferência seja marcada como com falha.

A configuração do tempo limite de recuperação de transferência de arquivos é opcional. Se você não configurá-la, as transferências seguirão o comportamento padrão, em que o agente continua tentando recuperar uma transferência paralisado até que ela seja bem-sucedida

Conceitos relacionados

[“Recuperação e reinicialização do MFT” na página 334](#)

Se seu agente ou gerenciador de filas estiver indisponível por alguma razão, por exemplo, devido a uma falha de energia ou de rede, o Managed File Transfer se recuperará da seguinte forma nestes cenários:

Conceitos de tempo limite de recuperação da transferência

É possível configurar o período de tempo, em segundos, durante o qual um agente de origem continuará tentando recuperar uma transferência de arquivos paralisada. Se a transferência não for bem-sucedida quando o agente atingir o tempo limite para o intervalo de novas tentativas, a transferência falhará.

Precedência do tempo limite de recuperação

Um valor de tempo limite de recuperação de transferência de uma transferência individual especificado por meio dos comandos **fteCreateTransfer**, **fteCreateTemplate** ou **fteCreateMonitor**, usando o IBM MQ Explorer ou especificado no elemento aninhado **fte:filespec**, tem precedência sobre o valor especificado para o parâmetro **transferRecoveryTimeout** no arquivo `agent.properties` do agente de origem.

Por exemplo, se o comando **fteCreateTransfer** for iniciado sem o parâmetro **-rt** e o par de valores, o agente de origem AGENT1 verificará se no arquivo `agent.properties` tem um valor **transferRecoveryTimeout** para determinar o comportamento do tempo limite de recuperação:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

Se o parâmetro **transferRecoveryTimeout** no arquivo `agent.properties` não estiver configurado ou estiver configurado como **-1**, o agente seguirá o comportamento padrão e tentará recuperar a transferência até que ela seja bem-sucedida.

No entanto, se o comando **fteCreateTransfer** incluir o parâmetro **-rt**, o valor desse parâmetro terá precedência sobre o valor no arquivo `agent.properties` e será usado como a configuração de tempo limite de recuperação da transferência:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

Contador de tempo limite de recuperação

O contador de tempo limite de recuperação é iniciado quando a transferência entra no estado de recuperação. Uma mensagem de log de transferência é publicada no tópico SYSTEM.FTE com a sequência de tópicos `Log/agent_name/transfer_ID` para indicar que o status da transferência mudou para recuperação e o horário do clock do agente de origem no qual o status mudou. Se a transferência continuar dentro do intervalo de novas tentativas configurado e não atingir o tempo limite de recuperação ($\text{contador} \leq \text{tempo limite de recuperação}$), o contador será reconfigurado para 0, pronto para iniciar novamente se a transferência entrar em recuperação.

Se o contador atingir o valor máximo configurado para o tempo limite de recuperação ($\text{contador} = \text{tempo limite de recuperação}$), a recuperação da transferência parará e o agente de origem relatará a transferência como com falha. Esse tipo de falha de transferência, causado pelo fato de que a transferência atingiu o tempo limite de recuperação, é indicado pelo código de mensagem RECOVERY TIMEOUT (69). Outra mensagem de log de transferência é publicada no tópico SYSTEM.FTE, com uma sequência de tópicos de `Log/agent_name/transfer_ID`, para indicar que a transferência falhou e inclui uma mensagem, o código de retorno e o log de eventos do agente de origem. O log de eventos do agente de origem é atualizado com uma mensagem quando qualquer um dos eventos a seguir ocorre durante a recuperação:

- Quando o parâmetro de tempo limite de recuperação é configurado para um valor maior que **-1**, a transferência entra em recuperação. O log de eventos do agente é atualizado para indicar o início do cronômetro de recuperação para o **TransferId** e o período de tempo que o agente de origem aguarda antes de iniciar o processamento do tempo limite de recuperação.
- Quando a transferência de recuperação continua, o log de eventos do agente de origem é atualizado com uma nova mensagem para indicar que o **TransferId** que estava em recuperação continua.

- Quando uma transferência de recuperação atinge o tempo limite, o log de eventos do agente de origem é atualizado para indicar o **TransferId** que falhou durante a recuperação, devido ao tempo limite de recuperação.

Essas mensagens de log permitem que os usuários (assinantes e criadores de logs) identifiquem as transferências que falharam em razão do tempo limite de recuperação da transferência.

O contador para o tempo limite de recuperação está sempre no agente de origem. No entanto, se o agente de destino falhar ao receber informações do agente de origem a tempo, ele poderá enviar uma solicitação para o agente de origem para colocar a transferência em recuperação. Para uma transferência na qual a opção de tempo limite de recuperação está configurada, o agente de origem inicia o contador de tempo limite de recuperação ao receber a solicitação do agente de destino.

A manipulação manual ainda é necessária para transferências que não usam a opção de tempo limite de recuperação, as transferências com falha e as parcialmente concluídas.

Para conjuntos de transferência, em que uma única solicitação de transferência é emitida para múltiplos arquivos e alguns dos arquivos foram concluídos com sucesso, mas um foi concluído apenas parcialmente, a transferência ainda será marcada como com falha, uma vez que não foi concluída conforme o esperado. O agente de origem pode ter atingido o tempo limite ao transferir o arquivo parcialmente concluído.

Assegure-se de que o agente de destino e o servidor de arquivos estejam prontos e em um estado para aceitar transferências de arquivos.

Você deve emitir a solicitação de transferência novamente para o conjunto inteiro, mas para evitar problemas porque alguns dos arquivos permanecem no destino da tentativa de transferência inicial, é possível emitir a nova solicitação com a opção `sobrescrever se existente` especificada. Isso assegura que o conjunto incompleto de arquivos da tentativa de transferência anterior seja limpo como parte da nova transferência, antes de os arquivos serem gravados no destino novamente.

A partir do IBM MQ 9.1.5, não é mais necessário remover manualmente os arquivos de peça deixados em um destino após uma tentativa inicial de transferência ter falhado. Se um tempo limite de recuperação de transferência for configurado para uma transferência, o agente de origem moverá a transferência para o estado `RecoveryTimedOut` se a recuperação de transferência atingir o tempo limite. Depois que a transferência foi ressincronizada, o agente de destino remove qualquer arquivo de parte que foi criado durante a transferência e envia uma mensagem de conclusão para o agente de origem.

Rastreios e mensagens

Os pontos de rastreio são incluídos para propósitos de diagnóstico. O valor de tempo limite de recuperação, o início do intervalo de nova tentativa, o início do período de retomada e a reconfiguração do contador e se a transferência atingiu o tempo limite e falhou, são registrados. No caso de um problema ou comportamento inesperado, é possível coletar os arquivos de log e de rastreio de saída do agente de origem e fornecê-los quando solicitados pelo suporte da IBM, para ajudar com a resolução de problemas.

As mensagens notificam quando:

- Uma transferência entra em recuperação ([BFGTR0081I](#))
- Uma transferência é finalizada porque atingiu o tempo limite da recuperação ([BFGSS0081E](#))
- Uma transferência continua depois de estar em recuperação ([BFGTR0082I](#))

Conceitos relacionados

[“Recuperação e reinicialização do MFT” na página 334](#)

Se seu agente ou gerenciador de filas estiver indisponível por alguma razão, por exemplo, devido a uma falha de energia ou de rede, o Managed File Transfer se recuperará da seguinte forma nestes cenários:

Configurando o tempo limite de recuperação de transferência para todas as transferências de um agente de origem

É possível configurar um tempo limite de recuperação de transferência que se aplique a todas as transferências de um agente de origem, incluindo o parâmetro **transferRecoveryTimeout** no arquivo `agent.properties`.

Sobre esta tarefa

Para configurar um tempo limite de recuperação de transferência que se aplique a todas as transferências de um agente de origem, inclua o parâmetro e o par de valores do **transferRecoveryTimeout** no arquivo `agent.properties`.

Há três opções para o parâmetro **transferRecoveryTimeout**:

-1

O agente continuará tentando recuperar a transferência paralisada até que ela seja bem-sucedida. O uso dessa opção é equivalente ao comportamento padrão do agente quando a propriedade não está configurada.

0

O agente para a transferência de arquivos assim que ela entra em recuperação.

>0

O agente continuará tentando recuperar a transferência paralisada durante o período de tempo em segundos, conforme configurado pelo valor de número inteiro positivo especificado.

Qualquer mudança feita no arquivo `agent.properties` entra em vigor somente após o agente ser reiniciado.

Se necessário, é possível substituir o valor de tempo limite de recuperação de transferência no arquivo `agent.properties` para uma transferência individual. Para obter mais informações, consulte [“Configurando o tempo limite de recuperação de transferência de transferências individuais”](#) na página 339.

Procedimento

- Para especificar que o agente continue tentando recuperar a transferência paralisada até que ela seja concluída com sucesso, configure um valor de tempo limite de recuperação de transferência de -1, conforme mostrado no exemplo a seguir:

```
transferRecoveryTimeout=-1
```

- Para especificar que o agente marque a transferência como com falha imediatamente ao entrar em recuperação, configure um valor de tempo limite de recuperação de transferência de 0, conforme mostrado no exemplo a seguir:

```
transferRecoveryTimeout=0
```

- Para especificar que o agente continue tentando novamente uma transferência paralisada por um determinado período de tempo antes que a transferência seja marcada como com falha, configure um valor de tempo limite de recuperação de transferência para o período de tempo, em segundos, que você deseja que o agente continue tentando novamente.

Por exemplo, configurar um valor de tempo limite de recuperação de transferência de 21600 significa que o agente continuará tentando recuperar a transferência por seis horas a partir de quando entrar em recuperação:

```
transferRecoveryTimeout=21600
```

O valor máximo para esse parâmetro é 999999999.

Configurando o tempo limite de recuperação de transferência de transferências individuais

É possível configurar o tempo limite de recuperação de transferência de uma transferência individual por meio da linha de comandos, com o IBM MQ Explorer ou usando tarefas do Apache Ant. Se houver um valor de tempo limite de recuperação de transferência definido no arquivo `agent.properties`, configurar o tempo limite de recuperação de transferência para uma transferência individual substituirá o valor definido no arquivo `agent.properties`.

Sobre esta tarefa

É possível configurar o parâmetro de tempo limite de recuperação de transferência de uma transferência individual quando você está:

- Criando uma transferência usando o comando **fteCreateTransfer** ou usando o IBM MQ Explorer.
- Criando um modelo de transferência usando o comando **fteCreateTemplate** ou usando o IBM MQ Explorer.
- Criando um monitor de recurso usando o comando **fteCreateMonitor** ou usando o IBM MQ Explorer.
- Copiando ou movendo arquivos usando as tarefas `fte:filecopy` ou `fte:filemove` do Ant.

Se você configurar um valor de tempo limite de recuperação de transferência para uma transferência individual, este valor substituirá o valor de tempo limite de recuperação de transferência configurado no arquivo `agent.properties` (consulte [“Configurando o tempo limite de recuperação de transferência para todas as transferências de um agente de origem”](#) na página 338).

Procedimento

- Para usar o comando **fteCreateTransfer** ou **fteCreateTemplate** para configurar o tempo limite de recuperação de transferência, especifique a opção apropriada para o parâmetro **-rt**:

-1

O agente continuará tentando recuperar a transferência paralisada até que ela seja bem-sucedida. O uso dessa opção é equivalente ao comportamento padrão do agente quando a propriedade não está configurada.

0

O agente para a transferência de arquivos assim que ela entra em recuperação.

>0

O agente continua tentando recuperar a transferência paralisada para o período de tempo especificado em segundos.

Exemplos para o comando **fteCreateTransfer**

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt -1 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 0 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

Exemplos para o comando **fteCreateTemplate**

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt -1 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 0 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 21600 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

Não há parâmetro **-rt** para o comando **fteCreateMonitor**. Se você configurar o parâmetro **-rt** com o comando **fteCreateTransfer** e também configurar o parâmetro **-gt**, o parâmetro de tempo limite de recuperação será incluído no documento XML com a definição de transferência gerada ao executar o comando **fteCreateTransfer**. Em seguida, o monitor de recurso usará esse documento XML quando você executar o comando **fteCreateMonitor**. No exemplo a seguir, os detalhes do tempo limite de recuperação de transferência seriam incluídos no arquivo `task.xml`:

```
fteCreateMonitor -ma AgentName -md C:\mqmft\monitors -mn Monitor_Name -mt task.xml -tr
"fileSize>=5MB,*.zip"
```

- Para usar a página do assistente Nova transferência, Novo Monitor ou Novo modelo do IBM MQ Explorer para configurar o tempo limite de recuperação de transferência, selecione a opção necessária no campo **Tempo limite de recuperação de transferência** (segundos):

Como agente de origem

Se você selecionar **Como agente de origem**, o valor do parâmetro **transferRecoveryTimeout** do arquivo `agent.properties` será usado se ele estiver configurado, caso contrário, o comportamento padrão do tempo limite de recuperação de transferência será aplicado.

Caixa de listagem numérica

Se você inserir um tempo em segundos na caixa de listagem numérica, o agente continuará tentando recuperar a transferência paralisada para o período de tempo especificado.

Nenhum

Se você selecionar **Nenhum**, nenhum tempo limite de recuperação de transferência será configurado e o agente continuará tentando recuperar a transferência paralisada até que ela seja bem-sucedida.

- Para configurar o tempo limite de recuperação usando as tarefas do Ant, inclua a opção e o valor **transferRecoveryTimeout** com os elementos **fte:filecopy** ou **fte:filemove** para mover ou copiar arquivos, por exemplo:

Exemplo para **fte:filecopy**

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result" transferRecoveryTimeout="0">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filecopy>
```

Exemplo para **fte:filemove**

```
<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>
```

Conceitos relacionados

[“Recuperação e reinicialização do MFT” na página 334](#)

Se seu agente ou gerenciador de filas estiver indisponível por alguma razão, por exemplo, devido a uma falha de energia ou de rede, o Managed File Transfer se recuperará da seguinte forma nestes cenários:

Tarefas relacionadas

[“Iniciando uma Nova Transferência de Arquivos” na página 220](#)

É possível iniciar uma nova transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos e escolher transferir um único arquivo ou vários arquivos em um grupo.

[“Criando um modelo de transferência de arquivos usando o IBM MQ Explorer” na página 265](#)

É possível criar um modelo de transferência de arquivos a partir do IBM MQ Explorer ou da linha de comandos. Você pode usar esse modelo para criar novas transferências de arquivos, usando os detalhes dos modelos, ou submetê-los para iniciar a transferência de arquivos.

[“Monitorando recursos do MFT” na página 232](#)

É possível monitorar recursos do Managed File Transfer; por exemplo, uma fila ou um diretório. Quando uma condição nesse recurso é satisfeita, o monitor de recurso inicia a tarefa, como uma transferência de arquivos. É possível criar um monitor de recurso usando o comando **fteCreateMonitor** ou a visualização **Monitores** no plug-in do Managed File Transfer para o IBM MQ Explorer.

[“Visualizando o Status de Transferências de Arquivos no Log de Transferência” na página 229](#)

É possível visualizar os detalhes de transferências de arquivos usando o **Log de transferência** no IBM MQ Explorer. Essas transferências podem ser aquelas iniciadas da linha de comandos ou do IBM MQ Explorer. Também é possível customizar o que é exibido no **Log de Transferência**.

Referências relacionadas

O arquivo MFT [agent.properties](#)

fteCreateTransfer: iniciar uma nova transferência de arquivos

fteCreateTemplate: criar novo modelo de transferência de arquivos

fteCreateMonitor: criar um monitor de recurso do MFT

Tarefa Ant [fte:filecopy](#)

Tarefa Ant [fte:filemove](#)

Windows

Linux

AIX

Administrando MQ Telemetry

O MQ Telemetry é administrado usando o IBM MQ Explorer ou uma linha de comandos. Use o explorador para configurar canais de telemetria, controlar o serviço de telemetria e monitorar os clientes MQTT que estão conectados ao IBM MQ. Configure a segurança do MQ Telemetry usando o JAAS, o TLS e o gerenciador de autoridade de objeto do IBM MQ.

Administrando Usando o IBM MQ Explorer

Use o explorador para configurar canais de telemetria, controlar o serviço de telemetria e monitorar os clientes MQTT que estão conectados ao IBM MQ. Configure a segurança do MQ Telemetry usando o JAAS, o TLS e o gerenciador de autoridade de objeto do IBM MQ.

Administrando Usando a Linha de Comandos

O MQ Telemetry pode ser completamente administrado na linha de comandos [usando comandos MQSC](#).

A documentação do MQ Telemetry também possui scripts de amostra que demonstram o uso básico do aplicativo cliente IBM MQ Telemetry Transport v3.

Leia e entenda as amostras em [IBM MQ Telemetry Transport programas de amostra](#) antes de usá-las..

Conceitos relacionados

[MQ Telemetry](#)

Referências relacionadas

[Propriedades MQXR](#)

Linux

AIX

Configurando um gerenciador de filas para telemetria no Linux e AIX

Siga estas etapas para configurar o MQ Telemetry manualmente. Se você precisar apenas de uma configuração simples que use o ID do usuário guest, será possível executar o assistente de suporte MQ Telemetry em IBM MQ Explorer.

Antes de começar

Se você precisar apenas de uma configuração simples, considere o uso do suporte MQ Telemetry em IBM MQ Explorer. Esse suporte inclui um assistente e um procedimento de comando de amostra `sampleMQM`. Esses recursos definem uma configuração inicial usando o ID do usuário `guest`. Consulte [Verificando a instalação de MQ Telemetry usando IBM MQ Explorer](#) e [IBM MQ Telemetry Transport programas de amostra](#).

Se você precisar de uma configuração mais complexa que use um método de autenticação diferente, use as etapas nesta tarefa. Inicie com as seguintes etapas iniciais:

1. Consulte [Considerações de instalação para MQ Telemetry](#) para obter informações sobre como instalar IBM MQ e o recurso MQ Telemetry.
2. Crie e inicie um gerenciador de filas. O gerenciador de filas é referido como `qMgr` nesta tarefa.
3. Como parte desta tarefa, você configurará o serviço de telemetria (MQXR). As configurações de propriedade MQXR são armazenadas em um arquivo de propriedades específicas da plataforma: `mqxr_win.properties`. Normalmente, você não precisa editar o arquivo de propriedades MQXR diretamente porque praticamente todas as configurações podem ser definidas por meio de comandos `admin` do MQSC ou IBM MQ Explorer. Se você decidir editar o arquivo diretamente, pare o gerenciador de filas antes de fazer suas mudanças. Consulte [MQXR propriedades](#).

Sobre esta tarefa

Siga as etapas nesta tarefa para configurar o MQ Telemetry manualmente, usando diferentes esquemas de autorização.

Procedimento

1. Abra uma janela de comando no diretório de amostra de telemetria.

O diretório de amostras de telemetria é `/opt/mqm/mqxr/samples`.

2. Crie a fila de transmissão de telemetria.

Se o `SYSTEM.MQTT.TRANSMIT.QUEUE` não existir, ele será criado automaticamente quando o serviço de telemetria (MQXR) for iniciado pela primeira vez e configurado para usar o ID do usuário `guest`. Entretanto, essa tarefa configura MQ Telemetry para usar um esquema de autorização diferente. Para essa tarefa, você cria o `SYSTEM.MQTT.TRANSMIT.QUEUE` e configura o acesso a ele antes de iniciar o serviço de telemetria (MQXR).

Execute o seguinte comando:

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

3. Configure a fila de transmissão padrão.

É mais fácil enviar mensagens diretamente para os clientes MQTT se `SYSTEM.MQTT.TRANSMIT.QUEUE` for a fila de transmissão padrão. Caso contrário, será necessário incluir uma definição de fila remota para cada cliente que receber mensagens do IBM MQ; consulte [“Enviando uma mensagem para um cliente diretamente”](#) na página 347. Observe que alterar a fila de transmissão padrão pode interferir na configuração existente.

Quando o serviço de telemetria (MQXR) é iniciado pela primeira vez, ele não configura `SYSTEM.MQTT.TRANSMIT.QUEUE` como a fila de transmissão padrão para o gerenciador de filas. Para definir essa configuração, você altera a propriedade da fila de transmissão padrão. Você faz isso usando o IBM MQ Explorer ou executando o comando a seguir:

```
echo "ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE')" | runmqsc qMgr
```

4. Siga um procedimento em [“Autorizando clientes do MQTT a acessar objetos do IBM MQ”](#) na página 349 para criar um ou mais IDs de usuário. Os IDs de usuário têm a autoridade para publicar, assinar e enviar publicações para clientes do MQTT.
5. Edite o arquivo `installMQXRService_unix.mqsc` para configurar o arquivo-chave que é usado para criptografar passphrase para canais TLS do MQTT:
 - a) Abra o arquivo `WMQ program installation directory/mqxr/samples/installMQXRService_unix.mqsc`.
 - b) Localize a linha que inclui o parâmetro **STARTARG** e edite a opção **-sf** para especificar o local de seu arquivo-chave de credenciais
 Por padrão, o arquivo `installMQXRService_unix.mqsc` usa um arquivo-chave padrão denominado [DEFAULT]. O arquivo-chave padrão é o mesmo para todas as instalações do IBM MQ, portanto, deve-se fornecer um arquivo-chave que seja exclusivo para sua instalação ao criptografar passphrases..

 Consulte também o código de exemplo em [“Criando o SYSTEM.MQXR.SERVICE”](#) na página 343
6. Instale o serviço de telemetria (MQXR) executando o comando a seguir:

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc queue_manager
```

7. Inicie o serviço.

```
echo "START SERVICE(SYSTEM.MQXR.SERVICE)" | runmqsc qMgr
```

O serviço de telemetria (MQXR) é iniciado automaticamente quando o gerenciador de filas é iniciado. Ele é iniciado manualmente nesta tarefa porque o gerenciador de filas já está em execução.

8. Usando o IBM MQ Explorer, configure canais de telemetria para aceitar conexões de clientes do MQTT.
 Os canais de telemetria devem ser configurados de forma que suas identidades sejam um dos IDs do usuário definidos na etapa “4” na página 343
 Consulte também [DEFINE CHANNEL \(MQTT\)](#).
9. Verifique a configuração executando o cliente de amostra.

Para o cliente de amostra trabalhar com seu canal de telemetria, o canal deve autorizar o cliente a publicar, inscrever-se e receber publicações. O cliente de amostra se conecta ao canal de telemetria na porta 1883 por padrão. Consulte também [Programas de amostra do IBM MQ Telemetry Transport](#).

Criando o SYSTEM.MQXR.SERVICE

Use o comando **runMQXRService** para criar o SYSTEM.MQXR.SERVICE

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+" -sf "/home/keyFileLocation/keyFile.txt"') +
STOPCMD('+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+/mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+/mqxr.stderr')
```

Windows

Configurando um gerenciador de filas para telemetria no Windows

Siga estas etapas para configurar o MQ Telemetry manualmente. Se você precisar apenas de uma configuração simples que use o ID do usuário guest, será possível executar o assistente de suporte MQ Telemetry em IBM MQ Explorer.

Antes de começar

Se você precisar apenas de uma configuração simples, considere o uso do suporte MQ Telemetry em IBM MQ Explorer. Esse suporte inclui um assistente e um procedimento de comando de amostra `sampleMQM`. Esses recursos definem uma configuração inicial usando o ID do usuário `guest`. Consulte [Verificando a instalação de MQ Telemetry usando IBM MQ Explorer](#) e [IBM MQ Telemetry Transport programas de amostra](#).

Se você precisar de uma configuração mais complexa que use um método de autenticação diferente, use as etapas nesta tarefa. Inicie com as seguintes etapas iniciais:

1. Consulte [Considerações de instalação para MQ Telemetry](#) para obter informações sobre como instalar IBM MQ e o recurso MQ Telemetry.
2. Crie e inicie um gerenciador de filas. O gerenciador de filas é referido como `qMgr` nesta tarefa.
3. Como parte desta tarefa você configurará o serviço de telemetria (MQXR). As configurações de propriedade MQXR são armazenadas em um arquivo de propriedades específicas da plataforma: `mqxr_win.properties`. Normalmente você não precisa editar o arquivo de propriedades MQXR diretamente porque praticamente todas as configurações podem ser definidas por meio de comandos `admin` do MQSC ou IBM MQ Explorer. Se você decidir editar o arquivo diretamente, pare o gerenciador de filas antes de fazer suas mudanças. Consulte [MQXR propriedades](#).

Sobre esta tarefa

Siga as etapas nesta tarefa para configurar o MQ Telemetry manualmente, usando diferentes esquemas de autorização.

Procedimento

1. Abra uma janela de comando no diretório de amostra de telemetria.

O diretório de amostras de telemetria é `WMQ program installation directory\mqxr\samples`.

2. Crie a fila de transmissão de telemetria.

Se o `SYSTEM.MQTT.TRANSMIT.QUEUE` não existir, ele será criado automaticamente quando o serviço de telemetria (MQXR) for iniciado pela primeira vez e configurado para usar o ID do usuário `guest`. Entretanto, essa tarefa configura MQ Telemetry para usar um esquema de autorização diferente. Para essa tarefa, você cria o `SYSTEM.MQTT.TRANSMIT.QUEUE` e configura o acesso a ele antes de iniciar o serviço de telemetria (MQXR).

Execute o seguinte comando:

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

3. Configure a fila de transmissão padrão.

É mais fácil enviar mensagens diretamente para os clientes MQTT se `SYSTEM.MQTT.TRANSMIT.QUEUE` for a fila de transmissão padrão. Caso contrário, será necessário incluir uma definição de fila remota para cada cliente que receber mensagens do IBM MQ; consulte [“Enviando uma mensagem para um cliente diretamente”](#) na página 347. Observe que alterar a fila de transmissão padrão pode interferir na configuração existente.

Quando o serviço de telemetria (MQXR) é iniciado pela primeira vez, ele não configura `SYSTEM.MQTT.TRANSMIT.QUEUE` como a fila de transmissão padrão para o gerenciador de filas. Para definir essa configuração, você altera a propriedade da fila de transmissão padrão. Você faz isso usando o IBM MQ Explorer ou executando o comando a seguir:

```
echo ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```


4. Siga um procedimento em [“Autorizando clientes do MQTT a acessar objetos do IBM MQ”](#) na página 349 para criar um ou mais IDs de usuário. Os IDs de usuário têm a autoridade para publicar, assinar e enviar publicações para clientes do MQTT.
5. Edite o arquivo `installMQXRService_win.mqsc` para configurar o arquivo-chave que é usado para criptografar passphrase para canais TLS do MQTT:
 - a) Abra o arquivo `WMQ program installation directory\mqxr\samples\installMQXRService_win.mqsc`.
 - b) Localize a linha que inclui o parâmetro **STARTARG** e edite a opção **-sf** para especificar o local de seu arquivo-chave de credenciais
 Por padrão, o arquivo `installMQXRService_win.mqsc` usa um arquivo-chave padrão denominado [DEFAULT] O arquivo-chave padrão é o mesmo para todas as instalações do IBM MQ, portanto, deve-se fornecer um arquivo-chave que seja exclusivo para sua instalação ao criptografar passphrases..

 Consulte também o código de exemplo em [“Criando SYSTEM.MQXR.SERVICE”](#) na página 345
6. Instale o serviço de telemetria (MQXR) executando o comando a seguir:

```
type installMQXRService_win.mqsc | runmqsc qMgr
```

7. Inicie o serviço.

```
echo START SERVICE(SYSTEM.MQXR.SERVICE) | runmqsc qMgr
```

O serviço de telemetria (MQXR) é iniciado automaticamente quando o gerenciador de filas é iniciado. Ele é iniciado manualmente nesta tarefa porque o gerenciador de filas já está em execução.

8. Usando o IBM MQ Explorer, configure canais de telemetria para aceitar conexões de clientes do MQTT.

Os canais de telemetria devem ser configurados de forma que suas identidades sejam um dos IDs do usuário definidos na etapa “4” na página 345

Consulte também [DEFINE CHANNEL \(MQTT\)](#).

9. Verifique a configuração executando o cliente de amostra.

Para o cliente de amostra trabalhar com seu canal de telemetria, o canal deve autorizar o cliente a publicar, subscrever-se e receber publicações. O cliente de amostra se conecta ao canal de telemetria na porta 1883 por padrão. Consulte também [Programas de amostra do IBM MQ Telemetry Transport](#).

Criando SYSTEM.MQXR.SERVICE

Use o comando `runMQXRService` para criar o SYSTEM.MQXR.SERVICE

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+\mqxr\bin\runMQXRService.bat') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\" -g "+MQ_DATA_PATH+\" -sf
"c:\keyFileLocation\keyFile.txt"') +
STOPCMD('+MQ_INSTALL_PATH+\mqxr\bin\endMQXRService.bat') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+\mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+\mqxr.stderr')
```

Windows Linux AIX Configurando o enfileiramento distribuído para enviar mensagens aos clientes do MQTT

Os aplicativos IBM MQ podem enviar mensagens dos clientes do MQTT v3 publicando para assinatura criada por um cliente ou enviando uma mensagem diretamente. Qualquer que seja o método usado, a mensagem será colocada em SYSTEM.MQTT.TRANSMIT.QUEUE e enviada para o cliente

pelo serviço de telemetria (MQXR). Há inúmeras maneiras de se colocar uma mensagem em SYSTEM.MQTT.TRANSMIT.QUEUE.

Publicando uma mensagem em resposta a uma assinatura do cliente MQTT

O serviço de telemetria (MQXR) cria uma assinatura em nome do cliente MQTT. O cliente é o destino para quaisquer publicações que correspondem à assinatura enviada pelo cliente. Os serviços de telemetria encaminham publicações correspondentes de volta para o cliente.

Um cliente MQTT está conectado ao IBM MQ como um gerenciador de filas, com seu nome do gerenciador de filas definido para o *ClientIdentifier*. O destino para publicações serem enviadas para o cliente é uma fila de transmissão, SYSTEM.MQTT.TRANSMIT.QUEUE. O serviço de telemetria encaminha mensagens nos clientes SYSTEM.MQTT.TRANSMIT.QUEUE para MQTT, usando o nome do gerenciador de filas de destino como a chave de um cliente específico

O serviço de telemetria (MQXR) abre a fila de transmissão usando *ClientIdentifier* como o nome do gerenciador de filas. O serviço de telemetria (MQXR) transmite o identificador de objeto da fila para o MQSUB chamada, para encaminhar as publicações que correspondem à assinatura do cliente. Na resolução de nome de objeto, o *ClientIdentifier* é criado como o nome do gerenciador de filas remotas e a fila de transmissão deve ser resolvida para SYSTEM.MQTT.TRANSMIT.QUEUE. Usando o nome do objeto de resolução padrão do IBM MQ, *ClientIdentifier* é resolvido como a seguir. Consulte Tabela 16 na página 346.

1. *ClientIdentifier* corresponde a nada.

ClientIdentifier é um nome do gerenciador de filas remotas. Ele não corresponde ao nome do gerenciador de filas locais, ao alias do gerenciador de filas ou a um nome da fila de transmissão.

O nome da fila não é definido. Atualmente, o serviço de telemetria (MQXR) configura SYSTEM.MQTT.PUBLICATION.QUEUE como o nome da fila. Um cliente MQTT v3 não suporta filas, portanto, o nome da fila resolvido é ignorado pelo cliente.

A propriedade do gerenciador de filas locais, Fila de transmissão padrão, o nome deve ser configurado como SYSTEM.MQTT.TRANSMIT.QUEUE, para que a publicação seja colocada em SYSTEM.MQTT.TRANSMIT.QUEUE para ser enviada ao cliente.

2. *ClientIdentifier* corresponde a um alias de gerenciador de filas chamado *ClientIdentifier*.

ClientIdentifier é um nome do gerenciador de filas remotas. Ele corresponde ao nome de um alias de gerenciador de filas.

O alias do gerenciador de filas deve ser definida com *ClientIdentifier* como o nome do gerenciador de filas remotas.

Durante a configuração do nome da fila de transmissão na definição de alias do gerenciador de filas, não é necessário que a transmissão padrão seja configurada como SYSTEM.MQTT.TRANSMIT.QUEUE.

Tabela 16. Resolução de nome de um alias do gerenciador de filas do MQTT					
	Entrada		Saída		
<i>ClientIdentifier</i>	Nome do gerenciador de filas	Nome da fila	Nome do gerenciador de filas	Nome da fila	Fila de transmissão
Não corresponde a nada	<i>ClientIdentifier</i>	<i>undefined</i>	<i>ClientIdentifier</i>	<i>undefined</i>	Fila de transmissão padrão. SYSTEM.MQTT.TRANSMIT.QUEUE

Tabela 16. Resolução de nome de um alias do gerenciador de filas do MQTT (continuação)

	Entrada		Saída		
<i>ClientIdentifier</i>	Nome do gerenciador de filas	Nome da fila	Nome do gerenciador de filas	Nome da fila	Fila de transmissão
Corresponde a um alias do gerenciador de filas denominado <i>ClientIdentifier</i>	<i>ClientIdentifier</i>	<i>undefined</i>	<i>ClientIdentifier</i>	<i>undefined</i>	SYSTEM.MQTT.TRANSMIT.QUEUE

Para obter mais informações sobre a resolução de nome, consulte [Resolução de Nome](#).

Qualquer programa IBM MQ pode publicar para o mesmo tópico. A publicação é enviada para seus assinantes, incluindo MQTT v3 os clientes que possuem uma assinatura para o tópico.

Se um tópico administrativo for criado em um cluster, com o atributo CLUSTER(*clusterName*), qualquer aplicativo no cluster poderá publicar no cliente; por exemplo:

```
echo DEFINE TOPIC('MQTTExamples') TOPICSTR('MQTT Examples') CLUSTER(MQTT) REPLACE | runmqsc qMgr
```

Nota: Não dê à SYSTEM.MQTT.TRANSMIT.QUEUE um atributo de cluster.

Assinantes e publicadores do cliente do MQTT podem se conectar a gerenciadores de filas diferentes. Os assinantes e os publicadores podem fazer parte do mesmo ou cluster ou ser conectados por uma hierarquia de publicação/assinatura. A publicação é entregue do publicador para o assinante utilizando o IBM MQ.

Enviando uma mensagem para um cliente diretamente

Uma alternativa para um cliente criar uma assinatura e receber uma publicação que corresponde ao tópico de assinatura, enviar uma mensagem para um cliente MQTT v3 diretamente. Aplicativos clientes do MQTT V3 não podem enviar mensagens diretamente, mas outros aplicativos, como o IBM MQ, podem.

O aplicativo IBM MQ deve conhecer o *ClientIdentifier* do cliente MQTT v3. Como os clientes MQTT v3 não têm filas, o nome da fila de destino é transmitido para o método MQTT v3 aplicativo cliente *messageArrived* como um nome de tópico. Por exemplo, em um programa MQI, crie um descritor de objeto com o cliente como *ObjectQmgrName*:

```
MQOD.ObjectQmgrName = ClientIdentifier ;
MQOD.ObjectName = name ;
```

Se o aplicativo é gravado usando as JMS, crie um destino ponto-a-ponto ; por exemplo:

```
JM 3.0
jakarta.jms.Destination jmsDestination =
(jakarta.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifier/name");
```

```
JMS 2.0
javax.jms.Destination jmsDestination =
(javax.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifier/name");
```

Para enviar uma mensagem não solicitada para um cliente MQTT, use uma definição de fila remota. O nome do gerenciador de filas remotas deve ser resolvido para *ClientIdentifier* do cliente. A fila de

transmissão deve ser resolvida como SYSTEM.MQTT.TRANSMIT.QUEUE; veja Tabela 17 na página 348. O nome da fila remota pode ser qualquer ou oisa. O cliente o reou ebe como uma sequência de tópicos.

Entrada		Saída		
Nome da fila	Nome do gerenciador de filas	Nome da fila	Nome do gerenciador de filas	Fila de transmissão
Nome da definição de fila remota	Em branco ou nome do gerenciador de filas locais	Nome da fila remota usado como uma sequência de tópicos	<i>ClientIdentifier</i>	SYSTEM.MQTT.TRANSMIT.QUEUE

Se o cliente estiver conectado, a mensagem será enviada diretamente ao cliente MQTT, que chama o método `messageArrived`; consulte [messageArrived method](#).

Se o cliente tiver desconectado com uma sessão persistente, a mensagem será armazenada em SYSTEM.MQTT.TRANSMIT.QUEUE; Consulte [MQTT sessões stateless e stateful](#) Ela é encaminhada para o cliente quando ele se reconecta novamente à sessão.

Se você enviar uma mensagem não persistente, ela será enviada para o cliente com qualidade de serviço no máximo uma vez, $QoS=0$. Se você enviar uma mensagem persistente diretamente para um cliente, por padrão, ela será enviada com qualidade de serviço exatamente uma vez, $QoS=2$. Como o cliente pode não ter um mecanismo de persistência, ele pode reduzir a qualidade de serviço que ele aceita para mensagens enviadas diretamente. Para reduzir a qualidade de serviço para mensagens enviadas diretamente a um cliente, faça uma assinatura para o tópico DEFAULT.QoS. Especifique a qualidade de serviço máxima que o cliente pode suportar.

Windows Linux AIX Identificação, autorização e autenticação do cliente do MQTT

O serviço de telemetria (MQXR) publica ou assina tópicos do IBM MQ em nome de clientes do MQTT, utilizando ou não do MQTT. O administrador do IBM MQ configura a identidade do canal do MQTT que é usada para autorização do IBM MQ. O administrador pode definir uma identidade comum para o canal ou usar `Username` ou `ClientIdentifier` de um cliente conectado ao canal.

O serviço de telemetria (MQXR) pode autenticar o cliente usando o `Username` fornecido pelo cliente ou usando um certificado de cliente. O `Username` é autenticado usando uma senha fornecida pelo cliente.

Resumindo: Identificação de cliente é a seleção da identidade do cliente. Dependendo do contexto, o cliente é identificado por `ClientIdentifier`, `Username`, uma identidade de cliente comum criada pelo administrador ou um certificado de cliente. O identificador de cliente usado para verificação de autenticidade não precisa ser o mesmo identificador que é usado para autorização.

Os programas do cliente MQTT configuram `Nome do usuário` e `Senha` que são enviados para o servidor utilizando um canal do MQTT. Eles também podem configurar as propriedades de TLS necessárias para criptografar e autenticar a conexão. O administrador decide se autentica o canal do MQTT e como autenticar o canal.

Para autorizar um cliente MQTT a acessar objetos do IBM MQ, autorize o `ClientIdentifier` ou `Username` do cliente ou autorize uma identidade de cliente comum. Para permitir que um cliente se conecte ao IBM MQ, autentique `Username` ou use um certificado de cliente. Configure o JAAS para autenticar o `Nome do usuário` e configure o TLS para autenticar um certificado de cliente.

Se você configurar uma `Senha` no cliente, criptografe a conexão usando VPN ou configure o canal MQTT para usar TLS, para manter a senha privada.

É difícil gerenciar certificados de cliente. Por essa razão, se os riscos associados à autenticação de senha forem aceitáveis, a autenticação de senha será usada na maioria das vezes para autenticar clientes.

Se houver uma maneira segura de gerenciar e armazenar o certificado de cliente, é possível contar com a autenticação de certificado. Porém, raramente os certificados podem ser gerenciados com segurança nos tipos de ambientes em que a telemetria é usada. Em vez disso, a autenticação dos dispositivos usando certificados de cliente é complementada pela autenticação de senhas de clientes no servidor. Devido a essa complexidade adicional, o uso de certificado de cliente é restrito a dispositivos altamente sensíveis. O uso de duas formas de autenticação é chamado de autenticação de dois fatores. Deve-se conhecer um dos fatores, como uma senha, e ter o outro, como um certificado.

Em um aplicativo altamente sensível, como um dispositivo chip-and-pin, o dispositivo é bloqueado durante a fabricação para evitar violação com hardware e software internos. Um certificado de cliente confiável com tempo limitado é copiado no dispositivo. O dispositivo é implementado no local onde deve ser usado. Outras autenticações são feitas ou da vez que o dispositivo é usado, por meio de senha ou de outro certificado do dispositivo smart ou rd.

Windows

Linux

AIX

Identidade e autorização do cliente do MQTT

Use o ID do cliente, Username ou uma identidade de cliente comum para autorização para acessar objetos do IBM MQ.

O administrador do IBM MQ tem três opções para selecionar a identidade do canal do MQTT. O administrador faz a escolha ao definir ou modificar o canal MQTT usado pelo cliente. A identidade é usada para autorizar o acesso aos tópicos do IBM MQ. A opção é feita na seguinte ordem:

1. O ID do cliente (consulte [USECLNTID](#)).
2. Uma identidade que o administrador fornece para o canal (o MCAUSER do canal. Consulte [MCAUSER](#)).
3. Se nenhuma das opções anteriores se aplica, o Username passado do cliente MQTT (Username é um atributo da classe `MqttConnectOptions`. Ele deve ser configurado antes de o cliente se conectar ao serviço. Seu valor padrão é nulo).

Evite problemas: A identidade escolhida por este processo é doravante referida, por exemplo, pelo comando `DISPLAY CHSTATUS (MQTT)`, como o MCAUSER do cliente. Esteja ciente de que este não é necessariamente a mesma identidade do MCAUSER do canal que é referido na opção (2).

Use o comando IBM MQ `setmqaut` para selecionar quais objetos e quais ações, estão autorizados para serem usados pela identidade associada ao canal do MQTT. Por exemplo, o código a seguir autoriza uma identidade de canal `MQTTClient`, fornecida pelo administrador do gerenciador de filas QM1:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

Windows

Linux

AIX

Autorizando clientes do MQTT a acessar objetos do

IBM MQ

Siga estas etapas para autorizar clientes do MQTT a publicar e assinar objetos do IBM MQ. As etapas seguem quatro padrões de controle de acesso alternativos.

Antes de começar

Os clientes estão autorizados a acessar os objetos do MQTT no IBM MQ quando uma identidade é designada a eles ao se conectam a um canal de telemetria. O administrador do IBM MQ configura o canal de telemetria usando o IBM MQ Explorer para fornecer um cliente um dos três tipos de identidade:

1. `ClientIdentifier`
2. Nome de usuário
3. Um nome que o administrador designa ao canal.

Seja qual for o tipo utilizado, a identidade deverá ser definida para o IBM MQ como um proprietário pelo serviço de autorização instalado. O serviço de autorização padrão no Windows ou Linux é chamado Object Authority Manager (OAM). Se você estiver usando o OAM, a identidade deverá ser definida como um ID do usuário.

Utilize a identidade para dar a um cliente ou coleção de clientes, permissão para publicar ou assinar tópicos definidos no IBM MQ. Se um MQTT cliente foi inscrito para um tópico, utilize a identidade para dar a ele permissão para receber as publicações resultantes.

É difícil de gerenciar um sistema com dezenas de milhares de clientes MQTT, cada uma requerendo permissões de acesso individual. Uma solução é definir identidades comuns e associar clientes individuais do MQTT com uma das identidades comuns. Defina quantas identidades comuns forem necessárias para definir diferentes combinações de permissões. Outra solução é gravar seu próprio serviço de autorização que possa lidar com milhares de usuários com mais facilidade do que o sistema operacional.

É possível combinar clientes MQTT em identidades comuns de duas maneiras, usando o OAM:

1. Defina diversos canais de telemetria, cada um com um ID de usuário diferente que o administrador aloca usando o IBM MQ Explorer. Clientes que se conectam usando números de porta TCP/IP diferentes são associados a diferentes canais de telemetria e recebem identidades diferentes.
2. Defina um único canal de telemetria, mas cada cliente deve selecionar um Username de um pequeno conjunto de IDs de usuário. O administrador configura o canal de telemetria para selecionar o cliente Username como sua identidade.

Nesta tarefa, a identidade do canal de telemetria é chamada de *mqttUser*, independentemente de como ele está definido. Se as coleções de clientes utilizam diferentes identidades, utilize vários do *mqttUsers*, um para cada coleção de clientes. Como a tarefa usa o OAM, cada *mqttUser* deve ser um ID de usuário.

Sobre esta tarefa

Nesta tarefa, você tem quatro padrões de controle de acesso que podem ser padronizados para requisitos específicos. Os padrões são diferentes em termos de granularidade de controle de acesso.

- [“Nenhum Controle de Acesso” na página 350](#)
- [“Controle de Acesso de Alta Granularidade” na página 350](#)
- [“Controle de Acesso de Média Granularidade” na página 351](#)
- [“Controle de acesso de baixa granularidade” na página 351](#)

O resultado dos modelos é designar conjuntos de permissões *mqttUsers* para publicar e assinar o IBM MQ e receber publicações do IBM MQ.

Nenhum Controle de Acesso

Aos clientes do MQTT é dada autoridade administrativa do IBM MQ e podem executar qualquer ação em qualquer objeto.

Procedimento

1. Crie um ID do usuário *mqttUser* para agir como a identidade de todos os clientes do MQTT.
2. Inclua *mqttUser* no grupo *mqm*. Consulte [Incluindo um usuário em um grupo no Windows](#) ou [Criando e gerenciando grupos no Linux](#)

Controle de Acesso de Alta Granularidade

Os clientes do MQTT têm autoridade para publicar e assinar e para enviar mensagens para clientes do MQTT. Eles não têm autoridade para executar outras ações ou para acessar outros objetos.

Procedimento

1. Crie um ID do usuário *mqttUser* para agir como a identidade de todos os clientes do MQTT.
2. Autorize *mqttUser* a publicar e assinar todos os tópicos e para enviar publicações para clientes do MQTT.

```
setmqaut -m qMgr -t topic -n SYSTEM.BASE.TOPIC -p mqttUser -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqttUser -all +put
```

Controle de Acesso de Média Granularidade

Os clientes do MQTT são divididos em grupos diferentes para publicar e assinar diferentes conjuntos de tópicos e enviar mensagens para clientes do MQTT.

Procedimento

1. Crie IDs de usuários múltiplos, *mqttUser* e vários tópicos administrativos na árvore de tópicos de publicação/assinatura.
2. Autorize diferente *mqttUsers* para diferentes tópicos.

```
setmqaut -m qMgr -t topic -n topic1 -p mqttUserA -all +pub +sub  
setmqaut -m qMgr -t topic -n topic2 -p mqttUserB -all +pub +sub
```

3. Crie um grupo *mqtt* e inclua todos os *mqttUsers* no grupo.
4. Autorize *mqtt* para enviar tópicos para clientes do MQTT .

```
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Controle de acesso de baixa granularidade

MQTT os clientes são incorporados em um sistema existente de controle de acesso, que autoriza grupos para executar ações em objetos.

Sobre esta tarefa

Um ID do usuário é designado a um ou mais grupos de sistemas operacionais, dependendo das autorizações que requer. Se os aplicativos do IBM MQ estão publicando e assinando o mesmo espaço de tópico como clientes MQTT, use esse modelo. Os grupos são chamados de Publish X, Subscribe Y e mqtt

Publish X

Membros de grupos Publish X podem publicar em *topicX*.

Subscribe Y

Membros de grupos do Subscribe Y podem assinar *topicY*.

mqtt

Os membros do grupo *mqtt* pode enviar publicações para clientes do MQTT.

Procedimento

1. Crie diversos grupos Publish X e Subscribe Y que são alocados para diversos tópicos administrativos na árvore de tópicos de publicação / assinatura.
2. Crie um grupo mqtt.
3. Crie IDs de usuários múltiplos, *mqttUser* e inclua os usuários a qualquer um dos grupos, dependendo do que eles estão autorizados a fazer.
4. Autorize diferentes grupos Publish X e Subscribe X para diferentes tópicos e autorize o grupo *mqtt* a enviar mensagens para clientes MQTT ..

```
setmqaut -m qMgr -t topic -n topic1 -p Publish X -all +pub  
setmqaut -m qMgr -t topic -n topic1 -p Subscribe X -all +pub +sub  
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Windows Linux AIX Autenticação de cliente do MQTT usando uma senha

Autentique Username usando a senha do cliente. É possível autenticar o cliente usando uma identidade diferente da usada para autorizar o cliente para publicar e assinar tópicos.

O serviço de telemetria (MQXR) usa JAAS para autenticar o cliente Username. JAAS usa o Password fornecido pelo cliente MQTT.

O administrador do IBM MQ decide se autenticar Username ou não autenticar nada, configurando o canal do MQTT ao qual um cliente se conecta. Clientes podem ser designados a diferentes canais, e cada canal pode ser configurado para autenticar seus clientes de maneiras diferentes. Usando JAAS, é possível configurar quais métodos devem autenticar o cliente e quais podem opcionalmente autenticar o cliente.

A opção de identidade para autenticação não afeta a escolha da identidade para autorização. É possível querer configurar uma identidade comum para autorização para comodidade administrativa, mas autenticar cada usuário para usar essa identidade. O procedimento a seguir descreve as etapas para autenticar usuários individuais para usarem uma identidade comum:

1. O administrador do IBM MQ configura a identidade do canal do MQTT com qualquer nome, como `MQTTClientUser`, usando o IBM MQ Explorer.
2. O administrador IBM MQ autoriza `MQTTClient` a publicar e se inscrever em qualquer tópico:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. O desenvolvedor de aplicativos do cliente MQTT cria um objeto `MqttConnectOptions` e configura `Username` e `Password` antes de se conectar ao servidor.
4. O desenvolvedor de segurança cria um `LoginModule` JAAS para autenticar `Username` com `Password` e inclui-lo no arquivo de configuração JAAS.
5. O administrador do IBM MQ configura o MQTT do canal para autenticar o `UserName` do cliente usando JAAS.

Windows Linux AIX Autenticação de cliente do MQTT usando o TLS

Conexões entre o cliente MQTT e o gerenciador de filas são sempre iniciadas pelo cliente MQTT. O cliente MQTT sempre o cliente SSL. A autenticação de cliente para o servidor e a autenticação de servidor para o cliente MQTT são opcionais.

Ao fornecer ao cliente um certificado digital assinado privado, você pode autenticar o cliente MQTT para IBM MQ. O Administrador do IBM MQ pode forçar os clientes MQTT a se autenticarem no gerenciador de filas usando TLS. Só é possível solicitar autenticação de cliente como parte da autenticação mútua.

Como uma alternativa para o uso de SSL, alguns tipos de Virtual Private Network (VPN), como IPsec, autenticam os terminais de uma conexão TCP/IP. A VPN criptografa cada pacote IP que flui pela rede. Após uma conexão VPN ser estabelecida, você estabeleceu uma rede confiável. É possível conectar clientes MQTT a canais de telemetria usando TCP/IP sobre a rede VPN.

A autenticação de cliente usando o TLS depende de o cliente ter um segredo. O segredo é a chave privada do cliente no caso de um certificado autoassinado ou uma chave fornecida por uma autoridade de certificação. A chave é usada para assinar o certificado digital do cliente. Qualquer pessoa em posse de uma chave pública correspondente pode verificar o certificado digital. Certificados podem ser confiáveis ou, se estiverem em cadeia, rastreados de volta por uma cadeia de certificados para um certificado de raiz confiável. A verificação de cliente envia todos os certificados na cadeia de certificados fornecida pelo cliente para o servidor. O servidor verifica a cadeia de certificados até localizar um certificado de sua confiança. O certificado confiável é o certificado público gerado a partir de um certificado autoassinado, ou um certificado raiz normalmente emitido por uma autoridade de certificação. Como etapa final opcional, o certificado confiável pode ser comparado com uma lista de revogação de certificado de "produção".

O certificado confiável pode ter sido emitido por uma autoridade de certificação e já estar incluído no armazenamento de certificados JRE. Ele pode ser um certificado autoassinado ou qualquer certificado que tenha sido incluído no keystore de canal de telemetria como um certificado confiável.

Nota: O canal de telemetria tem um keystore/armazenamento confiável combinados que retêm chaves privadas para um ou mais canais de telemetria e quaisquer certificados públicos necessários para

autenticar clientes. Como um canal SSL deve ter um keystore, e esse é o mesmo arquivo que do armazenamento confiável de canal, o armazenamento de certificados JRE nunca é usado como referência. A implicação é que se a autenticação de um cliente exigir um certificado raiz de CA, você deverá colocar o certificado raiz no keystore para o canal, mesmo se esse certificado raiz de CA já estiver no armazenamento de certificados JRE. O armazenamento de certificados JRE nunca é usado como referência.

Pense nas ameaças que a autenticação de cliente deverá contar e nas funções que o cliente e o servidor desempenham na contagem de ameaças. A autenticação de um certificado de cliente sozinha é insuficiente para evitar acesso não autorizado a um sistema. Se alguma outra pessoa tiver retido o dispositivo do cliente, o dispositivo do cliente não estará agindo necessariamente com a autoridade do portador do certificado. Nunca conte com apenas uma defesa contra ataques indesejados. Use pelo menos uma abordagem de autenticação de dois fatores e tenha um suplemento de certificado com conhecimento de informações particulares. Por exemplo, use JAAS e autentique o cliente usando uma senha emitida pelo servidor.

A principal ameaça para o certificado de cliente é cair nas mãos erradas. O certificado fica retido em um keystore protegido por senha no cliente. Como ele é colocado no keystore? Como o cliente MQTT consegue a senha para o keystore? Até que ponto a proteção da senha é segura? Os dispositivos de telemetria costumam ser fáceis de remover e, por isso, podem ser hackeados em particular. O hardware de dispositivo deve ser à prova de violação? A distribuição e a proteção de certificados do lado do cliente são reconhecidas como difíceis; elas são chamadas de problema de gerenciamento de chaves.

Uma segunda ameaça é o mau uso do dispositivo para acessar servidores de maneiras indesejadas. Por exemplo, se o aplicativo MQTT estiver corrompido, talvez seja possível usar um ponto fraco da configuração do servidor usando a identidade do cliente autenticado.

Para autenticar um cliente MQTT usando SSL, configure o canal de telemetria e o cliente.

Conceitos relacionados

[“Configuração do canal de telemetria para autenticação de cliente do MQTT usando TLS” na página 353](#) OIBM MQ o administrador configura canais de telemetria no servidor. Cada canal é configurado para aceitar uma conexão TCP/IP em um número de porta diferente. Canais de TLS são configurados com acesso a arquivos-chave protegido por passphrase. Se um canal de TLS for definido sem passphrase ou arquivo-chave, o canal não aceitará conexões de TLS.

[Configuração do cliente do MQTT para autenticação de cliente usando o TLS](#)

Configuração do canal de telemetria para autenticação de cliente do MQTT usando TLS

OIBM MQ o administrador configura canais de telemetria no servidor. Cada canal é configurado para aceitar uma conexão TCP/IP em um número de porta diferente. Canais de TLS são configurados com acesso a arquivos-chave protegido por passphrase. Se um canal de TLS for definido sem passphrase ou arquivo-chave, o canal não aceitará conexões de TLS.

Configure a propriedade com `.ibm.mq.MQTT.ClientAuth` de um canal de telemetria TLS para `REQUIRED` para forçar todos os clientes que se conectam àquele canal para fornecer prova de que verificaram certificados digitais. Os certificados de cliente são autenticados usando certificados de autoridades de certificação, levando a um certificado de raiz confiável. Se o certificado de cliente for autoassinado ou for assinado por um certificado de uma autoridade de certificação, os certificados assinados publicamente do cliente ou da autoridade de certificação deverão ser armazenados de forma segura no servidor.

Coloque o certificado de cliente assinado publicamente ou o certificado da autoridade de certificação no keystore do canal de telemetria. No servidor, os certificados assinados publicamente são armazenados no mesmo arquivo-chave que os certificados assinados particularmente, e não em um armazenamento confiável separado.

O servidor verifica a assinatura de todos os certificados de cliente enviados usando todos os certificados públicos e conjuntos de criptografia que possui. O servidor verifica a ou ia de chaves. O gerenciador de

filas pode ser configurado para testar o certificado com relação à lista de revogação de certificado. A propriedade da lista de nomes de revogação do gerenciador de filas é SSLCRLNL.

Se algum dos certificados enviado por um cliente for verificado por um certificado no keystore do servidor, o cliente será autenticado.

O administrador do IBM MQ pode configurar o mesmo canal de telemetria ao utilizar JAAS para verificar se o `UserName` ou `ClientIdentifier` do cliente com a Senha do cliente.

É possível usar o mesmo keystore para diversos canais de telemetria.

A verificação de pelo menos um certificado digital no keystore do cliente protegido por senha no dispositivo autentica o cliente para o servidor. O certificado digital é somente utilizado para autenticação pelo IBM MQ. Ele não é usado para verificar o endereço TCP/IP do cliente ou configurar a identidade do cliente para autorização ou contabilidade. A identidade do cliente adotada pelo servidor é o `Username` ou `ClientIdentifier` do cliente ou uma identidade criados pelo administrador do IBM MQ.

Também é possível usar conjuntos de cifras TLS para autenticação de cliente. Se você planeja usar conjuntos de cifras SHA-2, consulte [“Requisitos do sistema para usar conjuntos de cifras SHA-2 com canais do MQTT”](#) na página 358.

Conceitos relacionados

[“Configuração do canal de telemetria para autenticação de canal usando TLS”](#) na página 355

O IBM MQ o administrador configura canais de telemetria no servidor. Cada canal é configurado para aceitar uma conexão TCP/IP em um número de porta diferente. Canais de TLS são configurados com acesso a arquivos-chave protegido por passphrase. Se um canal de TLS for definido sem passphrase ou arquivo-chave, o canal não aceitará conexões de TLS.

[CipherSpecs e CipherSuites](#)

Referências relacionadas

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

Windows

Linux

AIX

Autenticação de canal de telemetria usando o

TLS

Conexões entre o cliente MQTT e o gerenciador de filas são sempre iniciadas pelo cliente MQTT. O cliente MQTT sempre o cliente SSL. A autenticação de cliente para o servidor e a autenticação de servidor para o cliente MQTT são opcionais.

O cliente sempre tenta autenticar o servidor, a menos que o cliente esteja configurado para usar um CipherSpec que suporte conexão anônima. Se a autenticação falhar, a conexão não será estabelecida.

Como uma alternativa para o uso de SSL, alguns tipos de Virtual Private Network (VPN), como IPsec, autenticam os terminais de uma conexão TCP/IP. A VPN criptografa cada pacote IP que flui pela rede. Após uma conexão VPN ser estabelecida, você estabeleceu uma rede confiável. É possível conectar clientes MQTT a canais de telemetria usando TCP/IP sobre a rede VPN.

A autenticação de servidor usando SSL autentica o servidor para o qual você está prestes a enviar informações confidenciais. O cliente realiza as verificações correspondentes aos certificados enviados do servidor, em relação aos certificados colocados em seu armazenamento confiável ou em seu JREcacerts loja.

O armazenamento de certificados JRE é um arquivo JKS,cacerts . Ele está localizado em JRE InstallPath\lib\security\. Ele é instalado com a senha padrão changeit. É possível armazenar certificados de sua confiança no armazenamento de certificados JRE ou no armazenamento confiável do cliente. Não é possível usar os dois armazenamentos. Use o armazenamento confiável do cliente se desejar manter os certificados públicos nos quais o cliente confia separados de outros certificadosJava uso de aplicativos. Use o armazenamento de certificados JRE se desejar usar um armazenamento de certificados comum para todosJava aplicativos em execução no cliente. Se você decidir usar o armazenamento de certificados JRE, revise os certificados que ele contém para se certificar de que confia neles.

É possível modificar a configuração JSSE fornecendo um provedor de confiança diferente. É possível customizar um provedor de confiança para executar diferentes verificações em um certificado. Em alguns ambientes OGSi que utilizaram o cliente MQTT, o ambiente do fornece um provedor de confiança diferente.



Para autenticar o canal de telemetria usando TLS, configure o servidor e o cliente.

Windows Linux AIX **Configuração do canal de telemetria para autenticação de canal usando TLS**

O IBM MQ o administrador configura canais de telemetria no servidor. Cada canal é configurado para aceitar uma conexão TCP/IP em um número de porta diferente. Canais de TLS são configurados com acesso a arquivos-chave protegido por passphrase. Se um canal de TLS for definido sem passphrase ou arquivo-chave, o canal não aceitará conexões de TLS.

Armazene o certificado digital do servidor, assinado com sua chave privada, no armazenamento de chaves que o canal de telemetria for utilizar no servidor. Armazene quaisquer certificados na ou ia de chaves no keystore, caso queira transmitir a ou ia de chaves para o cliente. Configure o canal de telemetria usando o explorador do IBM MQ para usar o TLS. Forneça a ele o caminho para o keystore e o passphrase para acessar o keystore. Se você não configurar o número da porta TCP/IP do canal, o número da porta do canal de telemetria do TLS será padronizado para 8883.

Também é possível usar os conjuntos de cifras do TLS para a autenticação de canal. Se você planeja usar conjuntos de cifras SHA-2, consulte [“Requisitos do sistema para usar conjuntos de cifras SHA-2 com canais do MQTT”](#) na página 358.

Importante:   A partir do IBM MQ 9.4.0, os repositórios de chaves e arquivos stash do CMS não são suportados com canais AMQP e MQTT que usam SSL/TLS. Use repositórios de chaves PKCS #12 e proteja as senhas do repositório de chaves usando o sistema de proteção de senha IBM MQ . É possível criar um repositório de chaves PKCS #12 usando o comando a seguir:

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

Esse comando cria um arquivo de repositório de chaves PKCS #12 denominado *filename.p12* que é protegido com a senha especificada

Conceitos relacionados

[“Configuração do canal de telemetria para autenticação de cliente do MQTT usando TLS”](#) na página 353

O IBM MQ o administrador configura canais de telemetria no servidor. Cada canal é configurado para aceitar uma conexão TCP/IP em um número de porta diferente. Canais de TLS são configurados com acesso a arquivos-chave protegido por passphrase. Se um canal de TLS for definido sem passphrase ou arquivo-chave, o canal não aceitará conexões de TLS.

[CipherSpecs e CipherSuites](#)

Referências relacionadas

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)



Exemplo de configuração do canal MQTT usando a autenticação TLS

Este exemplo orienta você em um exemplo de configuração de um canal do MQTT que usa a autenticação TLS.

O exemplo configura um canal entre MQTT e Mosquitto.

O exemplo usa um contêiner Docker para IBM MQ on Red Hat Enterprise Linux e Mosquitto no CentOS, mas se aplica a qualquer tipo de servidor. (CentOS foi usado para Mosquitto devido a autorizações de registro.)

Configure o keystore e o canal do IBM MQ para TLS unidirecional

Importante:   A partir do IBM MQ 9.4.0, os repositórios de chaves e arquivos stash do CMS não são suportados com canais AMQP e MQTT que usam SSL/TLS. Use repositórios de chaves PKCS #12 e proteja as senhas do repositório de chaves usando o sistema de proteção de senha IBM MQ .

Conclua as etapas a seguir:

1.   Crie o keystore IBM MQ :



```
runmqakm -keydb -create -db mqtt.p12 -pw "passw0rd" -type p12
```

2.   Crie um certificado pessoal:

```
runmqakm -cert -create -db mqtt.p12 -pw "passw0rd" -size 2048 -dn "CN= mqm, OU=MQTest, O=MQSuppor, C=US" -sig_alg SHA256_WITH_RSA -label ibmwebspheremqmqm
```

É possível usar o comando a seguir para confirmar a criação do certificado:

```
runmqakm -cert -list -v -db mqtt.p12 -pw "passw0rd"
```

3.   Crie o canal MQTT inserindo o comando a seguir no prompt runmqsc:

```
DEFINE CHANNEL(MQTTDEMO) CHLTYPE(MQTT) BACKLOG(4096) PORT(8883) MCAUSER('mqm')  
PROTOCOL(MQTTV311,MQTTV3,HTTP) SSLCAUTH(OPTIONAL) SSLCIPH('SSL_RSA_WITH_AES_256_CBC_SHA256')  
SSLKEYR('/var/mqm/mqtt/mqtt.p12') SSLKEYP('passw0rd') TRPTYPE(TCP)
```

Observe que o canal usa mapeamentos de cifra Java , consulte [TLS CipherSpecs e CipherSuites em classes IBM MQ para JMS](#).

4. Extraia o certificado:

```
runmqakm -cert -extract -db mqtt.kdb -stashed -label ibmwebspheremqmqm -target serverCert.pem
```

Instale o Mosquitto no CentOS em um contêiner Docker

Conclua as etapas a seguir para criar um contêiner Docker com Mosquitto em execução no CentOS:

1. `docker pull centos`
2. `docker run -it centos /bin/bash`
3. `yum -y install epel-release`
4. `yum -y install mosquitto`

Mova o certificado de assinante para Mosquitto

Conclua as etapas a seguir para mover o certificado criado no IBM MQ para o Mosquitto. Essas etapas são executadas na máquina host Docker .

1. Visualize os IDs do contêiner no Docker:

```
docker container ls
```

2. Copie o arquivo do contêiner do Docker para o Docker do sistema local..

```
cp MQ_Container_ID:/var/mqm/mqtt/serverCert.pem serverCert.pem
```

3. Copie o arquivo de sua máquina local para o diretório-raiz na máquina do centOS :

```
docker cp serverCert.pem CentOS_ContainerID:/serverCert.pem
```

Publicar com Mosquitto

Publique uma mensagem de teste no Mosquitto usando o seguinte comando:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d
```

Os argumentos de comando têm os significados a seguir:

-h

O endereço IP do host Red Hat Enterprise Linux (pode ser localizado usando **nslookup**).

-- cafile

O arquivo contendo o certificado de assinante.

-- inseguro

Essa opção é especificada porque o exemplo está usando um certificado autoassinado.. Não use essa opção ao usar certificados de CA reais.

-p

Número da porta.

-i

ID do cliente.

-t

O tópico que está sendo publicado.

-m

A mensagem sendo publicada.

-d

Ative as mensagens de depuração

Configure o canal do MQTT para autenticação TLS mútua

Insira o comando a seguir para reconfigurar o canal MQTT como SSLCAUTH (REQUIRED).

```
ALTER CHANNEL(MQTTDEMO) CHLTYPE(MQTT) SSLCAUTH(REQUIRED)
```

Crie um par chave / certificado no servidor Mosquitto e inclua em IBM MQ

Insira os comandos a seguir para criar o par chave / certificado no Mosquitto:

1. Use **openssl** para criar o par de chaves / certificados para Mosquitto:

```
openssl req -x509 -newkey rsa:4096 -keyout mosquittoKey.pem -out mosquittoCert.pem -subj "/CN=Mosquitto"
```

2. Listar os IDs do contêiner para os contêineres:

```
docker container ls
```

3. Copie o certificado Mosquitto para o docker do sistema local:

```
docker cp CentOS_ContainerID:mosquittoCert.pem .
```

4. Copie o certificado Mosquitto para IBM MQ:

```
docker cp mosquittoCert.pem MQ_Container_ID:/var/mqm/mqtt
```

5. Inclua o certificado no keystore IBM MQ :

```
runmqakm -cert -add -db mqtt.kdb -stashed -file mosquittoCert.pem
```

6. Reinicie o canal MQTT .

Publicar com Mosquitto e autenticação mútua

Conclua as etapas a seguir para publicar com Mosquitto usando autenticação mútua.

1. O comando a seguir deve publicar com êxito uma mensagem de teste:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d --cert mosquittoCert.pem --key mosquittoKey.pem
```

2. O comando a seguir deve falhar ao publicar uma mensagem de teste e gerar uma mensagem de erro porque ele não envia um certificado pessoal do Mosquitto:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d /var/mqm/qmgrs/mqttDemoQM/errors/ mqxr_0.log
```

Informações relacionadas

[Gerenciando chaves e certificados](#)

Windows Linux AIX Requisitos do sistema para usar conjuntos de cifras SHA-2 com canais do MQTT

Se você usar uma versão do Java que suporta os conjuntos de cifras SHA-2, será possível usar esses conjuntos para proteger seus canais e aplicativos clientes MQTT (telemetria).

Para o IBM MQ 8.0, que inclui o serviço de telemetria (MQXR), a versão mínima do Java é Java 7 partir do IBM, SR6. Os conjuntos de cifras SHA-2 são suportados por padrão no Java 7 partir do IBM, SR4 em diante. Portanto, é possível usar os conjuntos de cifras SHA-2 com o serviço de telemetria (MQXR) para proteger os canais MQTT (telemetria).

Se você estiver executando um cliente MQTT com um JRE diferente, será necessário assegurar-se de que ele também suporte os conjuntos de cifras SHA-2.

Conceitos relacionados

[Serviço de telemetria \(MQXR\)](#)

[“Configuração do canal de telemetria para autenticação de canal usando TLS” na página 355](#)

O IBM MQ o administrador configura canais de telemetria no servidor. Cada canal é configurado para aceitar uma conexão TCP/IP em um número de porta diferente. Canais de TLS são configurados com acesso a arquivos-chave protegido por passphrase. Se um canal de TLS for definido sem passphrase ou arquivo-chave, o canal não aceitará conexões de TLS.

Referências relacionadas

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

Windows Linux AIX Privacidade de publicação em canais de telemetria

A privacidade de publicações do MQTT enviadas em qualquer direção entre canais de telemetria é assegurada usando o TLS para criptografar transmissões por meio da conexão.

Os clientes do MQTT que se conectam aos canais de telemetria usam o TLS para assegurar a privacidade de publicações transmitidas no canal usando criptografia de chave simétrica. Como os terminais não são autenticados, não é possível confiar somente na criptografia do canal. Combine a proteção da privacidade com a autenticação mútua ou de servidor.

Como uma alternativa para o uso de SSL, alguns tipos de Virtual Private Network (VPN), como IPsec, autenticam os terminais de uma conexão TCP/IP. A VPN criptografa cada pacote IP que flui pela rede. Após uma conexão VPN ser estabelecida, você estabeleceu uma rede confiável. É possível conectar clientes MQTT a canais de telemetria usando TCP/IP sobre a rede VPN.

Para uma configuração típica, que criptografa o canal e autentica o servidor, consulte [“Autenticação de canal de telemetria usando o TLS” na página 354](#).

Criptografando conexões TLS sem autenticar o servidor expõe a conexão com ataques man-in-the-middle. Embora as informações que você troca estejam protegidas contra interceptação, você não sabe com quem está trocando-as. A menos que você controle a rede, você está exposto a alguém que intercepte suas transmissões de IP e se disfarce como o terminal.

É possível criar uma conexão TLS criptografada, sem autenticar o servidor, utilizando uma CipherSpec de troca de chave Diffie-Hellman anônimo que suporta TLS. O segredo principal, compartilhado entre o cliente e o servidor e utilizados para criptografar transmissões TLS, é estabelecida sem a troca de um certificado do servidor assinado em particular.

Como as conexões anônimas são inseguras, a maioria das implementações TLS não é padronizada para usar CipherSpecs anônimo. Se uma solicitação do cliente para conexão TLS for aceita por um canal de telemetria, o canal deverá ter um keystore protegido por uma passphrase. Por padrão, desde que as implementações TLS não usem CipherSpecs anônimo, o keystore deve conter um certificado assinado em particular que o cliente possa autenticar.

Se você usar CipherSpecs anônimo, o armazenamento de chaves do servidor deve existir, mas ele não precisa conter quaisquer certificados assinados particularmente.

Outra maneira de estabelecer uma conexão criptografada é substituir o provedor de confiança no cliente por sua própria implementação. O provedor de confiança não autenticaria o certificado do servidor, mas a conexão seria criptografada.



Atenção: Ao usar TLS com MQTT, é possível usar mensagens grandes, no entanto, pode haver um possível impacto no desempenho ao fazer isso. MQTT é otimizado para processar mensagens pequenas (geralmente entre 1KB e 1MB de tamanho).

Windows

Linux

AIX

Configuração do TLS de clientes MQTT Java e canais de telemetria

Configure o TLS para autenticar o canal de telemetria e o cliente MQTT Java e criptografar a transferência de mensagens entre eles. MQTT Os clientes do Java usam o Java Secure Socket Extension (JSSE) para conectar canais de telemetria usando TLS. Como uma alternativa para o uso de SSL, alguns tipos de Virtual Private Network (VPN), como IPsec, autenticam os terminais de uma conexão TCP/IP. A VPN criptografa cada pacote IP que flui pela rede. Após uma conexão VPN ser estabelecida, você estabeleceu uma rede confiável. É possível conectar clientes MQTT a canais de telemetria usando TCP/IP sobre a rede VPN.

É possível configurar a conexão entre um cliente Java MQTT e um canal de telemetria para usar o protocolo TLS sobre TCP/IP. O que é assegurado depende de como você configura o TLS para usar o JSSE. Começando com a configuração mais segura, é possível configurar três níveis diferentes de segurança:

1. Permita que somente clientes confiáveis do MQTT se conectem. Conecte um cliente do MQTT somente a um canal de telemetria confiável. Criptografe mensagens entre o cliente e o gerenciador de filas; consulte [“Autenticação de cliente do MQTT usando o TLS”](#) na página 352
2. Conecte um cliente do MQTT somente a um canal de telemetria confiável. Criptografe mensagens entre o cliente e o gerenciador de filas; consulte [“Autenticação de canal de telemetria usando o TLS”](#) na página 354.
3. Criptografe mensagens entre o cliente e o gerenciador de filas; consulte [“Privacidade de publicação em canais de telemetria”](#) na página 358.

Parâmetros de Configuração de JSSE

Modifique os parâmetros de JSSE para alterar a maneira como uma conexão TLS é configurada. Os parâmetros de configuração de JSSE são organizados em três conjuntos:

1. [Canal do MQ Telemetry](#)
2. [Cliente MQTT Java](#)
3. [JRE](#)

Configure os parâmetros de canal de telemetria usando IBM MQ Explorer. Configure o MQTT Java Client parâmetros no atributo `MqttConnectionOptions.SSLProperties`. Modifique os parâmetros de segurança do JRE editando arquivos no diretório de segurança do JRE no cliente e no servidor.

Canal do MQ Telemetry

Configure todos os parâmetros TLS do canal de telemetria usando o IBM MQ Explorer.

ChannelName

`ChannelName` é um parâmetro necessário em todos os canais.

O nome do canal identifica o canal associado a um determinado número de porta. Nomeie canais para ajudá-lo a administrar conjuntos de clientes do MQTT.

PortNumber

`PortNumber` é um parâmetro opcional em todos os canais. Ele é padronizado como 1883 para canais TCP e 8883 para canais TLS.

O número da porta TCP/IP associado a esse canal. Clientes do MQTT são conectados a um canal especificando a porta definida para o canal. Se o canal tiver propriedades TLS, o cliente deverá se conectar usando o protocolo TLS; por exemplo:

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId1");
mqttClient.connect();
```

KeyFileName

`KeyFileName` é um parâmetro necessário para canais TLS. Ele deve ser omitido para canais TCP.

`KeyFileName` é o caminho para o Java de chaves contendo os certificados digitais fornecidos por você. Use JKS, JCEKS ou PKCS12 como tipo de keystore no servidor.

Identifique o tipo de keystore usando uma das seguintes extensões de arquivo:

- .jks
- .jceks
- .p12
- .pkcs12

Um keystore com qualquer outra extensão de arquivo é assumido como um keystore JKS.

É possível combinar um tipo de keystore no servidor com outros tipos de keystore no cliente.

Coloque o certificado particular do servidor no keystore. O certificado é conhecido como o certificado do servidor. O certificado pode ser autoassinado ou fazer parte de uma sequência de certificados que é assinada por uma autoridade de assinatura.

Se você estiver usando uma sequência de certificados, coloque os certificados associados no keystore do servidor.

O certificado do servidor, e quaisquer certificados na ou de certificados, é enviado para os clientes autenticarem a identidade do servidor.

Se você tiver configurado `ClientAuth` como `Required`, o keystore deverá conter quaisquer certificados necessários para autenticar o cliente. O cliente envia um certificado autoassinado ou uma sequência de certificados e o cliente é autenticado pela primeira verificação desse material a um certificado no keystore. Usando uma sequência de certificados, um certificado pode verificar muitos clientes, mesmo se eles forem emitidos com diferentes certificados de cliente.

PassPhrase

`PassPhrase` é um parâmetro necessário para canais TLS. Ele deve ser omitido para canais TCP.

O passphrase é usado para proteger o keystore.

ClientAuth

ClientAuth é um parâmetro TLS opcional. Ele é padronizado para não autenticação de cliente. Ele deve ser omitido para canais TCP.

Configure ClientAuth se quiser que o serviço de telemetria (MQXR) autentique o cliente antes de permitir que o cliente se conecte ao canal de telemetria.

Se você configurar ClientAuth, o cliente deverá se conectar ao servidor usando TLS e autenticar o servidor. Em resposta à configuração de ClientAuth, o cliente envia seu certificado digital para o servidor e quaisquer outros certificados em seu keystore. Seu certificado digital é conhecido como certificado de cliente. Esses certificados são autenticados em relação a certificados mantidos no armazenamento de chaves do canal e no armazenamento JRE cacerts.

CipherSuite

CipherSuite é um parâmetro TLS opcional. Ele é padronizado para tentar todos os CipherSpecs ativados. Ele deve ser omitido para canais TCP.

Se desejar usar um determinado CipherSpec, configure CipherSuite para o nome do CipherSpec que deve ser usado para estabelecer a conexão TLS.

O serviço de telemetria e o cliente MQTT negociam um CipherSpec comum de todos os CipherSpecs ativados em cada extremidade. Se um CipherSpec específico for especificado em ambas as extremidades da conexão, ele deverá corresponder ao CipherSpec na outra extremidade.

Instale cifras adicionais incluindo provedores adicionais no JSSE.

Federal Information Processing Standards (FIPS)

FIPS é uma configuração opcional. Por padrão, ela não é configurada.

No painel de propriedades do gerenciador de filas ou usando o comando **runmqsc**, configure SSLFIPS. SSLFIPS especifica se apenas algoritmos certificados por FIPS devem ser usados.

Lista de nomes de revogação

A lista de nomes de revogação é uma configuração opcional. Por padrão, ela não é configurada.

No painel de propriedades do gerenciador de filas ou usando o comando **runmqsc**, configure SSLCRLNL. SSLCRLNL especifica uma lista de nomes de objetos de informações sobre autenticação que são usados para fornecer locais de revogação de certificado.

Nenhum outro parâmetro de gerenciador de filas que configura propriedades TLS é usado.

Cliente MQTT Java

Configure propriedades TLS para o cliente Java em `MqttConnectionOptions.SSLProperties`; por exemplo:

```
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.keyStoreType", "JKS");
com.ibm.micro.client.mqttv3.MqttConnectOptions conOptions = new MqttConnectOptions();
conOptions.setSSLProperties(sslClientProperties);
```

Os nomes e valores de propriedades específicas estão descritos na classe `MqttConnectOptions`. Para obter links para a documentação da API do cliente para as bibliotecas do cliente MQTT, consulte [Referência de programação do cliente MQTT](#).

Protocolo

Protocolo é opcional.

O protocolo é selecionado na negociação com o servidor de telemetria. Se você requerer um protocolo específico, é possível selecionar um. Se o servidor de telemetria não suportar o protocolo, a conexão falhará.

ContextProvider

ContextProvider é opcional.

KeyStore

KeyStore é opcional. Configure-o se ClientAuth estiver configurado no servidor para forçar a autenticação do cliente.

Coloque o certificado digital do cliente, assinado com o uso de sua chave privada, no keystore. Especifique a senha e o caminho do keystore. O tipo e o provedor são opcionais. JKS é o tipo padrão e IBMJCE é o provedor padrão.

Especifique um provedor de keystore diferente para fazer referência a uma classe que inclui um novo provedor de keystore. Passe o nome do algoritmo usado pelo provedor de keystore para instanciar KeyManagerFactory configurando o nome do gerenciador de chave.

TrustStore

TrustStore é opcional. É possível colocar todos os certificados em que confia no armazenamento JRE cacerts.

Configure o armazenamento confiável se quiser ter um armazenamento confiável diferente para o cliente. Pode não ser necessário configurar o armazenamento confiável se o servidor estiver usando um certificado emitido por uma CA conhecida que já possui seu certificado raiz armazenado em cacerts.

Inclua o certificado assinado publicamente do servidor ou o certificado raiz do armazenamento confiável e especifique o caminho e a senha do armazenamento confiável. JKS é o tipo padrão e IBMJCE é o provedor padrão.

Especifique um provedor de armazenamento confiável diferente para fazer referência a uma classe que inclui um novo provedor de armazenamento confiável. Passe o nome do algoritmo usado pelo provedor de armazenamento confiável para instanciar TrustManagerFactory configurando o nome do gerenciador de confiança.



JRE



Outros aspectos de segurança Java que afetam o comportamento do TLS no cliente e no servidor são configurados no JRE. Os arquivos de configuração em Windows estão em *Java Installation Directory\jre\lib\security*. Se você estiver usando o JRE fornecido com o IBM MQ, o caminho será igual ao mostrado na seguinte tabela:

Plataforma	Caminho de arquivo..
Windows	<i>WMQ Installation Directory\java\jre\lib\security</i>
Plataformas AIX and Linux	<i>WMQ Installation Directory/java/jre64/jre/lib/security</i>

Autoridades de certificação conhecidas

O arquivo cacerts contém os certificados raiz de autoridades de certificação bem conhecidas. O cacerts é usado por padrão, a menos que você especifique um armazenamento de confiança. Se você usar o armazenamento cacerts ou não fornecer um armazenamento confiável, você deverá revisar e editar a lista de assinantes em cacerts para atender aos seus requisitos de segurança.

  É possível usar o comando **runmqktool** para gerenciar o arquivo de certificados cacerts .. cacerts é um arquivo JKS.. Especifique o parâmetro **-storetype jks** quando o comando **runmqktool** for usado para gerenciar o arquivo de certificados

  A senha padrão do arquivo cacerts é changeit. Mude a senha usando o comando **runmqktool -storepasswd** para proteger o arquivo

Configurando classes de segurança

Use o arquivo `java.security` para registrar provedores de segurança adicionais e outras propriedades de segurança padrão.

Permissões

Use o arquivo `java.policy` para modificar as permissões concedidas a recursos.
`javaws.policy` concede permissões para `javaws.jar`

Grau de Intensidade da Criptografia

Alguns JREs são fornecidos com segurança de criptografia reduzida. Se você não puder importar as chaves para os keystores, a criptografia de força reduzida poderá ser a causa. Se necessário, faça download de arquivos de jurisdição fortes, mas limitados, a partir de [IBM kits do desenvolvedor, Informações de segurança](#)

Importante: Seu país de origem pode ter restrições quanto à importação, à posse, ao uso ou à reexportação para outro país de um software de criptografia. Antes de fazer download ou usar os arquivos de políticas irrestritas, você deve verificar as leis do seu país. Verifique seus regulamentos e suas políticas com relação à importação, à posse, ao uso e à reexportação do software de criptografia para determinar se essas ações são permitidas.

Modificando o Provedor de Confiança para Permitir que o Cliente se Conecte a qualquer Servidor

O exemplo ilustra como incluir um provedor de confiança e fazer referência a ele a partir do código do cliente MQTT. O exemplo não faz autenticação do cliente ou servidor. A conexão TLS resultante é criptografada sem ser autenticada.

O fragmento de código em Figura 16 na página 363 configura o provedor de confiança `AcceptAllProviders` e o gerenciador de confiança para o cliente MQTT.

```
java.security.Security.addProvider(new AcceptAllProvider());
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.trustManager", "TrustAllCertificates");
sslClientProperties.setProperty("com.ibm.ssl.trustStoreProvider", "AcceptAllProvider");
conOptions.setSSLProperties(sslClientProperties);
```

Figura 16. fragmento de código do clienteMQTT

```
package com.ibm.mq.id;
public class AcceptAllProvider extends java.security.Provider {
private static final long serialVersionUID = 1L;
public AcceptAllProvider() {
super("AcceptAllProvider", 1.0, "Trust all X509 certificates");
put("TrustManagerFactory.TrustAllCertificates",
AcceptAllTrustManagerFactory.class.getName());
}
}
```

Figura 17. `AcceptAllProvider.java`

```
protected static class AcceptAllTrustManagerFactory extends
javax.net.ssl.TrustManagerFactorySpi {
public AcceptAllTrustManagerFactory() {}
protected void engineInit(java.security.KeyStore keystore) {}
protected void engineInit(
javax.net.ssl.ManagerFactoryParameters parameters) {}
protected javax.net.ssl.TrustManager[] engineGetTrustManagers() {
return new javax.net.ssl.TrustManager[] { new AcceptAllX509TrustManager() };
}
}
```

Figura 18. `AcceptAllTrustManagerFactory.java`

```

protected static class AcceptAllX509TrustManager implements
javax.net.ssl.X509TrustManager {
public void checkClientTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Client authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public void checkServerTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Server authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public java.security.cert.X509Certificate[] getAcceptedIssuers() {
return new java.security.cert.X509Certificate[0];
}
private static void report(String string) {
System.out.println(string);
}
}
}

```

Figura 19. *AcceptAllX509TrustManager.java*

Windows

Linux

AIX

Configuração JAAS do Canal de Telemetria

Configure JAAS para autenticar o Username enviado pelo cliente.

O administrador do IBM MQ configura quais canais do MQTT requerem autenticação de cliente usando JAAS. Especifique o nome de uma configuração JAAS para cada canal que deve executar a autenticação JAAS. Todos os canais podem usar a mesma configuração JAAS ou podem usar diferentes configurações JAAS. As configurações são definidas em *WMQData directory\qmgrs\qMgrName\mqxr\jaas.config*.

O arquivo *jaas.config* é organizado pelo nome de configuração JAAS. Sob cada nome de configuração há uma lista de configurações de login; consulte [“Amostra de arquivo jaas.config”](#) na página 365.

O JAAS fornece quatro módulos de login padrão. A NT padrão e módulos de login do UNIX têm valor limitado.

JndiLoginModule

Autentica com relação a um serviço de diretório configurado em JNDI (Java Naming and Directory Interface).

Krb5LoginModule

Autentica usando protocolos Kerberos.

NTLoginModule

Autentica usando informações de segurança de NT para o usuário atual.

UnixLoginModule

Autentica usando as informações de segurança do UNIX para o usuário atual.

O problema com o uso de *NTLoginModule* ou *UnixLoginModule* é que o serviço de telemetria (MQXR) é executado com a identidade *mqm* e não com a identidade do canal do MQTT. *mqm* é a identidade passada para *NTLoginModule* ou *UnixLoginModule* para autenticação e não a identidade do cliente.

Para superar esse problema, grave seu próprio módulo de login ou use os outros módulos de login padrão. Um *JAASLoginModule.java* de amostra é fornecido com o MQ Telemetry. Trata-se de uma implementação da interface *javax.security.auth.spi.LoginModule*. Use-o para desenvolver seu próprio método de autenticação.

Quaisquer novas classes *LoginModule* que você fornecer deverão estar no caminho da classe do serviço de telemetria (MQXR). Não coloque suas classes nos diretórios IBM MQ que estão no caminho da classe. Crie seus próprios diretórios e defina o caminho da classe inteiro para o serviço de telemetria (MQXR).

É possível aumentar o caminho de classe utilizado pelo serviço de telemetria (MQXR) configurando caminho de classe no arquivo `service.env`. `CLASSPATH` deve ser capitalizado e a instrução do caminho da classe só pode conter literais. Não é possível usar variáveis no `CLASSPATH`; por exemplo `CLASSPATH=%CLASSPATH%` está incorreto. O serviço de telemetria (MQXR) configura seu próprio caminho de classe. O `CLASSPATH` definido em `service.env` é incluído nele.

O serviço de telemetria (MQXR) fornece dois retornos de chamada que retornam o Nome do usuário e a Senha para um cliente conectado ao canal MQTT. O Nome do Usuário e a Senha são configurados no objeto `MqttConnectOptions`. Consulte [“Método JAASLoginModule.Login\(\) de Amostra” na página 365](#) para obter um exemplo de como acessar Username e Password.

Amostra de arquivo `jaas.config`

Um exemplo de um arquivo de configuração JAAS com uma configuração nomeada, `MQXRConfig`

```
MQXRConfig {
samples.JAASLoginModule required debug=true;
//com.ibm.security.auth.module.NTLoginModule required;
//com.ibm.security.auth.module.Krb5LoginModule required
//      principal=principal@your_realm
//      useDefaultCcache=TRUE
//      renewTGT=true;
//com.sun.security.auth.module.NTLoginModule required;
//com.sun.security.auth.module.UnixLoginModule required;
//com.sun.security.auth.module.Krb5LoginModule required
//      useTicketCache="true"
//      ticketCache="$${user.home}/${}tickets";
};
```

Método `JAASLoginModule.Login()` de Amostra

Um exemplo de um módulo de login JAAS codificado para receber o Nome do usuário e a Senha fornecidos por um cliente do MQTT.

```
public boolean login()
throws javax.security.auth.login.LoginException {
    javax.security.auth.callback.Callback[] callbacks =
    new javax.security.auth.callback.Callback[2];
    callbacks[0] = new javax.security.auth.callback.NameCallback("NameCallback");
    callbacks[1] = new javax.security.auth.callback.PasswordCallback(
    "PasswordCallback", false);
    try {
        callbackHandler.handle(callbacks);
        String username = ((javax.security.auth.callback.NameCallback) callbacks[0])
        .getName();
        char[] password = ((javax.security.auth.callback.PasswordCallback) callbacks[1])
        .getPassword();
        // Accept everything.
        if (true) {
            loggedIn = true;
        } else
        throw new javax.security.auth.login.FailedLoginException("Login failed");

        principal= new JAASPrincipal(username);

    } catch (java.io.IOException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    } catch (javax.security.auth.callback.UnsupportedCallbackException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    }
}

return loggedIn;
}
```

Tarefas relacionadas

Resolução do problema: [o módulo de login JAAS não é chamado pelo serviço de telemetria](#)

Referências relacionadas

[Classe AuthCallback MQXR](#)

Administrando um cliente de AMQP

É possível administrar um cliente de AMQP usando IBM MQ Explorer ou em uma linha de comando. Use o Explorer para configurar canais e monitorar clientes de AMQP que estão conectados a IBM MQ. Configure a segurança dos clientes de AMQP usando TLS e JAAS.

Antes de Começar

Para obter informações sobre como instalar AMQP em sua plataforma, consulte [Escolhendo o que instalar](#).

Administrando Usando o IBM MQ Explorer

Use o Explorer para configurar canais de AMQP e monitorar os clientes de AMQP que estão conectados a IBM MQ. É possível configurar a segurança dos clientes de AMQP usando TLS e JAAS.

Administrando Usando a Linha de Comandos

É possível administrar um cliente de AMQP na linha de comando [usando comandos MQSC](#).

Windows V 9.4.0 Linux V 9.4.0 AIX O serviço AMQP não inicia automaticamente na inicialização do gerenciador de filas

Em IBM MQ 9.4.0, o comportamento padrão da configuração do atributo **CONTROL** para iniciar o serviço AMQP mudou. Ao criar e iniciar um novo gerenciador de filas, o serviço AMQP não é iniciado automaticamente como parte do processo de inicialização do gerenciador de filas

Entre IBM MQ 9.0.4 e IBM MQ 9.4.0, o comportamento padrão da configuração do atributo **CONTROL** para iniciar o serviço AMQP é QMGR.

Se o componente AMQP foi instalado, o serviço AMQP foi iniciado automaticamente, mesmo se não utilizado. Para evitar a inicialização padrão da máquina virtual (JVM) AMQP Java, você tinha duas opções:

- Não instalando o componente AMQP ou
- Alterando o atributo **CONTROL** do serviço AMQP para MANUAL após o gerenciador de filas ser iniciado

Em IBM MQ 9.4.0, os gerenciadores de filas recém-criados reverteram a configuração do atributo **CONTROL** do SYSTEM.AMQP.SERVICE para MANUAL, que era a configuração padrão anterior a IBM MQ 9.0.4

Gerenciadores de filas migrados, se utilizando AMQP, continuam a autoinicialização do serviço durante a inicialização do gerenciador de filas. Para determinar se AMQP foi utilizado, o seguinte é verificado:

- Canais AMQP existentes
- Mensagens iniciadas pelo canal no log de erros do AMQP.



Atenção:

- Isso acontece apenas uma vez; a primeira vez que o gerenciador de filas é iniciado depois de um upgrade
- Durante a migração, se o atributo **CONTROL** for alterado de QMGR para MANUAL, uma mensagem de informação será registrada no log de erro IBM MQ para indicar a mudança. Consulte [Local dos logs do AMQP, logs de erro e arquivos de configuração](#) para obter mais informações

Se você desejar que o serviço AMQP seja iniciado automaticamente, mude o atributo **CONTROL** do serviço para QMGR e reinicie o gerenciador de filas. As reinicializações subsequentes do gerenciador de filas iniciam o serviço AMQP.

Visualizando objetos IBM MQ em uso por clientes de AMQP

É possível visualizar os diferentes recursos IBM MQ em uso por clientes de AMQP, por exemplo conexões e assinaturas.

Conexões

Quando o serviço é iniciado Hconns AMQP novo será criado e conectado ao gerenciador de filas. Este conjunto de Hconns é usado quando clientes de AMQP publicam mensagens. É possível visualizar os Hconns usando o comando **DISPLAY CONN**. Por exemplo:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'IBM MQ Advanced Message Queuing Protocol')
```

Esse comando também mostra quaisquer Hconns específicos do cliente. O Hconns que possuem um atributo de ID do cliente em branco são os Hconns usados no conjunto

Quando um cliente de AMQP se conecta a um canal de AMQP, um novo Hconn é conectado ao gerenciador de filas. Este Hconn é usado para consumir mensagens de forma assíncrona para as assinaturas que o cliente de AMQP criou. É possível visualizar o Hconn usado por um cliente AMQP particular usando o comando **DISPLAY CONN**. Por exemplo:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

Assinaturas criadas por clientes

Quando um cliente de AMQP se inscreve em um tópico, uma nova assinatura IBM MQ é criada. O nome da assinatura inclui as seguintes informações:

- O nome do cliente. Se o cliente se uma assinatura compartilhada, o nome do compartilhamento é utilizado
- O tópico padrão que o cliente inscrito para
- Um prefixo. O prefixo é `private` se o cliente criou uma assinatura não compartilhada ou `share` se o cliente aderiu a uma assinatura compartilhada

Para visualizar as assinaturas em uso por um cliente AMQP específico, execute o comando **DISPLAY SUB** e filtre no prefixo `private`:

```
DISPLAY SUB('/:private:*')
```

Para visualizar as assinaturas compartilhadas que estão em uso por vários clientes, execute o comando **DISPLAY SUB** e filtre o `share` prefixo:

```
DISPLAY SUB('/:share:*')
```

Como as assinaturas compartilhadas podem ser usadas por vários clientes de AMQP, talvez você queira visualizar os clientes que estão consumindo mensagens da assinatura compartilhada. É possível fazer isso, listando os Hconns que atualmente possuem um identificador aberto na fila de assinaturas. Para visualizar os clientes atualmente utilizando um compartilhamento, conclua as seguintes etapas:

1. Localize o nome da fila que a assinatura compartilhada utiliza como um destino. Por exemplo:

```
DISPLAY SUB('/:private:recv_e298452:public') DEST
5 : DISPLAY SUB('/:private:recv_e298452:public') DEST
AMQ8096: WebSphere MQ subscription inquired.
SUBID(414D5120514D31202020202020202020707E0A565C2D0020)
SUB('/:private:recv_e298452:public)
DEST(SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
```

2. Execute o comando **DISPLAY CONN** para localizar os identificadores abertos nessa fila:

```
DISPLAY CONN(*) TYPE(HANDLE) WHERE (OBJNAME
EQ SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
21 : DISPLAY CONN(*) TYPE(HANDLE) WHERE(OBJNAME EQ
SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
```



```
AMQ8276: Display Connection details.  
CONN(707E0A56642B0020)  
EXTCONN(414D5143514D312020202020202020)  
TYPE(HANDLE)  
  
OBJNAME(SYSTEM.BASE.TOPIC)      OBJTYPE(TOPIC)  
  
OBJNAME(SYSTEM.MANAGED.DURABLE.560A7E7020002961)  
OBJTYPE(Queue)
```

3. Para cada uma das manipulações, visualize o ID do cliente de AMQP com a manipulação aberta:

```
DISPLAY CONN(707E0A56642B0020) CLIENTID  
23 : DISPLAY CONN(707E0A56642B0020) CLIENTID  
  
AMQ8276: Display Connection details.  
CONN(707E0A56642B0020)  
EXTCONN(414D5143514D312020202020202020)  
TYPE(CONN)  
CLIENTID(recv_8f02c9d)  
DISPLAY CONN(707E0A565F290020) CLIENTID  
24 : DISPLAY CONN(707E0A565F290020) CLIENTID  
AMQ8276: Display Connection details.  
CONN(707E0A565F290020)  
EXTCONN(414D5143514D312020202020202020)  
TYPE(CONN)  
CLIENTID(recv_86d8888)
```

Identificação, autorização e autenticação do cliente de AMQP

Como outros aplicativos do cliente IBM MQ, é possível proteger conexões AMQP de várias maneiras.

É possível utilizar os seguintes recursos de segurança para proteger conexões AMQP para IBM MQ:

- [Registros de Autenticação de Canal](#)
- [Autenticação de conexão](#)
- Canal do MCA do usuário de configuração
- Definições de autoridade do IBM MQ
- [conectividade TLS](#)

De uma perspectiva de segurança, o estabelecimento de uma conexão consiste nas duas etapas a seguir:

- Decidir se a conexão deve continuar
- Decidir quais o IBM MQ a identidade do aplicativo assume posteriormente para verificações de autoridade

As informações a seguir destacam diferentes configurações do IBM MQ e as etapas que são trabalhadas quando um cliente AMQP tenta fazer uma conexão. Nem todas as configurações do IBM MQ utilizar todas as etapas descritas. Por exemplo, algumas configurações não usam TLS para conexões dentro do firewall da empresa e algumas configurações usam o TLS, mas não usam certificados de cliente para autenticação. Muitos ambientes não utilizam módulos customizados ou módulos JAAS customizados.

Estabelecendo uma conexão

As etapas a seguir descrevem o que acontece quando uma conexão está sendo estabelecida por um cliente AMQP. As etapas determinam se a conexão continuará e qual identidade do IBM MQ o aplicativo assume para verificações de autoridade:

1. Se o cliente abre uma conexão TLS para IBM MQ e fornece um certificado, o gerenciador de filas tenta validar o certificado de cliente.
2. Se o cliente fornece credenciais de nome de usuário e senha, um quadro AMQP SASL é recebido pelo gerenciador de filas e a configuração de MQ CONNAUTH é verificada.
3. regras de autenticação de canal do MQ são verificados (por exemplo, se o endereço IP e TLS certificado DN são válidos)

4. O canal MCAUSER é declarado, a menos que as regras de autenticação de canal determinem o contrário.
5. Se um módulo JAAS tiver sido configurado, ele será chamado
6. verifique a autoridade CONNECT aplicado resultante MQ ID do usuário do MQ.
7. Conexão estabelecida com uma identidade do IBM MQ assumida.

Publicando uma mensagem

As etapas a seguir descrevem o que acontece quando uma mensagem está sendo publicada por um cliente AMQP. As etapas determinam se a conexão continuará e qual identidade do IBM MQ o aplicativo assume para verificações de autoridade:

1. link AMQP anexar quadro chega no gerenciador de filas. IBM MQ publicam autoridade para a sequência de tópico especificado está marcada para a identidade do usuário do MQ estabelecida durante a conexão.
2. A mensagem é publicada para sequência de tópico especificado.

Assinar um tópico padrão

As etapas a seguir descrevem o que acontece quando um cliente AMQP assina um tópico padrão. As etapas determinam se a conexão continuará e qual identidade do IBM MQ o aplicativo assume para verificações de autoridade:

1. link AMQP anexar quadro chega no gerenciador de filas. IBM MQ autoridade de assinatura para o padrão de tópico especificado está marcada para a identidade do usuário do MQ estabelecida durante a conexão.
2. A assinatura é criada.

Identidade e autorização do cliente de AMQP

Use o ID do cliente de AMQP, o nome de usuário de AMQP ou uma identidade de cliente comum definida no canal ou em uma regra de autenticação de canal, para autorização de acesso a objetos IBM MQ.

O administrador faz a escolha ao definir ou modificar o canal AMQP, configurando a configuração CONNAUTH do gerenciador de filas ou definindo regras de autenticação de canal. A identidade é usada para autorizar o acesso aos tópicos do IBM MQ. A opção é feita com base no seguinte:

1. O canal USECLNTID atributo.
2. O atributo ADOPTCTX da regra CONNAUTH do gerenciador de filas.
3. O atributo MCAUSER definido no canal.
4. O atributo USERSRC de uma regra de autenticação de canal correspondente.

Evite problemas: A identidade, escolhido por este processo é referido como, por exemplo, o comando DISPLAY CHSTATUS (AMQP), como o MCAUSER do cliente. Esteja ciente de que este não é necessariamente a mesma identidade do MCAUSER do canal que é referido na opção (2).

Use o IBM MQ **setmqaut** comando para selecionar quais objetos e quais ações, estão autorizados para serem usados pela identidade associada ao canal AMQP. Por exemplo, os comandos a seguir autorizam uma identidade de canal AMQPClient, fornecida pelo administrador do gerenciador de filas QM1:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPClient -all +pub +sub
```

e

```
setmqaut -m QM1 -t qmgr -p AMQPClient -all +connect
```

Autenticação do cliente de AMQP usando uma senha

Autentique o nome de usuário do cliente de AMQP usando a senha do cliente. É possível autenticar o cliente usando uma identidade diferente da identidade usada para autorizar o cliente para publicar e se inscrever para tópicos.

O serviço AMQP pode utilizar o MQ CONNAUTH ou JAAS para autenticar o nome de usuário do cliente. Se um desses for configurado, a senha fornecida pelo cliente é verificada pela configuração do MQ CONNAUTH ou o módulo JAAS.

O procedimento a seguir descreve as etapas de exemplo para autenticar usuários individuais em relação aos usuários e senhas do SO local e, se for bem-sucedido, adotar a identidade comum AMQPUser:

1. O administrador IBM MQ define a identidade MCAUSER do canal de AMQP para qualquer nome, como AMQPUser, usando o IBM MQ Explorer.
2. O administrador IBM MQ autoriza AMQPUser a publicar e se inscrever em qualquer tópico:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPUser -all +pub +sub +connect
```

3. O administrador do IBM MQ configura uma regra IDPWOS CONNAUTH para verificar o nome do usuário e a senha apresentados pelo cliente. A regra CONNAUTH deve configurar CHCKCLNT(REQUIRED) e ADOPTCTX(NO).

Nota: É recomendável usar as regras de autenticação de canal e para configurar o atributo do canal MCAUSER com um usuário que não possui privilégios, para permitir mais controle sobre conexões com o gerenciador de filas.

privacidade em canais de publicação

A privacidade de publicações enviadas em qualquer direção AMQP entre canais AMQP é protegida utilizando TLS para criptografar transmissões através da conexão.

Clientes AMQP que se conectam a canais AMQP utilizam o TLS para proteger a privacidade das publicações transmitida no canal utilizando criptografia de chave simétrica. Como os terminais não são autenticados, não é possível confiar somente na criptografia do canal. Combine a proteção da privacidade com a autenticação mútua ou de servidor.

Como uma alternativa para o uso de TLS, alguns tipos de Virtual Private Network (VPN), como IPsec, autenticam os terminais de uma conexão TCP/IP. A VPN criptografa cada pacote IP que flui pela rede. Após uma conexão VPN ser estabelecida, você estabeleceu uma rede confiável. É possível conectar clientes AMQP para canais AMQP usando TCP/IP sobre a rede VPN.

Criptografando conexões TLS sem autenticar o servidor expõe a conexão com ataques man-in-the-middle. Embora as informações que você troca estejam protegidas contra interceptação, você não sabe com quem está trocando-as. A menos que você controle a rede, você está exposto a alguém que intercepte suas transmissões de IP e se disfarce como o terminal.

É possível criar uma conexão TLS criptografada, sem autenticar o servidor, utilizando uma CipherSpec de troca de chave Diffie-Hellman anônimo que suporta TLS. O segredo principal, compartilhado entre o cliente e o servidor e utilizados para criptografar transmissões TLS, é estabelecida sem a troca de um certificado do servidor assinado em particular.

Como as conexões anônimas são inseguras, a maioria das implementações TLS não é padronizada para usar CipherSpecs anônimo. Se uma solicitação do cliente para conexão TLS será aceito por um canal AMQP, o canal deve ter um keystore protegido por uma passphrase. Por padrão, desde que as implementações TLS não usem CipherSpecs anônimo, o keystore deve conter um certificado assinado em particular que o cliente possa autenticar.

Se você usar CipherSpecs anônimo, o armazenamento de chaves do servidor deve existir, mas ele não precisa conter quaisquer certificados assinados particularmente.

Outra maneira de estabelecer uma conexão criptografada é substituir o provedor de confiança no cliente por sua própria implementação. O provedor de confiança não autenticaria o certificado do servidor, mas a conexão seria criptografada.

Configurando clientes de AMQP com TLS

É possível configurar clientes de AMQP para usar o TLS para proteger os dados que fluem na rede e autenticar a identidade do gerenciador de filas ao qual o cliente se conecta.

Para usar TLS para a conexão de um cliente de AMQP em um canal do AMQP, é necessário garantir que o gerenciador de filas tenha sido configurado para TLS. [Configurando o TLS em gerenciadores de filas](#) descreve como configurar o keystore a partir do qual um gerenciador de filas lê certificados TLS.

Quando o gerenciador de filas foi configurado com um armazenamento de chaves, deve-se configurar os atributos de TLS no canal AMQP ao qual os clientes irão se conectar. canais AMQP tem quatro atributos relacionados à configuração de TLS conforme a seguir:

SSLCAUTH

O atributo SSLCAUTH é usado para especificar se o gerenciador de filas deve exigir que um cliente de AMQP apresente um certificado de cliente para verificar sua identidade.

SSLCIPH

O atributo SSLCIPH especifica a cifra que o canal deve usar para codificar dados no fluxo TLS.

V 9.4.0 No IBM MQ 9.4.0, os canais AMQP suportam ANY* generic CipherSpecs. Para obter mais informações sobre CipherSpecs, consulte [Ativando CipherSpecs](#)

SSLPEER

O atributo SSLPEER é utilizado para especificar o nome distinto (DN) um certificado cliente deve corresponder se uma conexão deve ser permitida.

CERTLABL

O CERTLABL especifica o certificado do gerenciador de filas deve apresentar ao cliente. O keystore do gerenciador de filas pode conter vários certificados. Esse atributo permite que você especifique o certificado a ser utilizado para conexões com esse canal. Se nenhum CERTLABL for especificado, o certificado no repositório de chaves do gerenciador de filas com o rótulo correspondente ao atributo CERTLABL do gerenciador de filas será usado.

Quando você tiver configurado seu canal AMQP com os atributos TLS, deve-se reiniciar o serviço AMQP utilizando o seguinte comando:

```
STOP SERVICE(SYSTEM.AMQP.SERVICE) START SERVICE(SYSTEM.AMQP.SERVICE)
```

Quando um cliente de AMQP se conecta a um canal AMQP protegido por TLS, o cliente verifica a identidade do certificado apresentado pelo gerenciador de filas. Para fazer isso, você deve configurar seu cliente AMQP com um armazenamento confiável contendo o certificado do gerenciador de filas. As etapas para fazer isso variam de acordo com o cliente de AMQP que você está usando. Para obter informações sobre os vários clientes e APIs de AMQP, consulte a respectiva documentação do cliente de AMQP.

Referências relacionadas

[DEFINE CHANNEL](#) (definir um novo canal)

[STOP SERVICE](#) (parar um serviço) em [Multiplataformas](#)

[START SERVICE](#) (iniciar um serviço) em [Multiplataformas](#)

Desconectando clientes de AMQP do gerenciador de filas

Se você deseja desconectar clientes de AMQP do gerenciador de filas, execute o comando PURGE CHANNEL ou pare a conexão com o cliente de AMQP.

- Execute o comando **PURGE CHANNEL**. Por exemplo:

```
PURGE CHANNEL(MYAMQP) CLIENTID('recv_28dbb7e')
```

- Como alternativa, pare a conexão que o cliente de AMQP está usando para desconectá-lo, concluindo as seguintes etapas:

1. Localize a conexão que o cliente está usando, executando o comando **DISPLAY CONN** Por exemplo:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_28dbb7e')
```

A saída de comando é o seguinte:

```
DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
40 : DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
AMQ8276: Display Connection details.
CONN(707E0A565F2D0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE (CONN)
CLIENTID(recv_28dbb7e)
```

2. Pare a conexão. Por exemplo:

```
STOP CONN(707E0A565F2D0020)
```

Administrando o multicast

Utilize estas informações para saber mais sobre as tarefas de administração do IBM MQ Multicast como reduzir o tamanho de mensagens de multicast e ativar a conversão de dados.

Introdução ao Multicast

Use estas informações para uma introdução aos objetos de informações de comunicação e tópicos do IBM MQ Multicast.

Sobre esta tarefa

O sistema de mensagens do IBM MQ Multicast usa a rede para entregar mensagens mapeando tópicos para agrupar endereços. As tarefas a seguir são uma forma rápida para testar se o endereço IP e a porta necessários estão corretamente configurados para o sistema de mensagens de multicast.

Criando um objeto **COMMINFO** para multicast

O objeto de informações de comunicação (COMMINFO) contém os atributos associados à transmissão de multicast. Para obter mais informações sobre os parâmetros de objeto COMMINFO, consulte [DEFINE COMMINFO](#).

Use o exemplo da linha de comandos a seguir para definir um objeto COMMINFO para multicast:

```
DEFINE COMMINFO(MC1) GRPADDR(group address) PORT(port number)
```

em que *MC1* é o nome do seu objeto COMMINFO, *group address* é o seu endereço IP ou nome de DNS de multicast de grupo e *port number* é a porta na qual transmitir (o valor padrão é 1414).

Um novo objeto COMMINFO chamado *MC1* é criado; este nome é o nome que você deve especificar ao definir um objeto TOPIC no próximo exemplo.

Criando um objeto **TOPIC** para multicast

Um tópico é o assunto das informações que são publicadas em uma mensagem de publicação/ assinatura e um tópico é definido criando um objeto TOPIC. Os objetos TOPIC possuem dois parâmetros que definem se eles podem ser usados com multicast ou não. Estes parâmetros são: **COMMINFO** e **MCAST**.

- **COMMINFO** Este parâmetro especifica o nome do objeto de informações de comunicação multicast. Para obter mais informações sobre os parâmetros de objeto COMMINFO, consulte [DEFINE COMMINFO](#).

- **MCAST** Este parâmetro especifica se multicast é permitido nesta posição na árvore de tópicos.

Use o exemplo de linha de comandos a seguir para definir um objeto TOPIC para multicast:

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

Um novo objeto TOPIC chamado *ALLSPORTS* é criado. Ele possui uma sequência de tópicos *Sports*, seu objeto de informações de comunicação relacionado é chamado *MC1* (que é o nome especificado ao definir um objeto COMMINFO no exemplo anterior) e multicast é ativado.

Testando a publicação/assinatura multicast

Depois que os objetos TOPIC e COMMINFO foram criados, eles podem ser testados usando a amostra do amqspubc e a amostra do amqssubc. Para obter informações adicionais sobre estas amostras, consulte [Os programas de amostra de Publicação/Assinatura](#).

1. Abra duas janelas de linha de comandos; a primeira linha de comando é para a amostra de publicação do amqspubc e a segunda linha de comandos é para a amostra de assinatura do amqssubc.
2. Insira o comando a seguir na linha de comandos 1:

```
amqspubc Sports QM1
```

em que *Sports* é a sequência de tópicos do objeto TOPIC definido em um exemplo anterior, e *QM1* é o nome do gerenciador de filas.

3. Insira o comando a seguir na linha de comandos 2:

```
amqssubc Sports QM1
```

em que *Sports* e *QM1* são os mesmos usados na etapa “2” na página 373.

4. Insira `Hello world` na linha de comandos 1. Se a porta e o endereço IP especificados no objeto COMMINFO estiverem configurados corretamente; a amostra amqssubc, que está atendendo na porta para publicações do endereço especificado, gera `Hello world` na linha de comandos 2.

Topologia do tópico do IBM MQ Multicast

Utilize este exemplo para compreender a topologia do tópico do IBM MQ Multicast.

O suporte do IBM MQ Multicast requer que cada subárvore possua seu próprio grupo de multicast e fluxo de dados dentro da hierarquia total.

O esquema de endereçamento IP de *rede com classes* tem espaço de endereço designado para endereço multicast. A faixa completa de multicast de endereço IP é 224.0.0.0 a 239.255.255.255, mas alguns desses endereços são reservados. Para obter uma lista de endereços reservados entre em contato com o administrador do sistema ou consulte <https://www.iana.org/assignments/multicast-addresses> para obter informações adicionais. Recomenda-se o uso do endereço multicast com escopo definido localmente na faixa de 239.0.0.0 a 239.255.255.255.

No diagrama a seguir há dois possíveis fluxos de dados de multicast:

```
DEF COMMINFO(MC1) GRPADDR(239.XXX.XXX.XXX
)
DEF COMMINFO(MC2) GRPADDR(239.YYY.YYY.YYY)
```

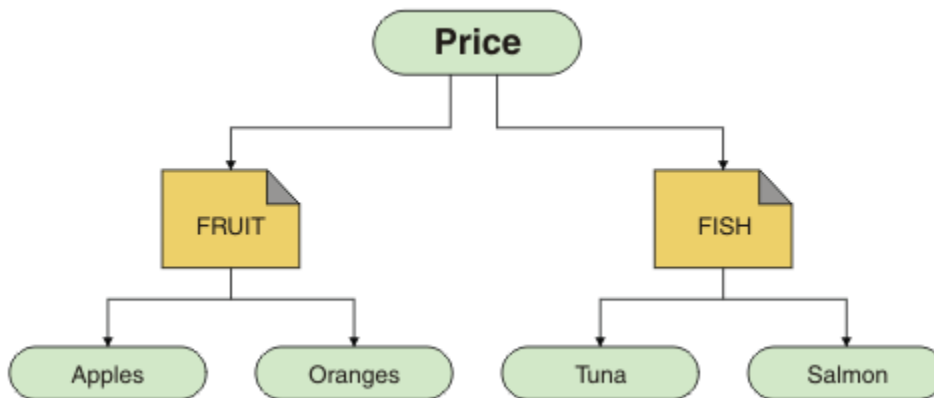
em que 239.XXX.XXX.XXX e 239.YYY.YYY.YYY são endereços multicast válido.

Essas definições de tópico são usadas para criar uma árvore de tópicos, conforme mostrado no diagrama a seguir:

```

DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)

```



Cada objeto de informações de comunicação multicast (COMMINFO) representa um fluxo de dados diferentes porque seus endereços do grupo são diferentes. Neste exemplo, o tópico FRUIT é definido para usar o objeto COMMINFO MC1, o tópico FISH é definido para usar o objeto COMMINFO MC2 e o nó Price não possui definições multicast.

IBM MQ Multicast tem um limite de 255 caracteres para sequências de tópicos. Essa limitação significa que os cuidados devem ser tomados com os nomes de nós e nós de folha dentro da árvore; se os nomes de nós e nós de folha forem muito longos, a sequência de tópicos poderá exceder 255 caracteres e retornar o código de razão 2425 (0979) (RC2425): MQRC_TOPIC_STRING_ERROR. É recomendável tornar sequências de tópicos o mais curtas possível porque sequências de tópicos mais longas podem ter um efeito negativo no desempenho.

Controlando o tamanho das mensagens multicast

Use essas informações para aprender sobre o formato de mensagem do IBM MQ e reduzir o tamanho de mensagens do IBM MQ.

IBM MQ mensagens têm um número de atributos associados a eles que estão contidas no descritor de mensagens. Para mensagens pequenas, esses atributos podem representar a maior parte do tráfego de dados e pode ter um efeito prejudicial significativo na taxa de transmissão. IBM MQ Multicast permite que o usuário configure qual, se houver algum, desses atributos são transmitidos junto com a mensagem.

A presença de atributos de mensagem, diferente de sequência de tópico, depende de se o objeto COMMINFO indica que eles devem ser enviados ou não. Se um atributo não for transmitido, o aplicativo de recebimento se aplicará a um valor padrão. Os valores padrão de MQMD não são necessariamente os mesmos do valor MQMD_DEFAULT e são descritos em [Tabela 19 na página 375](#).

O objeto COMMINFO contém o atributo MCPROP que controla quantos dos campos MQMD e propriedades do usuário são com a mensagem. Ao definir o valor deste atributo para um nível apropriado, é possível controlar o tamanho do IBM MQ Multicast mensagens:

MCPROP

As propriedades multicast controlam quantas propriedades do MQMD e propriedades do usuário são enviadas com a mensagem.

ALL

Todas as propriedades do usuário e todos os campos do MQMD são transmitidos.

RESPOSTA

Apenas as propriedades do usuário e os campos MQMD que lidam com a resposta das mensagens são transmitidos. Essas propriedades são:

- MsgType

- MessageId
- CorrelId
- ReplyToQ
- ReplyToQmgr

USUÁRIO


Apenas as propriedades do usuário são transmitidas.

Nenhum

Nenhuma propriedade do usuário ou campo do MQMD é transmitido.

COMPAT

Esse valor faz com que a transmissão da mensagem seja feita em um modo compatível com o RMM, o qual permite alguma interoperação com os aplicativos XMS atuais e os aplicativos do IBM Integration Bus RMM.

 XMS .NET Mensagens multicast (usando RMM) foi descontinuado de IBM MQ 9.2 e removido em IBM MQ 9.3.

Atributos de mensagens multicast

Atributos de mensagens podem vir de vários locais, como o MQMD, os campos no MQRFH2 e as propriedades de mensagem.

A tabela a seguir mostra o que acontece quando as mensagens são enviadas sujeitas ao valor de MCPROP (descrito anteriormente nesta seção) e o valor padrão utilizado quando um atributo não é enviado.

<i>Tabela 19. Atributos de Mensagens e como eles se relacionam com multicast</i>		
Atributo	Ação quando utilizar multicast	padrão se não transmitidos
TopicString	Sempre Incluído	Não-aplicável
MQMQ StrucId	Não transmitido	Não-aplicável
MQMD Versão	Não transmitido	Não-aplicável
Relatório	Incluído se não padrão	0
MsgType	Incluído se não padrão	MQMT_DATAGRAM
Expiração	Incluído se não padrão	0
Feedback	Incluído se não padrão	0
Codificação	Incluído se não padrão	MQENC_NORMAL(equiv)
CodedCharSetId	Incluído se não padrão	1208
Formato	Incluído se não padrão	MQRFH2
Priority	Incluído se não padrão	4
Persistence	Incluído se não padrão	MQPER_NOT_PERSISTENT
MsgId	Incluído se não padrão	Nulo
CorrelId	Incluído se não padrão	Nulo
BackoutCount	Incluído se não padrão	0
ReplyToQ	Incluído se não padrão	Em branco
ReplyToQMgr	Incluído se não padrão	Em branco
UserIdentifier	Incluído se não padrão	Em branco

Tabela 19. Atributos de Mensagens e como eles se relacionam com multicast (continuação)

Atributo	Ação quando utilizar multicast	padrão se não transmitidos
AccountingToken	Incluído se não padrão	Nulo
PutAppIType	Incluído se não padrão	MQAT_JAVA
PutAppIName	Incluído se não padrão	Em branco
PutDate	Incluído se não padrão	Em branco
PutTime	Incluído se não padrão	Em branco
ApplOriginData	Incluído se não padrão	Em branco
GroupID	Excluído	Não-aplicável
MsgSeqNumber	Excluído	Não-aplicável
Offset	Excluído	Não-aplicável
MsgFlags	Excluído	Não-aplicável
OriginalLength	Excluído	Não-aplicável
UserProperties	Incluído	Não-aplicável

Referências relacionadas

 [ALTER COMMINFO](#)
[DEFINE COMMINFO](#)

Ativando a conversão de dados para mensagens multicast

Use essas informações para entender como a conversão de dados funciona para o sistema de mensagens multicast do IBM MQ.

IBM MQ Multicast é um protocolo sem conexão, compartilhado e, portanto, não é possível que cada cliente faça solicitações específicas para a conversão de dados. Cada cliente inscrito ao mesmo fluxo multicast receberá os mesmos dados binários; portanto, se a conversão de dados do IBM MQ for necessária, a conversão será executada localmente em cada cliente.

Em uma instalação de plataforma mista, pode ser que a maioria dos clientes requeiram os dados em um formato que não seja o formato nativo do aplicativo de transmissão. Nessa situação, os valores **CCSID** e **ENCODING** do objeto multicast COMMINFO podem ser utilizados para definir a codificação da transmissão da mensagem para eficiência.

IBM MQ Multicast suporta a conversão de dados da carga útil da mensagem para a construção a seguir em formatos:

- MQADMIN
- MQEVENT
- MQPCF
- MQRFH
- MQRFH2
- MQSTR

Além desses formatos, também é possível definir seus próprios formatos e utilizar uma saída de conversão de dados [parâmetro MQDXP – saída de conversão de dados](#).

Para obter informações sobre programação de dados conversões, consulte [Conversão de dados no MQI para sistema de mensagens multicast](#).

Para obter mais informações sobre a conversão de dados, consulte [Conversão de dados](#).

Para obter mais informações sobre saídas de conversão de dados e `ClientExitPath`, consulte `ClientExitPath` subrotina do arquivo de configuração do cliente.

Monitoramento de aplicativo multicast

Use essas informações para aprender sobre como administrar e monitorar o IBM MQ Multicast.

O status do atual publicadores e assinantes para tráfego multicast (por exemplo, o número de mensagens enviadas e recebidas ou o número de mensagens perdidas) é transmitido periodicamente para o servidor do cliente. Quando o status é recebido, o atributo `COMMEV` do objeto `COMMINFO` especifica se o gerenciador de filas coloca ou não uma mensagem do evento em `SYSTEM.ADMIN.PUBSUB.EVENT`. A mensagem do evento contém as informações de status recebidas. Essa informação é um auxílio de diagnóstico de um valor inestimável para localizar a origem de um problema.

Use o comando do MQSC **DISPLAY CONN** para exibir as informações de conexão sobre os aplicativos conectados ao gerenciador de filas. Para obter mais informações sobre o comando **DISPLAY CONN**, consulte [DISPLAY CONN](#).

Use o comando MQSC **DISPLAY TPSTATUS** para exibir o status de seus publicadores e assinantes. Para obter mais informações sobre o comando **DISPLAY TPSTATUS**, consulte [DISPLAY TPSTATUS](#).

COMMEV e o indicador de confiabilidade de mensagem multicast

O *reliability indicator*, utilizado em conjunto com o atributo `COMMEV` do objeto `COMMINFO`, é um elemento-chave no monitoramento dos publicadores e assinantes do IBM MQ Multicast. O indicador de confiabilidade (o campo `MSGREL` que é retornado nos comandos de status de Publicação ou Assinatura) é um indicador do IBM MQ que ilustra a porcentagem de transmissões que não possui erros. Às vezes, as mensagens devem ser retransmitidas devido a um erro de transmissão, que é refletido no valor de `MSGREL`. Potenciais causas de erros de transmissão incluem assinantes lentos, redes ocupadas e interrupções na rede. `COMMEV` controla se as mensagens de eventos são geradas para manipulações multicast que são criadas utilizando o objeto `COMMINFO` e configuradas para um dos três valores possíveis:

DISABLED

Mensagens do evento não são gravadas.

ATIVADO

Mensagens de eventos são sempre gravadas, com uma frequência definida no parâmetro `MONINT` de `COMMINFO`.

EXCEÇÃO

Mensagens de eventos são gravadas se a confiabilidade da mensagem estiver sob o limite de confiabilidade. Um nível de confiabilidade de mensagem de 90% ou menos indica que pode haver um problema com a configuração de rede ou que um ou mais dos aplicativos de Publicação/Assinatura está em execução muito lenta:

- Um valor de **MSGREL (100, 100)** indica que não houve nenhum problema no curto prazo ou longo prazo.
- Um valor **MSGREL (80, 60)** indica que 20% das mensagens estão atualmente tendo problemas, mas que é igualmente uma melhoria no valor de longo prazo de 60.

Os clientes podem continuar a transmitir e receber tráfego multicast mesmo quando a conexão unicast para o gerenciador de filas é quebrada, portanto, os dados podem estar desatualizados.

Confiabilidade de mensagem multicast

Use estas informações para aprender como configurar o IBM MQ Multicast de assinatura e histórico de mensagens.

Um elemento-chave da superação da falha de transmissão com multicast é o armazenamento em buffer de dados transmitidos (um histórico de mensagens a serem mantidas no final da transmissão do link) por IBM MQ. Esse processo significa que nenhum buffer de mensagens é necessário no processo de colocação porque o IBM MQ fornece a confiabilidade. O tamanho deste histórico é configurado por meio

do objeto de informações de comunicação (COMMINFO), conforme descrito nas informações a seguir. Um buffer de transmissão maior significa que há mais histórico de transmissão a ser retransmitido, se necessário, mas devido à natureza do multicast, 100% de garantia de entrega não pode ser suportado.

O IBM MQ Multicast mensagem de histórico é controlado no objeto de informações de comunicação (COMMINFO) pelo atributo **MSGHIST**:

MSGHIST

Esse valor é a quantidade de histórico de mensagens em kilobytes que é mantida pelo sistema para manipular retransmissões no caso de NACKs (reconhecimentos negativos).

Um valor igual a 0 fornece o menor nível de confiabilidade. O valor padrão é 100 KB.

O IBM MQ Multicast histórico de nova assinatura é controlado no objeto as informações de comunicação (COMMINFO) pelo atributo **NSUBHIST**:

NSUBHIST

O novo histórico do assinante controla se um assinante que une-se a um fluxo de publicação recebe todos os dados que estiverem disponíveis atualmente ou recebe apenas as publicações feitas a partir do momento da assinatura.

Nenhum

Um valor de NONE faz o transmissor transmitir apenas a publicação feita a partir do momento da assinatura. NONE é o valor padrão.

ALL

Um valor ALL faz com que o transmissor retransmita todos os históricos de tópicos que forem conhecidos. Em algumas circunstâncias, esta situação pode causar um comportamento semelhante nas publicações retidas.

Nota: Usar um valor igual a ALL pode ter um efeito negativo no desempenho se houver um grande histórico do tópico, pois todo o histórico do tópico é retransmitido.

Referências relacionadas

[DEFINE COMMINFO](#)

 [ALTER COMMINFO](#)

Tarefas avançadas de multicast

Use estas informações para aprender sobre tarefas avançadas de administração de IBM MQ Multicast, como configuração de arquivos .ini e interoperabilidade com IBM MQ LLM.

Para considerações de segurança em uma instalação multicast, consulte [Multicast de segurança](#).

Ponte entre domínios de publicação/assinatura multicast e não multicast

Utilize essas informações para entender o que ocorre quando um publicador não multicast publica para um tópico do IBM MQ Multicast ativado.

Se um publicador não multicast publica em um tópico que está definido como ativado **MCAST** e **BRIDGE** ativado, o gerenciador de filas transmite a mensagem de saída através de multicast diretamente para todos os assinantes que podem estar atendendo. Um publicador não pode publicar para tópicos que não são ativados para multicast.

Tópicos existentes podem ser multicast ativados por meio da configuração dos parâmetros **MCAST** e **COMMINFO** de um objeto do tópico. Veja [Conceitos de multicast iniciais](#) para obter mais informações sobre esses parâmetros.

O atributo **BRIDGE** do objeto COMMINFO controla as publicações de aplicativos que não estão utilizando multicast. Se **BRIDGE** está configurado como ENABLED e o parâmetro **MCAST** do tópico também está configurado como ENABLED, as publicações de aplicativos que não estão utilizando multicast são vinculadas aos aplicativos que estão. Para obter mais informações sobre o parâmetro **BRIDGE**, consulte [DEFINE COMMINFO](#).

Configurando os arquivos .ini para Multicast

Use estas informações para entender os campos IBM MQ Multicast nos arquivos .ini.

A configuração Multicast IBM MQ adicional pode ser feita em um arquivo ini. O arquivo ini específico que deve ser utilizado é dependente do tipo de aplicativos:

- Cliente: configure o arquivo `MQ_DATA_PATH/mqclient.ini`.
- Gerenciador de filas: configure o arquivo `MQ_DATA_PATH/qmgrs/QMNAME/qm.ini`.

em que `MQ_DATA_PATH` é a localização do diretório de dados IBM MQ (`/var/mqm/mqclient.ini`) e `QMNAME` é o nome do gerenciador de filas ao qual o arquivo .ini se aplica.

O arquivo .ini contém campos usados para ajustar o comportamento de IBM MQ Multicast:

```
Multicast:
Protocol      = IP | UDP
IPVersion     = IPv4 | IPv6 | ANY | BOTH
LimitTransRate = DISABLED | STATIC | DYNAMIC
TransRateLimit = 100000
SocketTTL    = 1
Batch        = NO
Loop        = 1
Interface    = <IPAddress>
FeedbackMode = ACK | NACK | WAIT1
HeartbeatTimeout = 20000
HeartbeatInterval = 2000
```

Protocolo

UDP

Neste modo, os pacotes são enviados utilizando o protocolo UDP. Elementos de rede não pode fornecer assistência na distribuição de multicast como fazem em modo de IP no entanto. O formato do pacote permanece compatível com o PGM. Esse é o valor-padrão.

IP

Nesse modo, o transmissor envia pacotes de IP brutos. Elementos da rede com o assistente de suporte PGM na distribuição de pacote multicast confiável. Esse modo é totalmente compatível com o padrão PGM.

IPVersion

IPv4

Comunicar-se usando o protocolo IPv4 somente. Esse é o valor-padrão.

IPv6

Comunicar-se usando o protocolo IPv6 somente.

QUALQUER

Comunicar-se usando o IPv4, IPv6 ou ambos, dependendo de qual protocolo está disponível.

AMBOS

Suporta comunicação usando ambos IPv4 e IPv6.

LimitTransRate

DISABLED

Não há nenhum controle de taxa de transmissão. Esse é o valor-padrão.

STATIC

Implementa de controle de taxa de transmissão estático. O transmissor não transmitiria em uma taxa que excedesse a taxa especificada pelo parâmetro `TransRateLimit`.

DINÂMICO

O transmissor se adapta à sua taxa de transmissão de acordo com o feedback que obtém dos receptores. Neste caso, o limite de taxa de transmissão não pode ser maior que o valor especificado pelo parâmetro `TransRateLimit`. O transmissor tenta alcançar uma taxa de transmissão otimizada.

TransRateLimit

O limite da taxa de transmissão em Kbps.

SocketTTL

O valor de SocketTTL determina se o tráfego multicast pode passar por um roteador ou o número de roteadores pelos quais ele pode passar.

Batch

Controla se as mensagens são colocadas em lote ou enviadas imediatamente. Existem 2 valores possíveis:

- *NO* As mensagens não são colocadas em lote, elas são enviadas imediatamente.
- As mensagens são *YES* em lote.

Loop

Configure o valor para 1 para ativar loop multicast. loop Multicast define se os dados enviados são um circuito fechado para o host ou não.

Interface

O endereço IP da interface no qual os fluxos de tráfego multicast. Para obter mais informações e resolução de problemas, consulte: [Testando aplicativos multicast em uma rede não multicast e Configurando a rede apropriada para tráfego multicast](#)

FeedbackMode

NACK

Feedback, reconhecimentos negativos. Esse é o valor-padrão.

ACK

feedback por confirmações positivas.

WAIT1

Feedback por confirmações positivas quando o transmissor aguarda somente 1 ACK a partir de qualquer um dos receptores.

HeartbeatTimeout

O tempo limite de pulsação em milissegundos. Um valor igual a 0 indica que os eventos de tempo limite de pulsação não são gerados pelo receptor ou receptores do tópico. O valor padrão é 20000.

HeartbeatInterval

O intervalo de pulsação em milissegundos. Um valor igual a 0 indica que nenhuma pulsação é enviada. O intervalo de pulsação deve ser consideravelmente menor que o valor **HeartbeatTimeout** para evitar eventos de tempo limite de pulsação falsos. O valor-padrão é 2000.

Interoperabilidade multicast com o IBM MQ Low Latency Messaging

Use estas informações para entender a interoperabilidade entre o IBM MQ Multicast e IBM MQ Low Latency Messaging (LLM).

A transferência de carga útil básica é possível para um aplicativo que usa o LLM, com outro aplicativo usando multicast para trocar mensagens em ambas as direções. Embora multicast utilize tecnologia LLM, o próprio produto LLM não é integrado. Portanto, é possível instalar ambos, o LLM e o IBM MQ Multicast e operar e realizar manutenção nos dois produtos separadamente.

Aplicativos LLM que se comunicam com multicast podem precisar enviar e receber propriedades de mensagem. As propriedades de mensagem e os campos MQMD do IBM MQ são transmitidos como propriedades de mensagem de LLM com códigos de propriedade de mensagem LLM específicos conforme mostrado na tabela a seguir:

Propriedade IBM MQ	Tipo de propriedade LLM do IBM MQ	tipo de propriedade LLM	código de propriedade LLM
MQMD.Report	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1001
MQMD.MsgType	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1002

Tabela 20. Propriedades de mensagem do IBM MQ para mapeamento de propriedade LLM do IBM MQ (continuação)

Propriedade IBM MQ	Tipo de propriedade LLM do IBM MQ	tipo de propriedade LLM	código de propriedade LLM
MQMD.Expiry	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1003
MQMD.Feedback	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1004
MQMD.Encoding	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1005
MQMD.CodedCharSetId	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1006
MQMD.Format	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1007
MQMD.Priority	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1008
MQMD.Persistence	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1009
MQMD.MsgId	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_ByteArray	-1010
MQMD.BackoutCount	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1012
MQMD.ReplyToQ	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1013
MQMD.ReplyToQMger	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1014
MQMD.PutDate	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1020
MQMD.PutTime	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1021
MQMD.ApplOriginData	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1022
MQPubOptions	RMM_MSG_PROP_INT32	LLM_PROP_KIND_int32	-1053

Para obter mais informações sobre LLM, consulte a documentação do produto LLM: [IBM MQ Low Latency Messaging](#).

IBM i Administrando IBM MQ for IBM i

Comandos CL são o método preferencial para administrar IBM MQ em IBM i. Também é possível usar comandos MQSC, comandos PCF, comandos de controle e administração remota

Sobre esta tarefa

Tarefas de administração incluem criar, iniciar, alterar, visualizar, parar e excluir os clusters, processos e objetos IBM MQ (gerenciadores de filas, filas, listas de nomes, definições de processo, canais, canais de conexão do cliente, listeners, serviços e objetos de informações de autenticação).

Consulte os links a seguir para obter detalhes de como administrar o IBM MQ for IBM i

Conceitos relacionados

[Entendendo nomes de bibliotecas do gerenciador de filas do IBM MQ for IBM i](#)

[Serviços e componentes instaláveis no IBM i](#)

Tarefas relacionadas

[Mudando informações de configuração do IBM MQ em Multiplataformas](#)

[Configurando a Segurança em IBM i](#)

[“Chamando o manipulador de fila de devoluções no IBM i ..” na página 168](#)

No IBM MQ for IBM i, você chama o manipulador de DLQ configurando o comando **STRMQMDLQ**.

[Determinando problemas com aplicativos do IBM MQ for IBM i](#)

Referências relacionadas

[Objetos do Sistema e Padrão](#)

Gerenciando IBM MQ for IBM i usando os comandos de CL

Use essa informação para entender os comandos IBM MQ IBM i.

A maioria dos grupos de comandos do IBM MQ, incluindo aqueles associados a gerenciadores de filas, filas, tópicos, canais, listas, definições de processo e objetos de informações sobre autenticação pode ser acessada utilizando o comando relevante **WRK***.

O comando principal no conjunto é **WRKMQM**. Este comando permite, por exemplo, exibir uma lista de todos os gerenciadores de fila no sistema, juntamente com as informações de status. Alternativamente, é possível processar todos os comandos específicos do gerenciador de filas usando várias opções com relação a cada entrada.

No comando **WRKMQM**, é possível selecionar áreas específicas de cada gerenciador de filas, por exemplo, trabalhando com canais, tópicos ou filas e a partir desse local selecionar os objetos individuais.

Gravando definições de aplicativo IBM MQ

Ao criar ou customizar os aplicativos IBM MQ, é útil manter um registro de todas as definições de IBM MQ criadas. Este registro pode ser usado para:

- Propósitos de recuperação
- Manutenção
- A consolidação de aplicativos do IBM MQ

É possível registrar as definições de aplicativos IBM MQ de 1 de 2 maneiras:

1. Criando programas de CL para gerar seu IBM MQ as definições para o servidor.
2. Criando arquivos de texto MQSC como membros do SRC para gerar as definições de IBM MQ usando o comando de linguagem, IBM MQ.

Para obter detalhes adicionais sobre a definição dos objetos de fila, consulte [“Administrando o IBM MQ usando comandos MQSC”](#) na página 12 e [“Usando os formatos de comando programável do IBM MQ”](#) na página 27.

Referências relacionadas

[Referência de comandos CL do IBM MQ for IBM i](#)

Antes de iniciar a usar o IBM MQ for IBM i usando comandos CL

Utilize estas informações para iniciar o subsistema IBM MQ e crie um gerenciador de filas locais.

Antes de começar

Certifique-se de que o subsistema IBM MQ esteja em execução (utilizando o comando STRSBS QMQM/ QMQM) e que a fila de tarefas associadas a esse subsistema não está retida. Por padrão, o subsistema IBM MQ e a fila de tarefas são ambos denominados QMQM na biblioteca QMQM.

Sobre esta tarefa

Usando o IBM i da linha de comandos para iniciar um gerenciador de filas

Procedimento

1. Crie um gerenciador de filas local emitindo o comando CRTMQM a partir de uma linha de comandos do IBM i.

Quando você cria um gerenciador de filas, você tem a opção de tornar esse gerenciador de filas no gerenciador de filas padrão. O gerenciador de filas padrão (do qual pode haver somente um) é o gerenciador de filas para o qual um comando de CL se aplica, se o parâmetro nome do gerenciador de filas (MQMNAME) for omitido.

2. Inicie um gerenciador de filas locais emitindo o comando STRMQM a partir de uma linha de comandos do IBM i.

Se a inicialização do gerenciador de filas levar mais do que alguns segundos, o IBM MQ mostrará mensagens de status de forma intermitente detalhando o progresso de inicialização. Para obter mais informações sobre essas mensagens veja [Mensagens e códigos de razão](#).

Como proceder a seguir

É possível parar um gerenciador de filas emitindo o comando ENDMQM a partir da linha de comandos do IBM i e controlar um gerenciador de filas emitindo outros comandos do IBM MQ a partir de uma linha de comandos do IBM i.

Os gerenciadores de filas remotas não podem ser iniciados remotamente, mas devem ser criados e iniciados em seus sistemas por operadores locais. Uma exceção a isso é onde as instalações operacionais remotas (fora do IBM MQ for IBM i) existem para permitir tais operações.

O administrador de filas local não pode parar um gerenciador de filas remotas.

Nota: Como parte do quiesce de um sistema IBM MQ, você terá de colocar em modo quiesce os gerenciadores de filas ativos. Isso é descrito no [“Efetuando quiesce do IBM MQ for IBM i”](#) na página 453.

Criando Objetos do IBM MQ for IBM i

Use estas informações para entender os métodos para criar objetos do IBM MQ para IBM i.

Antes de começar

As tarefas a seguir sugerem várias maneiras nas quais é possível usar o IBM MQ for IBM i a partir da linha de comandos.

Sobre esta tarefa

Há dois métodos on-line para criar objetos do IBM MQ, que são:

Procedimento

1. Usando um comando Create, por exemplo: O comando **Create MQM Queue : CRTMQMQ**
2. Usando um comando Work with MQM object, seguido por F6, por exemplo: O comando **Work with MQM Queues : WRKMQMQ**

Como proceder a seguir

Para obter uma lista de todos os comandos, consulte [IBM MQ for IBM i comandos CL](#).

Nota: Todos os comandos MQM podem ser submetidas a partir do menu Comandos de Gerenciador de Filas de Mensagens. Para exibir este menu, digite GO CMDMQM na linha de comandos e pressione a tecla Enter.

O sistema exibe o painel de prompt automaticamente quando você seleciona um comando a partir desse menu. Para exibir o painel de orientação para um comando que você tenha digitado diretamente na linha de comandos, pressione F4 antes de pressionar a tecla Enter.

Criando uma fila local utilizando o comando CRTMQMQ

Procedimento

1. Digite CHGMQM na linha de comandos e pressione a tecla F4.
2. No **Painel Criar Fila MQM**, digite o nome da fila que deseja criar no campo Queue name . Para especificar um nome de maiúsculas e minúsculas, você coloque o nome entre apóstrofos.

3. Digite *LCL no campo Queue type.
4. Especifique um nome de gerenciador de filas, a menos que você esteja utilizando o gerenciador de filas padrão, e pressione a tecla Enter. É possível sobrescrever qualquer um dos valores com um novo valor. Rolar para frente para ver os campos adicionais. As opções utilizadas para clusters estão no final da lista de opções.
5. Quando você tiver mudado todos os valores, pressione a tecla Enter para criar a fila.

Criando uma fila local utilizando o comando WRKMQM

Procedimento

1. Digite WRKMQM na linha de comandos.
2. Insira o nome de um gerenciador de filas.
3. Se você deseja exibir o painel do prompt, pressione F4. O painel de prompt é útil para reduzir o número de filas exibidas, especificando um nome de fila genérica ou tipo de fila.
4. Pressione Enter e o **painel Trabalhar com Filas MQM** será exibido.. É possível sobrescrever qualquer um dos valores com um novo valor. Rolar para frente para ver os campos adicionais. As opções utilizadas para clusters estão no final da lista de opções.
5. Pressione F6 para criar uma nova fila; isso leva você ao painel **CRTMQM**. Consulte [“Criando uma fila local utilizando o comando CRTMQM”](#) na página 383 para obter instruções sobre como criar a fila. Quando você tiver criado a fila, o **painel Trabalhar com Filas MQM** é exibida novamente. A nova fila é incluída na lista ao pressionar F5=Refresh.

Mudando os Atributos do Gerenciador de Filas

Sobre esta tarefa

Para mudar os atributos do gerenciador de filas especificado no comando **CHGMQM**, especificando os atributos e valores que você deseja mudar. Por exemplo, utilize as seguintes opções para alterar os atributos de `jupiter.queue.manager`:

Procedimento

Digite **CHGMQM** na linha de comandos e pressione a tecla F4.

Resultados

O comando muda a fila de mensagens não entregues usada e permite eventos de inibição.

IBM i Trabalhando com filas locais no IBM i

Esta seção contém exemplos de alguns dos comandos que é possível utilizar para gerenciar filas locais. Todos os comandos mostrados também estão disponíveis utilizando opções do **Painel de comandos WRKMQM**.

Definindo uma fila local

Para um aplicativo, o gerenciador de filas locais é o gerenciador de filas ao qual o aplicativo está conectado. Filas que são gerenciadas pelo gerenciador de filas locais são consideradas locais para esse gerenciador de filas.

Use o comando **CRTMQM QTYPE *LCL** para criar uma definição de uma fila local e também para criar a estrutura de dados que é chamada de uma fila. Também é possível modificar as características da fila a partir daquelas da fila local padrão.

Neste exemplo, a fila que definimos, `orange.local.queue`, é especificada para ter estas características:

- Ela é ativada para gets, desativada para puts e opera em uma base primeiro a entrar, primeiro a sair (FIFO).
- Ela é *ordinária*, ou seja, não é uma fila de iniciação ou uma fila de transmissão e não gera mensagens do acionador.
- A profundidade máxima da fila é 1000 mensagens; o comprimento máximo da mensagem é 2000 bytes.

O comando a seguir faz isto no gerenciador de filas padrão:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL)
TEXT('Queue for messages from other systems')
PUTENBL(*NO)
GETENBL(*YES)
TRGENBL(*NO)
MSGDLYSEQ(*FIFO)
MAXDEPTH(1000)
MAXMSGLEN(2000)
USAGE(*NORMAL)
```

Nota:

1. USAGE *NORMAL indica que esta fila não é uma fila de transmissão.
2. Se você já tiver uma fila local com o nome `orange.local.queue` no mesmo gerenciador de filas, então, esse comando falhará. Use o atributo REPLACE *YES se você deseja sobrescrever a definição existente de uma fila, mas consulte também [“Mudando atributos de filas locais”](#) na página 386.

Definindo uma fila de mensagens não entregues

Cada gerenciador de filas deve ter uma fila local para ser utilizada como uma fila de mensagens não entregues de forma que mensagens que não puderem ser entregues ao seu destino correto possam ser armazenadas para recuperação posterior. Deve-se informar explicitamente o gerenciador de filas sobre a fila de mensagens não entregues. É possível fazer isso especificando uma fila de mensagens não entregues no comando **CRTMQM** ou é possível usar o comando **CHGMQM** para especificar um posteriormente. Também deve-se definir a fila de mensagens não entregues antes que ela possa ser utilizada.

Uma fila de mensagens não entregues de amostra chamada SYSTEM.DEAD.LETTER.QUEUE é fornecida com o produto. Esta fila é criada automaticamente quando você cria o gerenciador de filas. É possível modificar essa definição, se necessário. Não há necessidade de renomeá-la, embora você possa fazer isso, se desejar.

Uma fila de mensagens não entregues não possui requisitos especiais, exceto que:

- Ele deve ser uma fila local.
- Seu atributo MAXMSGL (comprimento máximo da mensagem) deve ativar a fila para acomodar as maiores mensagens que o gerenciador de filas possui para manipular **mais** o tamanho do cabeçalho de mensagens não entregues (MQDLH).

IBM MQ fornece um manipulador da fila de mensagens não entregues que permite que você especifique como as mensagens localizadas em uma fila de mensagens não entregues devem ser processadas ou removidas. Para obter mais informações, consulte [“Chamando o manipulador de fila de devoluções no IBM i ..”](#) na página 168.

Exibindo atributos do objeto padrão

Ao definir um objeto do IBM MQ, leva quaisquer atributos que você não especificar a partir do objeto padrão. Por exemplo, quando você define uma fila local, a fila herda quaisquer atributos que você omitir

na definição da fila local padrão, que é chamado `SYSTEM.DEFAULT.LOCAL.QUEUE`. Para ver exatamente quais são esses atributos, utilize o seguinte comando:

```
DSPMQMQ QNAME(SYSTEM.DEFAULT.LOCAL.QUEUE) MQMNAME(MYQUEUEMANAGER)
```

Copiando uma definição de fila local

É possível copiar uma definição de fila utilizando o comando `CPYMQMQ`. Por exemplo:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Esse comando cria uma fila com os mesmos atributos que a fila original `orange.local.queue`, em vez daqueles da fila local padrão do sistema.

Também é possível usar o comando **CPYMQMQ** para copiar uma definição de fila, mas substituindo uma ou mais mudanças nos atributos do original. Por exemplo:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('third.queue') MQMNAME(MYQUEUEMANAGER)
MAXMSGLEN(1024)
```

Este comando copia os atributos da fila `orange.local.queue` para a fila `third.queue`, mas especifica que o comprimento máximo da mensagem na fila de novo deve ser 1024 bytes, em vez de 2000.

Nota: Ao utilizar o comando **CPYMQMQ**, você copia os atributos da fila somente, não as mensagens na fila.

Mudando atributos de filas locais

É possível mudar os atributos de filas de duas maneiras, usando o comando **CHGMQMQ** ou o comando **CPYMQMQ** com o atributo `REPLACE *YES`. No “Definindo uma fila local” na página 384, você definiu a fila `orange.local.queue`. Se, por exemplo, você precisa aumentar o comprimento máximo da mensagem nessa fila para 10.000 bytes.

- Usando o comando **CHGMQMQ**:

```
CHGMQMQ QNAME('orange.local.queue') MQMNAME(MYQUEUEMANAGER) MAXMSGLEN(10000)
```

Esse comando muda um único atributo, aquele do comprimento máximo de mensagem; todos os outros atributos permanecem os mesmos.

- Usando o comando **CRTMQMQ** com a opção `REPLACE *YES`, por exemplo:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEMANAGER)
MAXMSGLEN(10000) REPLACE(*YES)
```

Este comando muda não somente o comprimento máximo da mensagem, mas todos os outros atributos que recebem seus valores padrão. A fila agora está ativada para put considerando que anteriormente ela estava desativada para put. Coloque ativado é o padrão, conforme especificado pela fila `SYSTEM.DEFAULT.LOCAL.QUEUE`, a menos que você tenha mudado.

Se você *diminuir* o comprimento máximo da mensagem em uma fila existente, as mensagens existentes não serão afetadas. Todas as mensagens novas, no entanto, devem atender aos novos critérios.

Limpando uma Fila Local

Para excluir todas as mensagens de uma fila local chamada `magenta.queue`, use o seguinte comando:

```
CLRMQMQ QNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Não é possível limpar uma fila se:

- Há mensagens não confirmadas que foram colocadas na fila sob o ponto de sincronização.
- Um aplicativo possui atualmente a fila aberta.

Excluindo uma fila local

Use o comando **DLTMQM** para excluir uma fila local.

Uma fila não pode ser excluída se tiver mensagens não confirmadas nela ou se estiver em uso.

Ativando Filas Grandes

IBM MQ suporta filas com mais de 2 GB. Consulte a documentação do sistema operacional para obter informações sobre como ativar o IBM i para suportar arquivos grandes.

As informações do produto IBM i podem ser localizadas em [IBM Documentation](#).

Alguns utilitários podem não ser capazes de lidar com arquivos com mais de 2 GB. Antes de ativar o suporte ao arquivo grande, verifique a documentação do seu sistema operacional para obter informações sobre as restrições sobre tal suporte.

Trabalhando com filas de alias no IBM i

Esta seção contém exemplos de alguns dos comandos que é possível utilizar para gerenciar filas de alias. Todos os comandos mostrados também estão disponíveis utilizando opções do **Painel de comandos WRKMQM**.

Uma fila de alias (algumas vezes conhecido como um alias da fila) fornece um método de redirecionamento de chamadas MQI. Uma fila de alias não é uma fila real, mas uma definição que resolve uma fila real. A definição da fila de alias contém um nome de fila de destino, que é especificado pelo atributo TGTQNAME.

Quando um aplicativo especifica uma fila de alias em uma chamada MQI, o gerenciador de filas resolve o nome de fila real no tempo de execução.

Por exemplo, um aplicativo foi desenvolvido para colocar mensagens em uma fila denominada `my.alias.queue`. Isso especifica o nome desta fila quando ela fizer uma solicitação **MQOPEN** e, indiretamente, se colocar uma mensagem nesta fila. O aplicativo não reconhece que a fila é uma fila de alias. Para cada chamada MQI que usa este alias, o gerenciador de filas resolve o nome de fila real, que poderia ser uma fila local ou uma fila remota definida neste gerenciador de filas.

Ao mudar o valor do atributo TGTQNAME, é possível redirecionar as chamadas MQI para outra fila, possivelmente em outro gerenciador de filas. Isso é útil para manutenção, migração e balanceamento de carga.

Definindo uma fila de alias

O seguinte comando cria uma fila de alias:

```
CRTMQMQ QNAME('my.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
MQMNAME(MYQUEUEMANAGER)
```

Este comando redireciona as chamadas MQI que especificam `my.alias.queue` para a fila `yellow.queue`. O comando não cria a fila de destino; as chamadas MQI falharão se a fila `yellow.queue` não existe no tempo de execução.

Se você alterar a definição de alias, poderá redirecionar as chamadas MQI para outra fila. Por exemplo:

```
CHGMQM QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Este comando redireciona as chamadas MQI para outra fila, `magenta.queue`.

Também é possível usar as filas de alias para fazer com que uma única fila (fila de destino) pareça ter diferentes atributos para diferentes aplicativos. Isso é feito definindo dois alias, um para cada aplicativo. Imagine que existem dois aplicativos:

- O aplicativo ALPHA pode colocar as mensagens no `yellow.queue`, mas não tem permissão para obter as mensagens dele.
- O aplicativo BETA pode obter as mensagens do `yellow.queue`, mas não tem permissão para colocar as mensagens nele.

É possível fazer isso utilizando os seguintes comandos:

```
/* This alias is put enabled and get disabled for application ALPHA */
CRTMQMQ QNAME('alphas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*YES) GETENBL(*NO) MQMNAME(MYQUEUEMANAGER)

/* This alias is put disabled and get enabled for application BETA */
CRTMQMQ QNAME('betas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*NO) GETENBL(*YES) MQMNAME(MYQUEUEMANAGER)
```

O ALPHA usa o nome da fila `alphas.alias.queue` em suas chamadas MQI; BETA usa o de nome da fila `betas.alias.queue`. Ambos acessam a mesma fila, mas de diferentes maneiras.

É possível utilizar o atributo `REPLACE *YES` quando você definir filas de alias, na mesma maneira que você usa esses atributos com as filas locais.

Usando outros comandos com filas de alias

É possível usar os comandos apropriados para exibir ou mudar os atributos da fila de alias. Por exemplo:

```
* Display the alias queue's attributes */
DSPMQMQ QNAME('alphas.alias.queue') MQMNAME(MYQUEUEMANAGER)

/* ALTER the base queue name, to which the alias resolves. */
/* FORCE = Force the change even if the queue is open. */

CHQMCMQ QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEMANAGER)
```

Trabalhando com filas modelo no IBM i

Esta seção contém exemplos de alguns dos comandos que é possível utilizar para gerenciar filas modelo. Todos os comandos mostrados também estão disponíveis utilizando opções do **Painel de comandos WRKMQMQ**.

Um gerenciador de filas cria uma fila dinâmica se receber uma chamada MQI de um aplicativo, especificando um nome de fila que foi definido como uma fila modelo. O nome da fila dinâmica nova é gerada pelo gerenciador de filas quando a fila é criada. Uma fila modelo é um modelo que especifica os atributos de qualquer fila dinâmica criada a partir dele.

Filas modelo fornecem um método conveniente para os aplicativos criarem filas conforme elas são necessárias.

Definindo uma fila modelo

É possível definir uma fila modelo com um conjunto de atributos da mesma maneira que você definir uma fila local. Filas modelos e filas locais possuem o mesmo conjunto de atributos, exceto que em filas modelos é possível especificar se as filas dinâmicas permanentes ou temporárias são criadas. (filas permanentes são mantidas entre reinícios do gerenciador de filas, os temporários não são). Por exemplo:

```
CRTMQMQ QNAME('green.model.queue') QTYPE(*MDL) DFNTYPE(*PERMDYN)
```

Este comando cria uma definição de fila modelo. A partir do atributo DFNTYPE, as filas reais criadas a partir deste modelo são filas dinâmicas permanentes. Os atributos não especificados são automaticamente copiados da fila padrão SYSYSTEM.DEFAULT.MODEL.QUEUE.

É possível usar o REPLACE *YES atributo quando você definir filas modelo, da mesma maneira que você os utilize com as filas locais.

Usando outros comandos com filas modelos

É possível utilizar os comandos apropriados para exibir ou alterar os atributos de uma fila modelo. Por exemplo:

```
/* Display the model queue's attributes */
DSPMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('green.model.queue')

/* ALTER the model queue to enable puts on any */
/* dynamic queue created from this model. */
CHGMQM MQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

IBM i

Trabalhando com acionamento no IBM i

Use essas informações para aprender sobre o acionamento e definições de processo.

IBM MQ fornece um recurso para iniciar um aplicativo automaticamente quando determinadas condições em uma fila são atendidas. Um exemplo das condições for quando o número de mensagens em uma fila atingir um número especificado. Este recurso é denominado *acionamento* e está descrito em detalhes em [Acionando Canais](#).

O que É o Acionamento?

O gerenciador de filas define certas condições que constituem os eventos acionadores. Se o acionamento estiver ativado para uma fila e um evento de acionador ocorrer, o gerenciador de filas envia uma mensagem de acionador para uma fila chamada uma fila de inicialização. A presença da mensagem do acionador na fila de inicialização indica que ocorreu um evento acionador.

As mensagens do acionador geradas pelo gerenciador de filas não são persistentes. Isto tem o efeito de redução de log (, portanto, melhor desempenho), minimizando as duplicadas durante o reinício, melhorando assim o tempo de reinício.

Qual é o monitor acionador?

O programa que processa a fila de inicialização é chamado de aplicativo monitor acionador e sua função é ler a mensagem do acionador e assumir a ação apropriada, com base nas informações contidas na mensagem do acionador. Normalmente essa ação seria iniciar algum outro aplicativo que processará a fila que fez com que a mensagem de acionador fosse gerada. Do ponto de vista do gerenciador de filas, não há nada de especial sobre o aplicativo monitor acionador – é outro aplicativo que lê as mensagens de uma fila (a fila de inicialização).

Mudando a atributos de submissão da tarefa do monitor do acionador

O monitor acionador fornecido como comando **STRMQMTRM** envia uma tarefa para cada mensagem do acionador usando o descritor de tarefa padrão do sistema, QDFTJOB. Isso tem limitações em que as tarefas submetidas são sempre chamadas QDFTJOB e têm os atributos da descrição de tarefa padrão, incluindo a lista de bibliotecas, *SYSVAL. IBM MQ fornece um método para substituir esses atributos. Por exemplo, é possível customizar as tarefas enviadas para ter nomes de tarefas mais significativos, conforme a seguir:

1. Na descrição da tarefa, especifique a descrição que desejar, por exemplo, valores de criação de log.
2. Especifique o Ambiente de dados da definição de processo utilizado no processo de acionamento:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)')
```

O monitor acionador executa um SBMJOB utilizando a descrição especificada.

É possível substituir outros atributos do SBMJOB, especificando a palavra-chave apropriado e valor no Ambiente de dados da definição de processo. A única exceção a isto é o CMD palavra-chave porque este atributo é preenchido pelo monitor do acionador. Um exemplo do comando para especificar os Dados do Ambiente da definição de processo no qual o nome da tarefa e descrição devem ser alterados da seguinte forma:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)  
JOB(TRIGGER)')
```

Definindo uma Fila do Aplicativo para o Acionamento

Uma fila do aplicativo é uma fila local que é usada pelos aplicativos para sistema de mensagens, por meio do MQI. O acionamento requer que inúmeros atributos de fila sejam definidos na fila do aplicativo. O acionamento em si é ativado pelo atributo TRGENBL.

Neste exemplo, um evento acionador deve ser gerado quando houver 100 mensagens de prioridade 5 ou superior na fila local `motor.insurance.queue`, conforme a seguir:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.insurance.queue') QTYPE(*LCL)  
PRCNAME('motor.insurance.quote.process') MAXMSGLEN(2000)  
DFTMSGPST(*YES) INITQNAME('motor.ins.init.queue')  
TRGENBL(*YES) TRGTYPE(*DEPTH) TRGDEPTH(100) TRGMSGPTY(5)
```

em que os parâmetros são:

MQMNAME(MYQUEUEMANAGER)

O nome do gerenciador de filas.

QNAME('motor.insurance.queue')

O nome da fila do aplicativo sendo definida.

PRCNAME('motor.insurance.quote.process')

O nome do aplicativo a ser iniciado por um programa do monitor acionador.

MAXMSGLEN(2000)

O comprimento máximo das mensagens na fila.

DFTMSGPST(*YES)

Mensagens nesta fila são persistentes por padrão.

INITQNAME('motor.ins.init.queue')

O nome da fila de inicialização na qual o gerenciador de filas deve colocar a mensagem do acionador.

TRGENBL(*YES)

O valor de atributo do acionador.

TRGTYPE(*DEPTH)

Um evento acionador é gerado quando o número de mensagens da prioridade necessária (**TRGMSGPTY**) atinge o número especificado em **TRGDEPTH**.

TRGDEPTH(100)

O número de mensagens necessárias para gerar um evento acionador.

TRGMSGPTY(5)

A prioridade de mensagens que devem ser contadas pelo gerenciador de filas ao decidir se deve gerar um evento acionador. Apenas as mensagens com prioridade 5 ou mais são contadas.

Definindo uma Fila de Inicialização

Quando ocorrer um evento do acionador, o gerenciador de filas colocará uma mensagem do acionador na fila de inicialização especificada na definição da fila do aplicativo. Filas de inicialização não possuem configurações especiais, mas é possível usar a seguinte definição na fila local `motor.ins.init.queue` para orientação:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.ins.init.queue') QTYPE(*LCL)
GETENBL(*YES) SHARE(*NO) TRGTYPE(*NONE)
MAXMSGL(2000)
MAXDEPTH(1000)
```

Criando uma Definição de Processo

Use o comando **CRTMQMPC** para criar uma definição de processo. Um processo de definição de uma fila do aplicativo com o aplicativo que está para processar as mensagens da fila. Isso é feito por meio do atributo `PRCNAME` no aplicativo de fila `motor.insurance.quote.process`, identificado neste exemplo:

```
CRTMQMPC MQMNAME(MYQUEUEMANAGER) PRCNAME('motor.insurance.quote.process')
TEXT('Insurance request message processing')
APPTYPE(*OS400) APPID(MQTEST/TESTPROG)
USRDATA('open, close, 235')
```

em que os parâmetros são:

MQMNAME(MYQUEUEMANAGER)

O nome do gerenciador de filas.

PRCNAME('motor.insurance.quote.process')

O nome da definição de processo.

TEXT('Insurance request message processing')

Uma descrição do programa de aplicativo ao qual desta definição se relaciona. Este texto é exibido ao usar o comando **DSPMQMPC**. Isso pode ajudá-lo a identificar o qual o processo faz. Se você usar os espaços na sequência, deve colocar a sequência entre aspas simples.

APPTYPE(*OS400)

O tipo do aplicativo a ser iniciado.

APPID(MQTEST/TESTPROG)

O nome do arquivo executável do aplicativo, especificado como um nome completo do arquivo.

USRDATA('open, close, 235')

dados definidos pelo usuário, que podem ser usados pelo aplicativo.

Exibindo sua definição de processo

Use o comando **DSPMQMPC** para examinar os resultados de sua definição. Por exemplo:

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPC('motor.insurance.quote.process')
```

Também é possível utilizar o comando **CHGMQMPC** para alterar uma definição de processo existente e o comando **DLTMQMPC** para excluir uma definição de processo.

Comunicação entre dois sistemas IBM MQ no IBM i

Esse exemplo de codificação ilustra como configurar dois sistemas IBM MQ for IBM i, usando comandos CL, para que eles possam se comunicar.

Os sistemas são chamados SYSTEMA e SYSTEMB e o protocolo de comunicação utilizado for TCP/IP.

Execute o seguinte procedimento:

1. Crie um gerenciador de filas no SYSTEMA, chamando-o QMGRA1.

```
CRTMQM  MQMNAME(QMGRA1) TEXT('System A - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

2. Inicie esse gerenciador de filas.

```
STRMQM  MQMNAME(QMGRA1)
```

3. Defina os objetos IBM MQ em SYSTEMA que você precisa para enviar mensagens para um gerenciador de filas em SYSTEMB

```
/* Transmission queue */
CRTMQMQ  QNAME(XMITQ.TO.QMGRB1) QTYPE(*LCL) +
MQMNAME(QMGRA1) TEXT('Transmission Queue +
to QMGRB1') MAXDEPTH(5000) USAGE(*TMQ)

/* Remote queue that points to a queue called TARGETB */
/* TARGETB belongs to queue manager QMGRB1 on SYSTEMB */
CRTMQMQ  QNAME(TARGETB.ON.QMGRB1) QTYPE(*RMT) +
MQMNAME(QMGRA1) TEXT('Remote Q pointing +
at Q TARGETB on QMGRB1 on Remote System +
SYSTEMB') RMTQNAME(TARGETB) +
RMTMQMNAME(QMGRB1) TMQNAME(XMITQ.TO.QMGRB1)

/* TCP/IP sender channel to send messages to the queue manager on SYSTEMB*/
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*SDR) +
MQMNAME(QMGRA1) TRPTYPE(*TCP) +
TEXT('Sender Channel From QMGRA1 on +
SYSTEMA to QMGRB1 on SYSTEMB') +
CONNNAME(SYSTEMB) TMQNAME(XMITQ.TO.QMGRB1)
```

4. Crie um gerenciador de filas no SYSTEMB, chamando-o QMGRB1.

```
CRTMQM  MQMNAME(QMGRB1) TEXT('System B - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

5. Inicie o gerenciador de filas no SYSTEMB.

```
STRMQM  MQMNAME(QMGRB1)
```

6. Defina os objetos IBM MQ necessários para receber mensagens do gerenciador de filas no SYSTEMA.

```
/* Local queue to receive messages on */
CRTMQMQ  QNAME(TARGETB) QTYPE(*LCL) MQMNAME(QMGRB1) +
TEXT('Sample Local Queue for QMGRB1')

/* Receiver channel of the same name as the sender channel on SYSTEMA */
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*RCVR) +
MQMNAME(QMGRB1) TRPTYPE(*TCP) +
TEXT('Receiver Channel from QMGRA1 to +
QMGRB1')
```

7. Finalmente, inicie um listener TCP/IP no SYSTEMB para que o canal possa ser iniciado. Este exemplo utiliza a porta padrão 1414.

```
STRMQMLSR MQMNAME(QMGRB1)
```

Você está pronto para enviar mensagens de teste entre SYSTEMA e SYSTEMB. Utilizando uma das amostras fornecidas, coloque uma série de mensagens a sua fila remota em SYSTEMA.

Inicie o canal em SYSTEMA, usando o comando **STRMQMCHL** ou usando o comando **WRKMQMCHL** e digitando uma solicitação de início (Opção 14) em relação ao canal emissor.

O canal deve ir para o status RUNNING e as mensagens são enviadas para a fila TARGETB em SYSTEMB.

Verifique as suas mensagens emitindo o comando:

```
WRKMQMMSG QNAME(TARGETB) MQMNAME(QMGRB1).
```

IBM i Definições de recurso de amostra no IBM i

Esta amostra contém o AMQSAMP4 de amostra do IBM i programa de CL.

```

/*****/
/*                                          */
/* Program name: AMQSAMP4                  */
/*                                          */
/* Description: Sample CL program defining MQM queues */
/*           to use with the sample programs */
/*           Can be run, with changes as needed, after */
/*           starting the MQM */
/*                                          */
/* <N_OCO_COPYRIGHT>                      */
/* Licensed Materials - Property of IBM */
/*                                          */
/* 63H9336 */
/* (c) Copyright IBM Corp. 1993, 2024. All Rights Reserved. */
/*                                          */
/* US Government Users Restricted Rights - Use, duplication or */
/* disclosure restricted by GSA ADP Schedule Contract with */
/* IBM Corp. */
/* <NOC_COPYRIGHT>                      */
/*                                          */
/*****/
/* Function:                               */
/*                                          */
/* AMQSAMP4 is a sample CL program to create or reset the */
/* MQI resources to use with the sample programs. */
/*                                          */
/* This program, or a similar one, can be run when the MQM */
/* is started - it creates the objects if missing, or resets */
/* their attributes to the prescribed values. */
/*                                          */
/*                                          */
/*                                          */
/* Exceptions signaled: none */
/* Exceptions monitored: none */
/*                                          */
/* AMQSAMP4 takes a single parameter, the Queue Manager name */
/*                                          */
/*****/
QSYS/PGM PARM(&QMGRNAME)

/*****/
/* Queue Manager Name Parameter */
/*****/
QSYS/DCL VAR(&QMGRNAME) TYPE(*CHAR)

/*****/
/* EXAMPLES OF DIFFERENT QUEUE TYPES */
/*                                          */
/* Create local, alias and remote queues */
/*                                          */
/* Uses system defaults for most attributes */
/*                                          */
/*****/
/* Create a local queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.LOCAL') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Sample local queue') /* description */+
SHARE(*YES) /* Shareable */+
DFTMSGPST(*YES) /* Persistent messages OK */

/* Create an alias queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ALIAS') +
MQMNAME(&QMGRNAME) +
```

```

QTYPE(*ALS) REPLACE(*YES)      +
+
TEXT('Sample alias queue')      +
DFTMSGPST(*YES) /* Persistent messages OK */+
TGTQNAME('SYSTEM.SAMPLE.LOCAL')

/* Create a remote queue - in this case, an indirect reference */
/* is made to the sample local queue on OTHER queue manager */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REMOTE') +
MQMNAME(&QMGRNAME) +
QTYPE(*RMT) REPLACE(*YES)      +
+
TEXT('Sample remote queue')/* description */+
DFTMSGPST(*YES) /* Persistent messages OK */+
RMTQNAME('SYSTEM.SAMPLE.LOCAL') +
RMTMQMNAME(OTHER) /* Queue is on OTHER */

/* Create a transmission queue for messages to queues at OTHER */
/* By default, use remote node name */
CRTMQMQ QNAME('OTHER') /* transmission queue name */+
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Transmission queue to OTHER') +
USAGE(*TMQ) /* transmission queue */

/*****
/* SPECIFIC QUEUES AND PROCESS USED BY SAMPLE PROGRAMS */
/*
/* Create local queues used by sample programs */
/* Create MQI process associated with sample initiation queue */
/*
/*****
/* General reply queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REPLY') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('General reply queue') +
DFTMSGPST(*NO) /* Not Persistent */

/* Queue used by AMQSINQ4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.INQ') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSINQ4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO) /* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSSET4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.SET') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSSET4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO)/* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSECH4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ECHO') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSECH4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO)/* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

```

```

/* Initiation Queue used by AMQSTRG4, sample trigger process */
CRTMQMQ QNAME('SYSTEM.SAMPLE.TRIGGER') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Trigger queue for sample programs')

/* MQI Processes associated with triggered sample programs */
/*
/***** Note - there are versions of the triggered samples *****/
/***** in different languages - set APPID for these *****/
/***** process to the variation you want to trigger *****/
/*
CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSINQ4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSINQ4') /* C +
/* APPID('QMOM/AMQ0INQ4') /* COBOL */ +
/* APPID('QMOM/AMQ3INQ4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSSET4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSSET4') /* C */ +
/* APPID('QMOM/AMQ0SET4') /* COBOL */ +
/* APPID('QMOM/AMQ3SET4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSECH4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSECH4') /* C */ +
/* APPID('QMOM/AMQ0ECH4') /* COBOL */ +
/* APPID('QMOM/AMQ3ECH4') /* RPG - ILE */

/*****
/*
/* Normal return.
/*
/*****
SNDPGMMSG MSG('AMQSAMP4 Completed creating sample +
objects for ' *CAT &QMGRNAME)
RETURN
ENDPGM

/*****
/*
/* END OF AMQSAMP4
/*
/*****

```



Maneiras alternativas de administrar o IBM MQ for IBM i

Usar os comandos CL é o método preferencial para administrar o IBM MQ for IBM i. No entanto, é possível usar vários outros métodos de administração, incluindo comandos MQSC, comandos PCF, comandos de controle e administração remota

Sobre esta tarefa

Você normalmente usa os comandos CL do IBM i para administrar o IBM MQ for IBM i. Para obter uma visão geral desses comandos, veja [“Gerenciando IBM MQ for IBM i usando os comandos de CL”](#) na página 382.

Também é possível usar comandos MQSC e comandos PCF conforme descrito nos subtópicos e é possível usar comandos de controle conforme descrito em [“Administrando o IBM MQ for Multiplatforms usando comandos de controle”](#) na página 10.

É possível usar eventos de instrumentação do IBM MQ para monitorar a operação de gerenciadores de filas. Consulte [Eventos de instrumentação](#) para obter informações sobre eventos de instrumentação do IBM MQ e como utilizá-los.

Use qualquer um dos métodos de administração descritos nos subtópicos a seguir como uma alternativa para usar comandos CL IBM i :

Administração local e remota no IBM i

Administre os objetos do IBM MQ for IBM i localmente ou remotamente.

Sobre esta tarefa

Administração local significa executar tarefas de administração em quaisquer gerenciadores de filas que você definiu no seu sistema local. No IBM MQ, é possível considerar isso como administração local porque nenhum canal do IBM MQ está envolvido, isto é, a comunicação é gerenciada pelo sistema operacional. Para executar esse tipo de tarefa, deve-se efetuar logon no sistema remoto e emitir os comandos de lá ou criar um processo que possa emitir os comandos para você.

IBM MQ suporta administração a partir de um único ponto através do que é conhecido como *remota de administração*. A administração remota consiste em enviar mensagens de controle de formato de comando programável (PCF) para o SYSTEM.ADMIN.COMMAND.QUEUE no gerenciador de filas de destino.

Existem diversas formas de gerar mensagens PCF. Eles são descritos nas etapas a seguir:

Procedimento

- Escreva um programa usando mensagens PCF. Consulte [“Administração usando comandos PCF no IBM i”](#) na página 398.
- Grave um programa usando MQAI, que envia mensagens PCF. Consulte [“Usando o MQAI para simplificar o uso de PCFs”](#) na página 38.
- Use o IBM MQ Explorer, disponível com IBM MQ for Windows, que permite usar uma interface gráfica com o usuário (GUI) e gera as mensagens PCF corretas. Consulte [“Usando o IBM MQ Explorer com o IBM MQ for IBM i”](#) na página 398.
- Use **STRMQMQSC** para enviar comandos indiretamente a um gerenciador de filas remotas. Consulte [“Administrando usando comandos MQSC no IBM i”](#) na página 396.

Por exemplo, é possível emitir um comando remoto para mudar uma definição de fila em um gerenciador de filas remotas.

Alguns comandos não podem ser emitidos desta forma, em especial, criar ou iniciar gerenciadores de filas e o comando iniciar servidores. Para executar esse tipo de tarefa, deve-se efetuar logon no sistema remoto e emitir os comandos de lá ou criar um processo que possa emitir os comandos para você.

Administrando usando comandos MQSC no IBM i

No IBM i, você cria uma lista de comandos em um arquivo de Script e, em seguida, executa o arquivo usando o comando **STRMQMQSC**. Utilize os comandos MQSC para gerenciar os objetos do gerenciador de filas, incluindo o próprio gerenciador de filas, filas, definições de processo, listas, canais, canais de conexão do cliente, listeners, serviços, tópicos e objetos de informações de autenticação.

Sobre esta tarefa

Os comandos de script do IBM MQ (MQSC) são gravados no formato legível, no texto EBCDIC. Você emite comandos do MQSC para um gerenciador de filas utilizando o comando CL **STRMQMQSC** IBM MQ. Este

método é um método de lote somente, obtendo sua entrada a partir de um arquivo físico de origem no sistema do servidor de bibliotecas. O nome padrão para este arquivo físico fonte é QMQSC.



Atenção: Não use a biblioteca QTEMP como a biblioteca de origem STRMQMMQSC, pois o uso da biblioteca QTEMP é limitado. Deve-se utilizar outra biblioteca como um arquivo de entrada para o comando.

Para portabilidade entre ambientes IBM MQ, limite o comprimento de linha em arquivos de comando MQSC a 72 caracteres. Use o sinal de mais para indicar que o comando é continuado na próxima linha

Os atributos do objeto especificados no MQSC são mostrados neste tópico em maiúsculas (por exemplo, RQMNAME), embora eles não tenham distinção entre maiúsculas e minúsculas.

Nota:

1. O formato de um arquivo MQSC não depende de sua localização no sistema de arquivos.
2. Os nomes de atributo do MQSC são limitados a oito caracteres.
3. Comandos MQSC estão disponíveis em todas as plataformas IBM MQ .

Para uma descrição de cada comando MQSC e sua sintaxe, veja [Comandos MQSC](#).

Procedimento

1. Crie o arquivo de origem QMQSC.

O IBM MQ for IBM i não fornece um arquivo de origem chamado QMQSC. Para processar comandos MQSC, deve-se criar o arquivo de origem QMQSC em uma biblioteca de sua escolha, emitindo o seguinte comando:

```
CRTSRCPF FILE(MYLIB/QMQSC) RCDLEN(240) TEXT('IBM MQ - MQSC Source')
```

2. Trabalhe com os membros..

A origem do MQSC está contida em membros dentro deste arquivo de origem. Para trabalhar com os membros digite o seguinte comando:

```
WRKMBRPDM MYLIB/QMQSC
```

Agora é possível incluir novos membros e manter existentes.

```
.
.
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +
DESCR(' ') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(1024) +
DEFSOPT(SHARED) +
NOHARDENBO +
USAGE(NORMAL) +
NOTRIGGER;
.
.
```

Figura 20. Extraída de um arquivo de comando MQSC, myprog.in, mostrando um comando MQSC (DEFINE QLOCAL) com seus atributos.

Informações relacionadas

[Administrando o IBM MQ usando comandos MQSC](#)

Administração usando comandos PCF no IBM i

O objetivo de comandos do IBM MQ programmable command format (PCF) é permitir que as tarefas de administração seja programado para um programa de administração. Dessa maneira é possível criar filas e definições de processo e mudar gerenciadores de filas a partir de um programa.

Os comandos PCF abrangem a mesma faixa de funções fornecidas pelos comandos MQSC. No entanto, ao contrário de comandos MQSC, os comandos PCF e suas respostas não estão em um formato de texto que você possa ler.

É possível gravar um programa para emitir comandos PCF para qualquer gerenciador de filas na rede a partir de um único nó. Desta maneira, é possível centralizar e automatizar as tarefas de administração.

Cada comando PCF é uma estrutura de dados que está integrado na parte de dados do aplicativo em uma mensagem do IBM MQ. Cada comando é enviado ao gerenciador de filas de destino usando a função MQPUT MQI da mesma maneira que qualquer outra mensagem. O servidor de comandos no gerenciador de filas que recebe a mensagem interpreta-a como uma mensagem de comando e executa o comando. Para obter as respostas, o aplicativo emite uma chamada MQGET e os dados de resposta são retornados em outra estrutura de dados. O aplicativo pode então processar a resposta e agir de acordo.

Resumidamente, estas são algumas das coisas que o programador do aplicativo deve especificar para criar uma mensagem de comando PCF:

Descritor de Mensagens

Esse é um descritor de mensagem padrão do IBM MQ, no qual:

- Tipo de mensagem (*MsgType*) é MQMT_REQUEST.
- Formato da mensagem (*Format*) é MQFMT_ADMIN.

Dados do aplicativo

Contém a mensagem PCF que inclui o cabeçalho PCF, em que:

- O tipo de mensagem PCF (*Type*) especifica MQCFT_COMMAND.
- O identificador de comando especifica o comando, por exemplo, *Change Queue* (MQCMD_CHANGE_Q)

PCFs de escape são comandos PCF que contêm comandos MQSC no texto de mensagem. É possível usar PCFs para enviar comandos para um gerenciador de filas remotas. Consulte [“Usando o MQAI para simplificar o uso de PCFs”](#) na página 38 para obter informações adicionais.

Para obter uma descrição completa das estruturas de dados PCF e como implementá-las, consulte [Estruturas para comandos e respostas](#).

Usando o IBM MQ Explorer com o IBM MQ for IBM i

Use estas informações para administrar o IBM MQ for IBM i usando o IBM MQ Explorer.

IBM MQ for Windows (plataforma x86) e IBM MQ para Linux (plataformas x86 e x86-64) fornecem uma interface de administração denominado IBM MQ Explorer para executar tarefas de administração como uma alternativa para o uso de CL, controle ou comandos MQSC.

O IBM MQ Explorer permite executar a administração local ou remota de sua rede a partir de um computador que esteja executando o Windows (plataforma x86) ou Linux (plataformas x86 e x86-64), apontando o IBM MQ Explorer nos gerenciadores de filas e clusters de seu interesse.

Com o IBM MQ Explorer, é possível:

- Iniciar e parar um gerenciador de filas (somente em sua máquina local).
- Definir, exibir e alterar as definições de objetos do IBM MQ, tais como filas, tópicos e canais.
- Procurar as mensagens em uma fila.
- Iniciar e parar um canal.
- Visualizar as informações de status sobre um canal.

- Visualizar gerenciadores de filas em um cluster.
- Verificar para ver quais aplicativos, usuários ou canais possuem uma fila específica aberta.
- Criar um novo cluster de gerenciadores de filas usando o assistente **Criar Novo Cluster**.
- Incluir um gerenciador de filas em um cluster usando o assistente **Incluir Gerenciador de Filas para Cluster**.
- Gerenciar o objeto de informações sobre autenticação, usado com a segurança do canal Segurança da Camada de Transporte (TLS).

Usando a orientação on-line, é possível:

- Definir e controlar vários recursos incluindo gerenciadores de filas, filas, canais, definições de processo, canais de conexão do cliente, listeners, tópicos, serviços, listas e clusters.
- Iniciar ou parar um gerenciador de filas e seus processos associados.
- Visualizar os gerenciadores de filas e seus objetos associados em sua estação de trabalho ou de outras estações de trabalho.
- Verificar o status de gerenciadores de filas, clusters e canais.

Assegure-se de que os requisitos a seguir tenham sido satisfeitos antes de tentar usar o IBM MQ Explorer para gerenciar o IBM MQ em uma máquina servidor. Verifique se:

1. Um servidor de comandos está em execução para qualquer gerenciador de filas que estiver sendo administrado, iniciado no servidor pelo comando de CL **STRMQMSVR**.
2. Um listener TCP/IP adequado existe para cada gerenciador de filas remotas. Este é o listener IBM MQ iniciado pelo comando **STRMQMLSR**.
3. O canal de conexão do servidor, chamado SYSTEM.ADMIN.SVRCONN, existe em cada gerenciador de filas remotas. Você mesmo deve criar esse canal. Ele é obrigatório para cada gerenciador de filas remotas está sendo administrado. Sem ela, administração remota não é possível.
4. Verifique se a fila SYSTEM.MQEXPLORER.REPLY.MODEL existe.

Gerenciando o servidor de comandos para administração remota no IBM i

Use estas informações para aprender sobre a administração remota de IBM MQ for IBM i comando do servidor.

Cada gerenciador de filas pode ter um servidor de comandos associado a ele. Um servidor de comandos processa qualquer comando de entrada a partir de gerenciadores de filas remotas ou comandos PCF a partir de aplicativos. Ele apresenta os comandos para o gerenciador de filas para processamento e retorna um código de conclusão ou mensagem do operador dependendo da origem do comando.

Um servidor de comandos é obrigatório para toda a administração que envolva PCFs, o MQAI e também para administração remota.

Nota: Para administração remota, deve-se assegurar que o gerenciador de filas de destino esteja em execução. Caso contrário, as mensagens que contêm comandos não podem deixar o gerenciador de filas a partir do qual elas são emitidas. Em vez disso, essas mensagens são enfileiradas na fila de transmissão local que serve ao gerenciador de filas remotas. Evite esta situação, se for possível.

Há comandos de controle separados para iniciar e parar o servidor de comandos. É possível executar as operações descritas nas seções seguintes usando o IBM MQ Explorer.

Iniciando e Parando o servidor de comandos

Para iniciar o servidor de comandos, utilize este comando CL:

```
STRMQMSVR MQMNAME('saturn.queue.manager')
```

em que `saturn.queue.manager` é o gerenciador de filas para o qual o servidor de comandos está sendo iniciado.

Para parar o servidor de comandos, utilize um dos seguintes comandos de CL:

1.

```
ENDMQMSVR MQMNAME('saturn.queue.manager') OPTION(*CNTRLD)
```

para executar uma parada controlada, em que `saturn.queue.manager` é o gerenciador de filas para o qual o servidor de comandos está sendo parado. Esta é a opção padrão, o que significa que o `OPTION(*CNTRLD)` pode ser omitido.

2.

```
ENDMQMSVR MQMNAME('saturn.queue.manager') OPTION(*IMMED)
```

para executar uma parada imediata, em que `saturn.queue.manager` é o gerenciador de filas para o qual o servidor de comandos está sendo parado.

Exibindo o status do servidor de comandos

Para administração remota, assegure que o servidor de comandos no gerenciador de filas de destino esteja em execução. Se ele não estiver em execução, os comandos remotos não podem ser processados. Todas as mensagens que contêm comandos são enfileiradas na fila de comandos do gerenciador de filas de destino do `SYSTEM.ADMIN.COMMAND.QUEUE`.

Para exibir o status do servidor de comandos para um gerenciador de filas, chamado aqui `saturn.queue.manager`, o comando CL é:

```
DSPMQMSVR MQMNAME('saturn.queue.manager')
```

Emita este comando na máquina de destino. Se o servidor de comandos está executando, o painel mostrado na é exibido: [Figura 21 na página 400](#)

```
Display MQM Command Server (DSPMQMSVR)
```

```
Queue manager name . . . . . > saturn.queue.manager
```

```
MQM Command Server Status. . . . > RUNNING
```

```
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys
```

Figura 21. Exibir MQM painel do Servidor de Comandos

Executando comandos do console da web

Você deve configurar seu ambiente conforme descrito no texto a seguir, para que os comandos Qshell relacionados ao console da web sejam executados corretamente em IBM MQ for IBM i.

Sobre esta tarefa

Quando o Qshell é iniciado, ele inicializa tabelas internas para processamento de comandos com base no CCSID da tarefa. Para os comandos Qshell relacionados ao console da web para executar corretamente em IBM i, deve-se configurar seu ambiente.

Um código do idioma é configurado definindo a variável de ambiente LANG para o nome do caminho para um objeto do código do idioma. Por exemplo, para configurar o código do idioma para inglês dos EUA, a variável de ambiente LANG é configurada da seguinte forma:

```
LANG=/QSYS.LIB/EN_US.LOCALE
```

No Qshell, é possível verificar a configuração pelo conjunto de comandos de emissão para listar todas as variáveis de ambiente. Geralmente, é LANG que pode impactar o código do idioma para o ambiente de tempo de execução. Ele também pode ter LC_ALL.

Para executar comandos Qshell corretamente, a configuração do ambiente de código de idioma deve ser consistente com a configuração de sua tarefa.

Procedimento

Use o comando CL DSPJOB JOB (JobNumber/USERProfile/JobName)

- a) Selecione a opção 2 para exibir os atributos de definição de tarefa.
- b) Os atributos a seguir devem ser consistentes com a configuração do ambiente LANG ou LC_ALL

- Identificador de idioma
- Identificador de país ou região
- Identificador do conjunto de caracteres codificados

Por exemplo, se

```
LANG=/QSYS.LIB/FR_FR.LOCALE
```

, os atributos da tarefa deverão ser:

- ID do idioma..... FRA
- ID do país ou região..... FR
- ID do conjunto de caracteres codificados.... 297

Como proceder a seguir

Para obter mais informações sobre o suporte ao idioma nacional, consulte o tópico do IBM Documentation [Considerações sobre o suporte ao idioma Nacional \(NLS\)](#).

IBM i

Gerenciamento de trabalho para IBM i

Estas informações descrevem a maneira como o IBM MQ manipula solicitações de trabalho e detalha as opções disponíveis para priorizar e controlar as tarefas associadas com o IBM MQ.

Aviso

Não altere os objetos do gerenciamento de trabalho do IBM MQ a menos que você compreenda totalmente os conceitos do gerenciamento de trabalho do IBM i e IBM MQ.

Informações adicionais sobre subsistemas e descrições de tarefas podem ser localizadas em [Gerenciamento de trabalho](#) na documentação do produto IBM i. Preste bastante atenção às seções em [Iniciando tarefas](#) e [Tarefas em lote](#).

O IBM MQ for IBM i incorpora o ambiente do IBM i UNIX e os encadeamentos do IBM i. **Não** faça qualquer mudança nos objetos no Sistema de Arquivos Integrado (IFS).

Durante operações normais, um gerenciador de filas do IBM MQ inicia um número de tarefas em lote para desempenhar diferentes tarefas. Por padrão, essas tarefas em lote executadas no subsistema QMQM que é criado quando o IBM MQ é instalado.

O gerenciamento de trabalho refere-se ao processo de adaptação de tarefas do IBM MQ para obter o melhor desempenho de seu sistema ou para tornar a administração mais simples.

Por exemplo, você pode:

- mudar a prioridade de execução das tarefas para tornar o gerenciador de filas mais responsivo do que o outro.
- Redirecionar a saída de inúmeras tarefas para uma determinada fila de saída.
- Fazer com que todas as tarefas de um determinado tipo sejam executadas em um subsistema específico.
- Isolar os erros em um subsistema.

O gerenciamento de trabalho é realizado criando ou mudando as descrições da tarefa associada às tarefas do IBM MQ. É possível configurar o gerenciamento de trabalho para:

- Uma instalação inteira do IBM MQ.
- Gerenciadores de fila individuais.
- Tarefas individuais para gerenciadores de filas individuais.

IBM i Tarefas do IBM MQ para IBM i

Esta é uma tabela das tarefas do IBM MQ for IBM i e uma breve descrição de cada delas.

Quando um gerenciador de filas está em execução, você verá algumas ou todas as seguintes tarefas em lote em execução sob o perfil do usuário QMQM no subsistema IBM MQ. As tarefas estão brevemente descritas em [Tabela 21 na página 402](#).

É possível visualizar todas as tarefas conectadas a um gerenciador de filas usando a opção 22 no painel **Trabalhar com Gerenciador de Filas** (WRKMQM). É possível visualizar os listeners utilizando o comando WRKMQLSR.

Nome da tarefa	Função
AMQZMUC0	Gerenciador de utilitário. Esta tarefa executa os utilitários do gerenciador de filas crítico, por exemplo, o gerenciador de sequência de diário.
AMQZXMA0	O controlador de execução que é a primeira tarefa iniciado pelo gerenciador de filas. Ele manipula as solicitações MQCONN e inicia os processos do agente para processar chamadas API do IBM MQ.
AMQZFUMA	Gerenciador de autoridade de objeto (OAM).
AMQZLAA0	Os agentes do gerenciador de filas que executam a maior parte do trabalho para aplicativos que se conectam ao gerenciador de filas utilizando MQCNO_STANDARD_BINDING.
AMQZLSA0	Agente do gerenciador de filas.
AMQZMUFO	Gerenciador de Utilitários
AMQZMGRO	Controlador de processo. Essa tarefa é usada para iniciar e gerenciar listeners e serviços.
AMQZMURO	Gerenciador de utilitário. Esta tarefa executa os utilitários do gerenciador de filas crítico, por exemplo, o gerenciador de sequência de diário.
AMQFQPUB	Daemon de publicação/assinatura enfileirado.
AMQFCXBA	Tarefa do trabalhador do broker.
RUNMQBRK	Tarefa de controle do broker.
AMQRMPPA	Tarefa de agrupamento de processo do canal.
AMQCRSTA	Respondente do canal chamado pelo TCP/IP.

Tabela 21. Tarefas da IBM MQ. (continuação)

Nome da tarefa	Função
AMQCRS6B	Canal receptor da LU62 e conexão do cliente (consulte a nota).
AMQRRMFA	Gerenciador de Repositório para clusters.
AMQCLMAA	Listener TCP/IP não encadeado.
AMQPSEA	Processador de comandos PCF que manipula PCF e solicitações de administração remota.
RUNMQTRM	Monitor acionador.
RUNMQDLQ	Manipulador da fila de mensagens não entregues.
RUNMQCHI	O inicializador de canais.
RUNMQCHL	Tarefa de canal emissor que é iniciada para cada canal emissor.
RUNMQLSR	Listener de TCP/IP encadeado.
AMQRCMLA	Processador de comandos MQSC e PCF do canal.

Nota: A tarefa do receptor LU62 executa no subsistema de comunicações e obtém suas propriedades de tempo de execução das entradas de roteamento e comunicações que são usadas para iniciar a tarefa. Consulte [Terminal iniciado \(Receptor\)](#) para obter mais informações.

IBM i Objetos de gerenciamento de trabalho no IBM i

Quando o IBM MQ está instalado, vários objetos são fornecidos na biblioteca QMQM para ajudar com o gerenciamento de trabalho. Esses objetos são aqueles necessários para tarefas do IBM MQ a serem executadas em seu próprio subsistema.

Descrições de tarefa de amostra são fornecidas para duas das tarefas em lote do IBM MQ. Se nenhuma descrição de tarefa específica é fornecida para uma tarefa do IBM MQ, ele executará com a descrição de tarefa padrão QMQMJOB.

Os objetos de gerenciamento de trabalho fornecidos ao instalar o IBM MQ são listados em [Tabela 22](#) na página 403 e os objetos criados para um gerenciador de filas são listados em [Tabela 23](#) na página 404

Nota: Os objetos de gerenciamento de trabalho podem ser localizados na biblioteca QMQM e os objetos do gerenciador de filas podem ser localizados na biblioteca do gerenciador de filas.

Tabela 22. Objetos de gerenciamento de trabalho

Nome	tipo	Descrição
AMQZLAA0	*JOB	A descrição da tarefa que é utilizada pelo agente de processos do IBM MQ
AMQZLSA0	*JOB	O agente do gerenciador de filas de ligações isoladas
AMQZXMA0	*JOB	A descrição da tarefa que é usada pelos controladores de execução do IBM MQ
QMQM	*SBS	O subsistema no qual todas as tarefas de execução do IBM MQ
QMQM	*JOB	A fila de tarefas conectadas ao subsistema fornecida
QMQMJOB	*JOB	O padrão da descrição da tarefa do IBM MQ, utilizado se não houver uma descrição de tarefa específica para uma tarefa
QMQMMSG	*MSG	A fila de mensagens padrão para tarefas do IBM MQ.

Tabela 22. Objetos de gerenciamento de trabalho (continuação)

Nome	tipo	Descrição
QMQMRUN20	*CLS	Uma descrição de classe alta prioridade para tarefas do IBM MQ
QMQMRUN35	*CLS	Uma descrição da classe para prioridade de tarefas do IBM MQ
QMQMRUN50	*CLS	Uma descrição de classe baixa prioridade para IBM MQ tarefas

Tabela 23. Objetos de gerenciamento de trabalho criados para um gerenciador de filas

Nome	tipo	Descrição
AMQA000000	*JRNRCV	Receptor de diário local
AMQAJRN	*JRN	Diário local
AMQJRNINF	*USRSPC	Espaço do usuário que é atualizado com os mais recentes receptores de diário necessários para inicialização e recuperação de mídia de um gerenciador de filas. Este espaço de usuário pode ser consultado por um aplicativo para determinar quais receptores de diário de arquivamento e que pode ser excluído com segurança.
AMQAJRNMSG	*MSGQ	fila de mensagens do diário local
AMQCRC6B	*PGM	Programa para iniciar a conexão LU6.2
AMQRFOLD	*FILE	arquivo de definição de canal do gerenciador de filas migrado
QMQMMSG	*MSGQ	fila de mensagens do gerenciador de filas

Como o IBM MQ usa objetos de gerenciamento de trabalho no IBM i

Estas informações descrevem a maneira na qual o IBM MQ usa os objetos de gerenciamento de trabalho e fornece exemplos de configuração.



Atenção: Não altere as configurações de entrada da fila de tarefas no subsistema QMQM para limitar o número de tarefas permitidas no subsistema por prioridade. Se tentar fazer isso, é possível parar a execução de tarefas essenciais do IBM MQ após serem enviados e causa a inicialização do gerenciador de filas falhar.

Para entender como configurar o gerenciamento de trabalho, deve-se primeiro entender como o IBM MQ usa descrições de tarefa.

A descrição da tarefa utilizada para iniciar a tarefa controla vários atributos da tarefa. Por exemplo:

- A fila de tarefas na qual a tarefa está enfileirada e em qual subsistema a tarefa é executada.
- Os dados de roteamento utilizados para iniciar a tarefa e a classe que a tarefa utiliza para seus parâmetros de tempo de execução.
- A fila de saída que a tarefa usa para imprimir arquivos.

O processo de iniciar uma tarefa do IBM MQ podem ser consideradas em três etapas:

1. IBM MQ selecionará uma descrição da tarefa.

IBM MQ usa a técnica a seguir para determinar qual descrição de tarefa utilizar para uma tarefa do lote:

- a. Consulte na biblioteca do gerenciador de filas para uma descrição de tarefa com o mesmo nome que a tarefa. Veja [Entendendo IBM MQ for IBM i nomes da biblioteca do gerenciador de filas](#) para obter mais detalhes sobre a biblioteca do gerenciador de filas.

- b. Procure na biblioteca do gerenciador de filas para a descrição de tarefa padrão QMQMJOB.
 - c. Consulte na biblioteca QMQM para uma descrição de tarefa com o mesmo nome que a tarefa.
 - d. Utilize a descrição de tarefa padrão, QMQMJOB, na biblioteca QMQM.
2. A tarefa é submetida para a fila de tarefas.

Descrições de tarefa fornecidas com o IBM MQ foram configuradas, por padrão, para colocar tarefas na fila de tarefas QMQM na biblioteca QMQM. A fila de tarefas QMQM está conectada ao subsistema QMQM fornecido, portanto, por padrão, as tarefas começam a executar no subsistema QMQM.

3. A tarefa entra no subsistema e passa pelas etapas de roteamento.

Quando a tarefa entra no subsistema, os dados de roteamento especificados na descrição da tarefa são utilizados para localizar entradas de roteamento para a tarefa.

Os dados de roteamento devem corresponder a uma das entradas de roteamento definidas no subsistema QMQM e isto define qual das classes fornecidas (QMQMRUN20, QMQMRUN35 ou QMQMRUN50) é utilizada pela tarefa.

Nota: Se as tarefas do IBM MQ não parecerem estar iniciando, certifique-se de que o subsistema esteja executando e a fila de tarefas não esteja retida,

Se você tiver modificado os objetos de gerenciamento de trabalho do IBM MQ, certifique-se de que tudo está associada corretamente. Por exemplo, se você especificar uma fila de tarefas diferente de QMQM/QMQM na descrição da tarefa, certifique-se de que uma ADDJOBQE seja executada para o subsistema, isto é, QMQM.

É possível criar uma descrição da tarefa para cada tarefa documentada em [Tabela 21 na página 402](#) usando a planilha a seguir como um exemplo:

```

What is the queue manager library name? _____
Does job description AMQZXMA0 exist in the queue manager library? Yes   No
Does job description QMQMJOB exist in the queue manager library? Yes   No
Does job description AMQZXMA0 exist in the QMQM library?           Yes   No
Does job description QMQMJOB exist in the QMQM library?           Yes   No

```

Se você responder Não para todas estas perguntas, crie uma tarefa global descrição QMQMJOB na biblioteca QMQM.

A fila de mensagens do IBM MQ

Um IBM MQ de mensagens da fila, QMQMMSG, é criado em cada biblioteca do gerenciador de filas. Mensagens do sistema operacional são enviadas para esta fila quando as tarefas do gerenciador de filas terminam e o IBM MQ envia mensagens para a fila. Por exemplo, para relatar quais receptores de diário são necessárias na inicialização. Mantenha o número de mensagens nesta fila de mensagens em um tamanho gerenciável para torná-lo mais fácil de controlar.

Exemplos do sistema padrão para IBM i

Estes exemplos mostram como uma instalação do IBM MQ não modificada funciona quando algumas das tarefas padrão são apresentadas no momento da inicialização do gerenciador de filas.

Primeiro, a tarefa do controlador de execução AMQZXMA0 inicia.

1. Emita o comando **STRMQM** para o gerenciador de filas TESTQM.
2. IBM MQ procura a biblioteca do gerenciador de filas QMTESTQM, primeiramente pela descrição de tarefa AMQZXMA0 e, em seguida, pela descrição de tarefa QMQMJOB.

Nenhuma dessas descrições de tarefa existe, assim, o IBM MQ procura pela descrição de tarefa AMQZXMA0 na biblioteca do produto QMQM. Essa descrição da tarefa existe, portanto, é utilizada para enviar a tarefa.

3. A descrição da tarefa utiliza a fila de tarefas padrão do IBM MQ, portanto, a tarefa será enviada à fila de tarefas QMQM/QMQM.

4. Os dados de roteamento na descrição da tarefa AMQZXMA0 são QMQMRUN20, portanto, o sistema procura as entradas de roteamento do subsistema para uma que corresponda a esses dados.

Por padrão, a entrada de roteamento com número de sequência 9900 possui dados de comparação que correspondem a QMQMRUN20, portanto, a tarefa é iniciada com a classe definida nessa entrada de roteamento, que também é chamado QMQMRUN20.

5. A classe QMQM/QMQMRUN20 foi executada com prioridade configurada como 20, portanto, a tarefa AMQZXMA0 é executada no subsistema QMQM com a mesma prioridade que a maioria das tarefas interativas no sistema.

Configurando exemplos de gerenciamento de trabalho no IBM i

Use estas informações para aprender como é possível mudar e criar descrições de tarefas do IBM MQ para mudar os atributos de tempo de execução de tarefas do IBM MQ.

A chave para a flexibilidade do IBM MQ de trabalho de gerenciamento reside na maneira em que o IBM MQ procura as descrições de tarefa:

- Se você criar ou mudar as descrições da tarefa em uma biblioteca do gerenciador de filas, essas mudanças substituirão as descrições da tarefa global no QMQM, mas as mudanças são locais e afetam esse gerenciador de filas particular sozinho.
- Se você criar ou mudar as descrições de tarefa global na biblioteca QMQM, as descrições de tarefa afetarão todos os gerenciadores de filas no sistema, a menos que substituído localmente para gerenciadores de filas individuais.

1. O exemplo a seguir aumenta a prioridade de tarefas de controle de canal para um gerenciador de filas individual.

Para tornar o gerenciador de repositório e tarefas do inicializador de canais, AMQRRMFA e RUNMQCHI, execute o mais rápido possível para o gerenciador de filas TESTQM, execute as seguintes etapas:

a. Crie duplicatas locais da descrição de tarefa QMQM/QMQMJOBDB com os nomes dos processos do IBM MQ que você deseja controlar na biblioteca do gerenciador de filas. Por exemplo:

```
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMTESTQM)
NEWOBJ (RUNMQCHI)
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMTESTQM)
NEWOBJ (AMQRRMFA)
```

b. Mude o parâmetro dos dados de roteamento na descrição da tarefa para assegurar que as tarefas usem a classe QMQMRUN20.

```
CHGJOBDB JOBDB(QMTESTQM/RUNMQCHI) RTGDTA('QMQMRUN20')
CHGJOBDB JOBDB(QMTESTQM/AMQRRMFA) RTGDTA('QMQMRUN20')
```

As tarefas AMQRRMFA e RUNMQCHI para o gerenciador de filas TESTQM agora:

- Usam as novas descrições de tarefa local na biblioteca do gerenciador de filas
- Executam com prioridade 20 porque a classe QMQMRUN20 é utilizada quando as tarefas entram no subsistema.

2. O exemplo a seguir define uma nova classe de prioridade de execução para o subsistema QMQM.

a. Crie uma classe duplicada na biblioteca QMQM para permitir que outros gerenciadores de filas acessem a classe emitindo o comando a seguir:

```
CRTDUPOBJ OBJ(QMQMRUN20) FROMLIB(QMQM) OBJTYPE(*CLS) TOLIB(QMQM)
NEWOBJ (QMQMRUN10)
```

b. Mude a classe para ter a nova prioridade de execução emitindo o comando a seguir:

```
CHGCLS CLS(QMQM/QMQMRUN10) RUNPTY(10)
```

- c. Inclua a nova definição de classe para o subsistema emitindo o comando a seguir:

```
ADDRTGE SBSDB(QMQM/QMQM) SEQNBR(8999) CMPVAL('QMQMRUN10') PGM(QSYS/QCMD)  
CLS(QMQM/QMQMRUN10)
```

Nota: É possível especificar qualquer valor numérico para o número de sequência de roteamento, mas os valores devem estar em ordem sequencial. Este número de sequência informa ao subsistema a ordem na qual as entradas de roteamento devem ser pesquisadas para uma correspondência dos dados de roteamento.

- d. Mude a descrição da tarefa local ou global para utilizar a nova classe de prioridade emitindo o comando a seguir:

```
CHGJOB JOB(QMQMlibname/QMQMJOB) RTGDTA('QMQMRUN10')
```

Agora, todas as tarefas do gerenciador de filas associado ao QMlibraryname utilizar uma prioridade de execução de 10.

3. O exemplo a seguir executa um gerenciador de filas em seu próprio subsistema

Para fazer com que todas as tarefas para o gerenciador de filas TESTQM executem no subsistema QBATCH, execute as etapas a seguir:

- a. Crie uma duplicata local da descrição de tarefa QMQM/QMQMJOB na biblioteca do gerenciador de filas com o comando

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMTESTQM)
```

- b. Mude o parâmetro da fila de tarefas na descrição da tarefa para assegurar que as tarefas utilizem a fila de tarefas QBATCH.

```
CHGJOB JOB(QMTESTQM/QMQMJOB) JOBQ(*LIBL/QBATCH)
```

Nota: A fila de tarefas está associada à descrição do subsistema. Se você achar que as tarefas estão permanecendo na fila de tarefas, verifique se a definição de fila de tarefas está definida no SBSDB. Use o comando DSPSBSDB para o subsistema e assuma a opção 6, Entradas da fila de tarefas.

Agora, todas as tarefas para o gerenciador de filas TESTQM:

- Utilizam a nova descrição de tarefa padrão local na biblioteca do gerenciador de filas
- São enviadas à fila de tarefas QBATCH.

Para assegurar que as tarefas sejam roteadas e priorizadas corretamente:

- Crie entradas de roteamento para as tarefas do IBM MQ no subsistema QBATCH ou
- Confie em uma entrada de roteamento de depósito que chame QCMD, independentemente do quais dados de roteamento sejam utilizados.

Essa opção funciona somente se a opção de tarefas ativas máximas para a fila de tarefas QBATCH for definida como *NOMAX. O padrão do sistema é 1.

4. O exemplo a seguir cria outro subsistema IBM MQ

- a. Crie um subsistema duplicado na biblioteca QMQM, emitindo o comando a seguir:

```
CRTDUPOBJ OBJ(QMQM) FROMLIB(QMQM) OBJTYPE(*SBSDB) TOLIB(QMQM) NEWOBJ(QMQM2)
```

- b. Remova a fila de tarefas QMQM emitindo o comando a seguir:

```
RMVJOBQE SBS(D(QMQM/QMQM2) JOBQ(QMQM/QMQM)
```

- c. Crie uma nova fila de tarefas para o subsistema emitindo o comando a seguir:

```
CRTJOBQ JOBQ(QMQM/QMQM2) TEXT('Job queue for IBM MQ Queue Manager')
```

- d. Inclua uma entrada da fila de tarefas para o subsistema emitindo o comando a seguir:

```
ADDJOBQE SBS(D(QMQM/QMQM2) JOBQ(QMQM/QMQM2) MAXACT(*NOMAX)
```

- e. Crie um QMQMJOB(D duplicado na biblioteca do gerenciador de filas emitindo o comando a seguir:

```
CRTDUPOBJ OBJ(QMQMJOB(D) FROMLIB(QMQM) OBJTYPE(*JOB(D) TOLIB(QMlibraryname)
```

- f. Mude a descrição da tarefa para utilizar a nova fila de tarefas emitindo o comando a seguir;

```
CHGJOB(D JOB(D(QMlibraryname/QMQMJOB(D) JOBQ(QMQM/QMQM2)
```

- g. Inicie o subsistema emitindo o comando a seguir:

```
STRSBS SBS(D(QMQM/QMQM2)
```

Nota:

- a. É possível especificar o subsistema em qualquer biblioteca. Se por algum motivo o produto for reinstalado ou a biblioteca QMQM substituída, todas as mudanças feitas serão removidas.
- b. Todas as tarefas do gerenciador de filas associadas à QMlibraryname agora executam no subsistema QMQM2.

IBM i

Disponibilidade, backup, recuperação e reinicialização no IBM i

Utilize estas informações para entender como o IBM MQ for IBM i usa o suporte ao registro no diário do IBM i para ajudar sua estratégia de backup e restauração.

Deve-se estar familiarizado com os métodos de backup e recuperação padrão do IBM i e com o uso de diários e seus receptores de diários associados em IBM i, antes de ler esta seção. Para obter informações sobre esses tópicos, consulte [Backup e recuperação](#).

Para entender a estratégia de backup e recuperação, primeiro você precisa para entender como o IBM MQ for IBM i organiza seus dados no IBM i sistema de arquivos e o sistema de arquivos integrado (IFS).

IBM MQ for IBM i mantém seus dados em uma biblioteca individual para cada instância do gerenciador de filas e em arquivos de fluxo no sistema de arquivos IFS.

As bibliotecas específicas do gerenciador de fila contêm diários, receptores de diário e objetos necessários para controlar o gerenciamento de trabalho do gerenciador de filas. Os diretórios e arquivos IFS contêm arquivos de configuração do IBM MQ, as descrições de objetos do IBM MQ e os dados que eles contêm.

Cada mudança nesses objetos, que são recuperáveis em uma falha do sistema, é registrada em um diário *antes* de ser aplicada ao objeto apropriado. O efeito disso é que tais mudanças podem ser recuperadas reproduzindo as informações registradas no diário.

É possível configurar o IBM MQ for IBM i para usar instâncias do gerenciador de filas múltiplos em diferentes servidores para fornecer disponibilidade do gerenciador de filas aumentado e acelerar a recuperação no caso de um servidor ou falha do gerenciador de filas.

Use estas informações para entender como o IBM MQ for IBM i usa diários em sua operação para controlar atualizações em objetos locais.

Cada biblioteca do gerenciador de filas contém um diário para esse gerenciador de fila e o diário possui o nome QM *GRLIB*/AMQ *A* *JRN*, em que QM *GRLIB* é o nome da biblioteca do gerenciador de filas e *A* é uma carta, *A* no caso de um gerenciador de fila de instância única, que é exclusivo para a instância do gerenciador de filas

QM *GRLIB* tem o nome QM, seguido pelo nome do gerenciador de filas em um formulário exclusivo. Por exemplo, um gerenciador de filas chamado TEST tem um gerenciador de biblioteca denominada QMTEST. A biblioteca de gerenciador de filas pode ser especificada ao criar um gerenciador de filas usando o comando **CRTMQM**.

Diários ter associado os receptores de diário que contêm as informações que estão sendo registradas atualmente. Os receptores são objetos aos quais as informações podem somente ser anexadas e finalmente preenchidas.

Os receptores de diário utilizam o espaço em disco de valor com informações desatualizadas. No entanto, é possível colocar as informações em armazenamento permanente para minimizar esse problema. Um receptor de diário for conectado ao diário em qualquer momento determinado. Se o receptor de diário atingir seu limite de tamanho predeterminado, ele será desconectado e substituído por um novo receptor de diário. É possível especificar o limite de receptores de diário quando se cria um gerenciador de filas usando **CRTMQM** e o parâmetro **THRESHOLD**.

Os receptores de diário associados ao local do IBM MQ para IBM i diário existir em cada biblioteca do gerenciador de filas e adotam uma convenção de nomenclatura, conforme a seguir:

```
AMQ Arnnnnn
```

em que

A

é uma letra A-Z.. É A para gerenciadores de filas de instância única. Ele varia para diferentes instâncias de um gerenciador de filas de várias instâncias.

nnnnn

é decimal 00000 to 99999 que é incrementado por 1 para o próximo diário na sequência..

R

é decimal 0 to 9, que é incrementado em 1 cada vez que um receptor é restaurado..

A sequência dos diários se baseia em data. No entanto, a nomenclatura do diário seguinte se baseia nas regras a seguir:

1. AMQA1nnnnn vai para AMQA1(nnnnn+1) e nnnnn agrupa quando atinge 99999. Por exemplo, AMQA099999 vai para AMQA000000e AMQA999999 vai para AMQA900000.
2. Se um diário com um nome gerado pela regra 1 já existir, a mensagem CPI70E3 será enviada para a fila de mensagens QSYSOPR e a comutação automática do receptor para.

O receptor atualmente conectado continuará a ser usado até que você investigar o problema e manualmente conectar um novo receptor.

3. Se nenhum novo nome está disponível na sequência (ou seja, todos os possíveis nomes do diário estão no sistema), você precisa fazer ambos:
 - a. Exclua diários que não são mais necessários (consulte [“Gerenciamento de diário no IBM i”](#) na página 414).
 - b. Registre as mudanças de diário no receptor de diário mais recente utilizando (**RCDMQMIMG**) e, em seguida, repita a etapa anterior. Isso permite que os nomes de receptor de diário antigos sejam reutilizados.

O diário AMQAJRN usa a opção MNGRCV (*SYSTEM) para ativar o sistema operacional a mudar automaticamente os receptores de diário quando o limite for atingido. Para obter mais informações sobre como o sistema gerencia os receptores, consulte *IBM i Backup e Recuperação*.

O valor padrão do limite do receptor de diário é 100.000 KB. É possível configurar isso para um valor maior ao criar o gerenciador de filas. O valor inicial do atributo LogReceiverSize é gravado na sub-rotina LogDefaults do arquivo mq.s.ini.

Quando um receptor de diário se estende além do seu limite especificado, o receptor for desconectado e um novo receptor de diário é criado, herdando os atributos do receptor anterior. Mudanças nos atributos LogReceiverSize ou LogASP depois que um gerenciador de filas foi criado são ignoradas quando o sistema anexa automaticamente um novo receptor de diário

Consulte [Alterando IBM MQ as informações de configuração em Multiplataformas](#) para obter detalhes adicionais sobre como configurar o sistema

Se você precisar mudar o tamanho dos receptores de diário após o gerenciador de filas ter sido criado, crie um novo receptor de diário e configure seu proprietário para QMQM usando os seguintes comandos:

```
CRTJRNRCV JRNRCV(QM GRLIB/AMQ Arnnnnn) THRESHOLD(xxxxxx) +  
TEXT('MQM LOCAL JOURNAL RECEIVER')  
CHGOBJOWN OBJ(QM GRLIB/AMQ Arnnnnn) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
```

em que

QMGRLIB

É o nome da biblioteca do gerenciador de filas

A

É o identificador de instância (geralmente A).

rnrrrrr

É o receptor de diário seguinte na sequência de nomenclatura descrita anteriormente

xxxxxx

É o limite do receptor novo (em KB)

Nota: O tamanho máximo do receptor é controlado pelo sistema operacional. Para verificar esse valor, consulte a palavra-chave THRESHOLD no comando **CRTJRNRCV**.

Agora anexe o novo receptor ao diário AMQAJRN com o comando:

```
CHGJRN JRN(QMGRLIB/AMQ A JRN) JRNRCV(QMGRLIB/AMQ Arnnnnn)
```

Consulte [“Gerenciamento de diário no IBM i”](#) na página 414 para obter detalhes sobre como gerenciar esses receptores de diário.

IBM i *Uso de diário do gerenciador de filas no IBM i*

Use estas informações para entender como o IBM MQ for IBM i usa diários em sua operação para controlar atualizações em objetos locais.

As atualizações persistentes para filas de mensagens acontecem em duas fases. Os registros que representa a atualização são primeiro gravados no diário, então, o arquivo de fila é atualizado.

Os receptores de diário podem, portanto, se tornar mais atualizados que os arquivos de fila. Para garantir que o processamento de reinicialização comece a partir de um ponto de verificação consistente, o IBM MQ usa pontos de verificação.

Um ponto de verificação é um momento em que o registro descrito no diário é o mesmo que o registro na fila. O próprio ponto de verificação consiste nas séries de registros de diário necessários para reiniciar o gerenciador de filas. Por exemplo, o estado de todas as transações (ou seja, unidades de trabalho) ativas no momento do ponto de verificação.

Os pontos de verificação são gerados automaticamente pelo IBM MQ. Eles são executados quando o gerenciador de filas é iniciado e encerrado e após o registro de um determinado número de operações.

É possível forçar um gerenciador de filas para obter um ponto de verificação emitindo o comando RCDMQMIMG em relação a todos os objetos em um gerenciador de filas e exibindo os resultados, conforme a seguir:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

Conforme as filas manipulam mensagens adicionais, o registro de ponto de verificação se torna inconsistente com o estado atual das filas.

Quando o IBM MQ for reiniciado, ele localiza o último registro de ponto de verificação no log. Essas informações são mantidas no arquivo de ponto de verificação que é atualizado no final de cada ponto de verificação. O registro de ponto de verificação representa o ponto de consistência mais recente entre o log e os dados. Os dados desse ponto de verificação é utilizado para reconstruir as filas como existiam no momento de verificação. Quando as filas são recriadas, o log é, então, transmitido para trazer as filas de volta para o estado em que estavam antes de o sistema falha ou fechar.

Para entender como o IBM MQ utiliza o diário, considere o caso de uma fila local chamada TESTQ no gerenciador de filas TEST. Isso é representado pelo arquivo IFS:

```
/QIBM/UserData/mqm/qmgrs/TEST/queues
```

Se uma mensagem especificada é colocada nessa fila e, em seguida, recuperada da fila, as ações que ocorrem são mostrados na Figura do [Figura 22 na página 411](#).

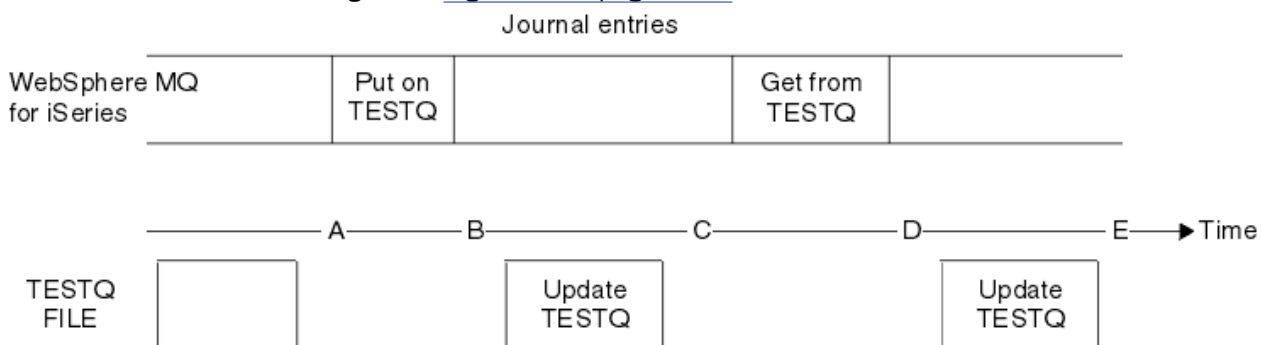


Figura 22. Sequência de eventos ao atualizar objetos MQM

Os cinco pontos, A até E, mostrados no diagrama representam pontos no tempo que definem os estados a seguir:

- A** A representação de arquivo IFS da fila é consistente com as informações contidas no diário.
- B** Uma entrada de diário é gravada no diário definindo uma operação Put na fila..
- C** A atualização apropriada é feita para a fila.
- D** Uma entrada de diário for enviada para o diário que define uma operação Get da fila.
- E** A atualização apropriada é feita para a fila.

A chave para os recursos de recuperação do IBM MQ for IBM i é que o usuário pode salvar a representação de arquivo do IFS de TESTQ como no tempo de A e depois recuperar a representação de arquivo do IFS de TESTQ como no tempo E, restaurando o objeto salvo e reproduzindo as entradas no diário a partir de tempo A em diante.

Essa estratégia é usado pelo IBM MQ for IBM i para recuperar mensagens persistentes após falha do sistema. IBM MQ mantém uma determinada entrada na receptores de diário e assegura que em inicialização ele reproduz as entradas no diário a partir desse ponto em diante. Esta entrada de

inicialização é periodicamente recalculada para que o IBM MQ tenha somente que executar a reprodução mínima necessária na próxima inicialização.

IBM MQ fornece recuperação individual de objetos. Todas as informações persistentes relacionadas a um objeto é registrado no local do IBM MQ for IBM i diários. Qualquer objeto do IBM MQ que se torna danificado ou corrompido pode ser completamente reconstruído a partir das informações contidas no diário.

Para obter mais informações sobre como o sistema gerencia os receptores, consulte [“Disponibilidade, backup, recuperação e reinicialização no IBM i”](#) na página 408.

IBM i *Imagens de mídia no IBM i*

No IBM i, uma imagem de mídia é uma cópia completa de um objeto IBM MQ que é registrado no diário. Alguns objetos corrompidos ou danificados podem ser recuperados automaticamente por meio de suas imagens de mídia.

Um objeto do IBM MQ de longa duração pode representar um grande número de entradas no diário, voltando para o ponto em que ele foi criado. Para evitar isso, o IBM MQ for IBM i tem o conceito de uma imagem de mídia de um objeto.

Essa imagem de mídia é uma cópia completa do IBM MQ objeto registradas no diário. Se uma imagem de um objeto é tomada, o objeto pode ser reconstruído pela reprodução das entradas no diário a partir dessa imagem posterior. A entrada no diário que representa o ponto de reprodução para cada objeto do IBM MQ é referida como sua entrada de recuperação de mídia. IBM MQ mantém o controle do:

- Entrada de recuperação de mídia para cada objeto do gerenciador de filas.
- Entrada mais antiga a partir de dentro deste conjunto (consulte a mensagem de erro AMQ7462 em [“Gerenciamento de diário no IBM i”](#) na página 414 para obter detalhes.

Imagens do objeto *CTLG e do objeto *MQM são executadas regularmente porque esses objetos são cruciais para reiniciar o gerenciador de filas.

As imagens de outros objetos são obtidas quando conveniente. Por padrão, as imagens de todos os objetos são tomadas quando um gerenciador de filas é encerrado usando o comando **ENDMQM** com o parâmetro ENDCCTJOB(*YES). Essa operação pode levar um tempo considerável para gerenciadores de filas muito grandes. Se você precisar encerrar rapidamente, especifique o parâmetro RCDMQMIMG(*NO) com ENDCCTJOB(*YES). Em tais casos, você é recomendado a registrar uma imagem de mídia completa nos diários após o gerenciador de filas ter sido reiniciado, utilizando o comando a seguir:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME)
```

IBM MQ automaticamente registros de uma imagem de um objeto, se encontrar um ponto conveniente em que um objeto pode ser descrito por modo compacto uma pequena entrada no diário. No entanto, isso pode nunca acontecer para alguns objetos, por exemplo, filas que sempre contêm grandes números de mensagens.

Em vez de permitir que a data da entrada de recuperação de mídia mais antiga continue por um período desnecessariamente longo, use o comando IBM MQ RCDMQMIMG, que permite obter uma imagem de objetos selecionados manualmente.

A recuperação a partir de imagens de mídia

IBM MQ recuperará automaticamente alguns objetos a partir de sua imagem de mídia se for reconhecido que estão corrompidos ou danificados. Em particular, isso se aplica ao *MQM especiais e objetos *CTLG como parte da inicialização do gerenciador de filas normal. Se qualquer transação de ponto de sincronização estava incompleta no momento do último encerramento do gerenciador de filas, qualquer fila afetada também é recuperada automaticamente para concluir a operação de inicialização.

Deve-se recuperar objetos de outros manualmente, utilizando o comando do IBM MQ RCRMQMOBJ. Este comando reproduz as entradas no diário para recriar o objeto IBM MQ. Se um objeto do IBM MQ se

tornarem danificadas, as ações válidas são somente para excluir ou recriar a ele por esse método. Observe, no entanto, que as mensagens não persistentes não podem ser recuperadas desta forma.

IBM i Pontos de verificação no IBM MQ for IBM i

Os pontos de verificação são feitos em momentos diferentes para fornecer um ponto de início consistente conhecido para recuperação.

O encadeamento do ponto de verificação dentro do processo AMQZMUC0 é responsável por tirar o ponto de verificação nos pontos a seguir:

- Inicialização do gerenciador de filas (STRMQM).
- Encerramento do gerenciador de filas (ENDMQM).
- Após um período de tempo decorrido desde que o último ponto de verificação (o período padrão é 30 minutos) e um número mínimo de registros de log foram gravados desde o ponto de verificação anterior (o valor padrão é 100).
- Após um número de registros de log foram gravados. O valor padrão é 10000.
- Depois que o tamanho limite diário tiver sido excedido e um novo receptor de diário foi criado automaticamente.
- Quando uma imagem de mídia integral é obtida com:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

IBM i Backups de dados do IBM MQ for IBM i

Use estas informações para entender os dois tipos de IBM MQ de backup para cada gerenciador de filas.

Para cada gerenciador de filas, há dois tipos de IBM MQ de backup a serem considerados:

- Data e backup diário.

Para assegurar-se de que ambos os conjuntos de dados são consistentes, faça isso somente depois de encerrar o gerenciador de filas.

- Backup diário.

É possível fazer isso enquanto o gerenciador de filas está ativo.

Para os dois métodos, você precisa para encontrar os nomes do diretório IFS gerenciador de filas e a biblioteca do gerenciador de filas. É possível localizar esses no arquivo de configuração (mq5.ini) do IBM MQ. Para obter mais informações, consulte [Alterando IBM MQ as informações de configuração em Multiplataformas](#)

Utilize os procedimentos a seguir para fazer ambos os tipos de backup:

Backup de dados e de diário de um gerenciador de filas específico

Nota: Não utilize uma solicitação salvar-enquanto-ativo quando o gerenciador de filas estiver em execução. Essa solicitação não pode ser concluída, a menos que todas as definições de consolidação com mudanças pendentes sejam consolidadas ou revertidas. Se este comando for utilizado quando o gerenciador de filas está ativo, as conexões de canal podem não terminar normalmente. Sempre use o procedimento a seguir.

1. Crie um receptor de diário vazio, utilizando o comando:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Use o comando **RCDMQMIMG** para registrar uma imagem MQM para todos os objetos IBM MQ e, em seguida, forçar um ponto de verificação usando o comando:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Termine os canais e assegure que o gerenciador de filas não esteja em execução. Se o seu gerenciador de filas estiver em execução, pare-o com o comando **ENDMQM**.
4. Faça backup da biblioteca do gerenciador de filas emitindo o seguinte comando:

```
SAVLIB LIB(QMTEST)
```

5. Faça backup dos diretórios IFS gerenciador de filas emitindo o seguinte comando:

```
SAV DEV(...) OBJ((' /QIBM/UserData/mqm/qmgrs/test'))
```

backup diário de um gerenciador de filas específico

Como todas as informações relevantes são mantidas nos diários, contanto que você execute um salvamento completo em algum momento, backups parciais podem ser executados salvando os receptores de diário. Esses registram todas as mudanças desde o momento do backup completo e são desempenhados emitindo os comandos a seguir:

1. Crie um receptor de diário vazio, utilizando o comando:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Use o comando **RCDMQMIMG** para registrar uma imagem MQM para todos os objetos IBM MQ e, em seguida, forçar um ponto de verificação usando o comando:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNTA(*YES) MQMNAME(TEST)
```

3. Salve os receptores de diário utilizando o comando:

```
SAVOBJ OBJ(AMQ*) LIB(QMTEST) OBJTYPE(*JRNRCV) .....
```

Uma estratégia de backup simples é desempenhar um backup completo do IBM MQ bibliotecas a cada semana e executar um backup diário. Isso, é claro, depende de como você configurou sua estratégia de backup para sua empresa.

Gerenciamento de diário no IBM i

Como parte de sua estratégia de backup, cuide de seus receptores de diário. É útil para remover os receptores de diário a partir do IBM MQ as bibliotecas por várias razões:

- Para liberar espaço; isso se aplica a todos os receptores de diário
- Para melhorar o desempenho ao iniciar (STRMQM)
- Para melhorar o desempenho de recriar objetos (RCRMQMOBJ)

Antes de excluir um receptor de diário, deve-se tomar cuidado para que você tenha uma cópia de backup e que você não precise mais do receptor de diário.

Os receptores de diário podem ser removidos da biblioteca do gerenciador de filas *após* terem sido desconectados dos diários e salvos, desde que eles estejam disponíveis para restauração se necessário para uma operação de recuperação.

O conceito de gerenciamento de diário é mostrado em [Figura 23 na página 415](#).

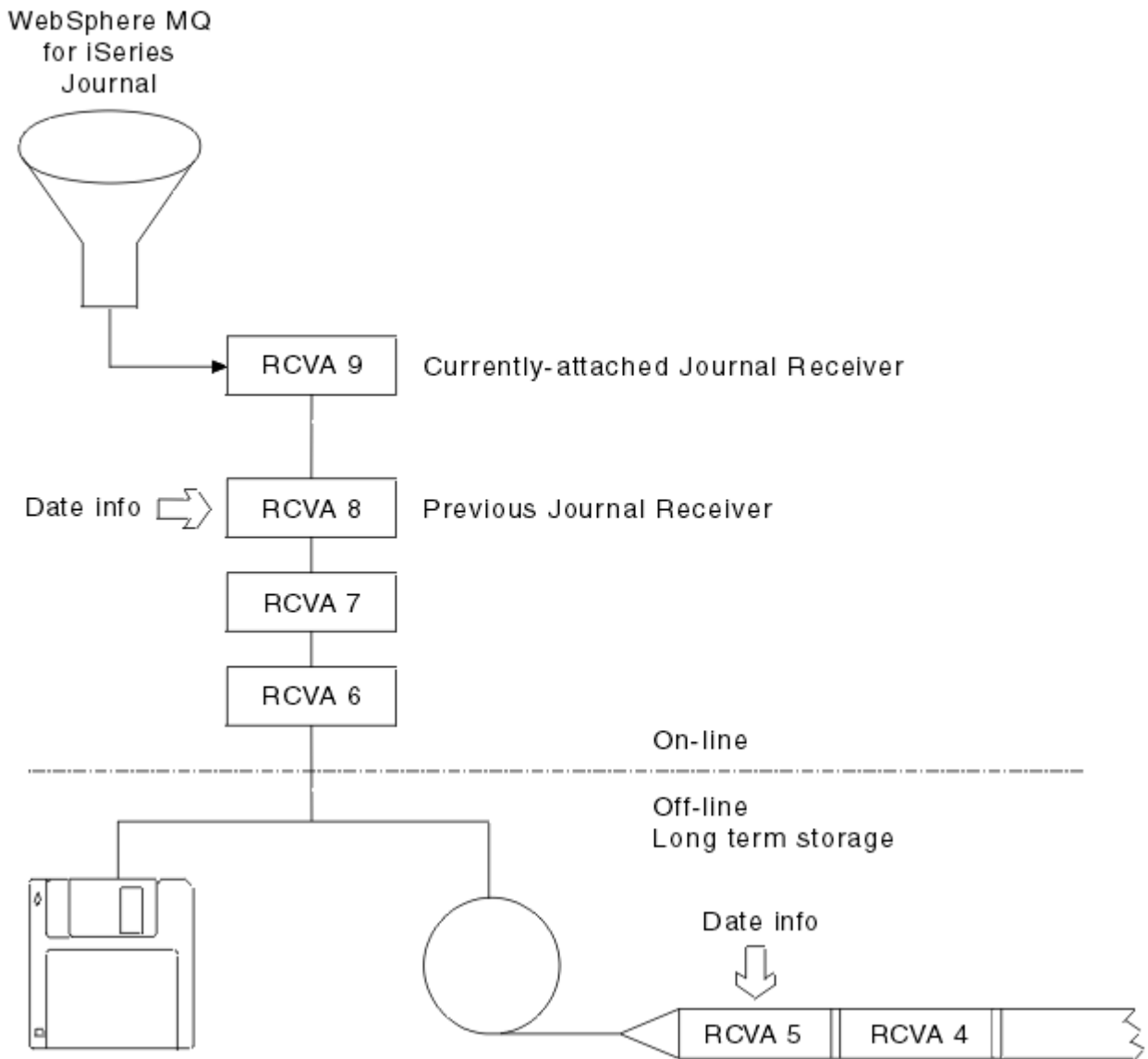


Figura 23. Registro no diário no IBM i

É importante saber até quanto voltar nos diários o IBM MQ tem probabilidade de ir, a fim de determinar quando um receptor de diário do qual foi feito backup pode ser removido da biblioteca do gerenciador de filas e quando o próprio backup pode ser descartado.

O IBM MQ emite duas mensagens para a fila de mensagens do gerenciador de filas (QMQMMSG na biblioteca do gerenciador de filas) para ajudar a determinar esse horário. Estas mensagens são emitidas quando ele é iniciado, quando ele muda um receptor de diário local e você utiliza RCDMQIMG para forçar um ponto de verificação. As duas mensagens são:

AMQ7460

ponto de recuperação de inicialização. Esta mensagem define a data e a hora da entrada de inicialização a partir do qual o IBM MQ reproduz o diário no caso de uma transmissão de recuperação de inicialização. Se o receptor de diário que contém esse registro está disponível no IBM MQ as bibliotecas, esta mensagem também contém o nome do receptor de diário que contém o registro.

AMQ7462

Entrada de recuperação de mídia mais antiga. Esta mensagem define a data e a hora da entrada mais antiga para utilizar para recriar um objeto a partir de sua imagem de mídia.

O receptor de diário identificado é o mais antigo necessário. Quaisquer outros receptores de diário com datas de criação mais antigas do IBM MQ não são mais necessários. Se somente estrelas são

exibidas, você precisará restaurar backups a partir da data indicada para determinar qual é o receptor de diário mais antigo.

Quando essas mensagens são registradas, o IBM MQ também grava um objeto de espaço do usuário para a biblioteca do gerenciador de filas que contém somente uma entrada: o nome do receptor de diário mais antigo que precisa ser mantido no sistema. Este espaço de usuário é chamado AMQJRNINF e os dados são gravados no formato:

```
JJJJJJJJJLLLLLLLLLLLLYYYYMMDDHHMMSSmmm
```

em que:

JJJJJJJJJ

É o nome do receptor mais antigo do qual o IBM MQ ainda precisa.

LLLLLLLLLLL

É o nome da biblioteca do receptor do diário.

YYYY

É o ano da entrada de diário mais antigo que o IBM MQ precisa.

MM

É o mês da entrada de diário mais antigo que o IBM MQ precisa.

DD

É o dia da entrada de diário mais antigo que o IBM MQ precisa.

HH

É a hora da entrada de diário mais antigo que o IBM MQ precisa.

SS

É os segundos da entrada de diário mais antigo que o IBM MQ precisa.

mmm

São os milissegundos da entrada de diário mais antigo que o IBM MQ precisa.

Quando o receptor de diário mais antigo foi excluído do sistema, este espaço do usuário contém asteriscos (*) para o nome do receptor de diário.

Nota: Executar RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) periodicamente pode economizar tempo de inicialização para IBM MQ e reduzir o número de receptores de diário locais que você precisa salvar e restaurar para recuperação.

O IBM MQ for IBM i não se refere aos receptores de diário a menos que esteja executando uma recuperação aprovada para inicialização ou para a recriação de um objeto. Se ele achar que um diário que ele requer não está presente, ele emitirá a mensagem AMQ7432 para a fila de mensagens do gerenciador de filas (QMQMMSG), relatando a hora e a data da entrada no diário que ele requer para concluir a aprovação da recuperação.

Se isso acontecer, restaure todos os receptores de diário que foram desconectados após esta data a partir do backup para permitir que a aprovação da recuperação seja bem-sucedida.

Mantenha o receptor de diário que contém a entrada de inicialização e quaisquer arquivos de diário subsequentes, disponíveis na biblioteca do gerenciador de filas.

Mantenha o receptor de diário contendo o Media Recovery Entry mais antigo e todos os receptores de diário subsequentes, disponíveis em todos os momentos e presentes na biblioteca do gerenciador de filas ou em backup.

Quando você forçar um ponto de verificação:

- Se o receptor de diário nomeado em AMQ7460 não for avançado, isso indica que há uma unidade de trabalho incompleta que precisa ser confirmada ou recuperada.
- Se o receptor de diário nomeado em AMQ7462 não for avançado, isto indica que existem um ou mais objetos danificados.

Restaurando um gerenciador de filas completo (dados e diários) no IBM i

Use estas informações para restaurar um ou mais gerenciadores de filas a partir de um backup ou de uma máquina remota.

Se você precisar recuperar um ou mais gerenciadores de filas do IBM MQ a partir de um backup, execute as seguintes etapas.

1. Coloque em modo quiesce os gerenciadores de filas do IBM MQ.
2. Localize o conjunto de backup mais recente, consistindo em seu mais recente backup completo e, subsequentemente, feito backup dos receptores de diário.
3. Execute uma operação RSTLIB, a partir do backup completo, para restaurar os dados de bibliotecas do IBM MQ para seu estado no momento do backup completo, emitindo os seguintes comandos:

```
RSTLIB LIB(QMQRLIB1) .....
RSTLIB LIB(QMQRLIB2) .....
```

Se um receptor de diário foi parcialmente salvo em um backup de diário e totalmente salvo em um backup subsequente, restaure somente o totalmente salvo. Restaurar diários individualmente, em ordem cronológica.

4. Execute uma operação de RST para restaurar os diretórios do IFS do IBM MQ para o sistema de arquivos IFS, utilizando o seguinte comando:

```
RST DEV(...) OBJ('/QIBM/UserData/mqm/qmgrs/testqm') ...
```

5. Inicie o gerenciador de filas de mensagens. Este reproduz todos diário registros gravados desde o backup completo e restaura todos os objetos do IBM MQ para o estado consistente no momento do backup diário.

Se você desejar restaurar um gerenciador de filas completo em uma máquina diferente, utilize o seguinte procedimento para restaurar tudo a partir da biblioteca do gerenciador de filas. (Usamos TEST como o nome do gerenciador de fila de amostra.)

1. CRTMQM TEST
2. DLTLIB LIB(QMTEST)
3. RSTLIB SAVLIB(QMTEST) DEV(*SAVF) SAVF(QMGRLIBSAV)
4. Exclua os seguintes arquivos do IFS:

```
/QIBM/UserData/mqm/qmgrs/TEST/QMQMCHKPT
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMQMOBJCAT
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMANAGER
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.AUTH.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CHANNEL.INITQ/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.COMMAND.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.REPOSITORY.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.TRANSMIT.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.PENDING.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.ADMIN.COMMAND.QUEUE/q
```

5. STRMQM TEST
6. RCRMQMOBJ OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(TEST)

Restaurando receptores de diário para um gerenciador de filas específico no IBM i

Use estas informações para entender as diferentes maneiras de restaurar receptores do diário.

A ação mais comum é para restaurar um receptor de diário de backup para uma biblioteca do gerenciador de filas, se um receptor que foi removido for necessário novamente para uma função de recuperação subsequente.

Essa é uma tarefa simples e requer que os receptores de diário a serem restaurados utilizando o padrão do IBM i RSTOBJ:

```
RSTOBJ OBJ(QMQMATA/AMQA000005) OBJTYPE(*JRNRCV) .....
```

Uma série de receptores de diário pode precisar ser restaurado, em vez de um único receptor. Por exemplo, AMQA000007 é o mais antigo receptor nas bibliotecas IBM MQ e ambos AMQA000005 e AMQA000006 precisam ser restaurados.

Neste caso, restaure os receptores individualmente por ordem cronológica reversa. Isso não é sempre necessário, mas é uma boa prática. Em situações graves, pode ser necessário utilizar o comando do IBM i WRKJRNA para associar os receptores de diário restaurados com o diário.

Ao restaurar os diários, o sistema automaticamente cria um receptor de diário conectado com um novo nome na sequência do receptor de diário. No entanto, o novo nome gerado pode ser o mesmo que um receptor de diário que você precisa restaurar. A intervenção manual é necessária para superar esse problema; para criar um novo receptor de diário em sequência e novo diário antes de restaurar o receptor de diário.

Por exemplo, considere o problema com diário salvo AMQAJRN e os seguintes receptores de diário:

- AMQA000000
- AMQA100000
- AMQA200000
- AMQA300000
- AMQA400000
- AMQA500000
- AMQA600000
- AMQA700000
- AMQA800000
- AMQA900000

Ao restaurar diário AMQAJRN para uma biblioteca do gerenciador de filas, o sistema cria automaticamente AMQA000000 receptor de diário. Esse conflito com o receptor gerado automaticamente um dos receptores de diário existentes (AMQA000000) que deseja restaurar, que você não pode restaurar.

A solução é:

1. Crie manualmente o receptor de diário seguinte (consulte [“Diários do gerenciador de filas no IBM i”](#) na página 409):

```
CRTJRNRCV JRNRCV(QMQRLIB/AMQA900001) THRESHOLD(XXXXX)
```

2. Crie manualmente o diário com o receptor de diário:

```
CRTJRN JRN(QMGRLIB/AMQAJRN) MNGRCV(*SYSTEM) +  
JRNRCV(QMGRLIB/AMQA900001) MSGQ(QMGRLIB/AMQAJRNMSG)
```

3. Restaure o AMQA000000 os receptores de diário local para AMQA900000.

Gerenciadores de filas de várias instâncias no IBM i

Os gerenciadores de filas de multi-instâncias melhoram a disponibilidade, comutando automaticamente para um servidor em espera se o servidor ativo falhar. Os servidores ativo e em espera são multi-instâncias do mesmo gerenciador de filas; eles compartilham os mesmos dados do gerenciador de filas. Se a instância ativa falhar, você precisa transferir seu diário para o espera que assume que o gerenciador de filas pode recriar suas filas.

Configure os sistemas IBM i nos quais você estiver executando gerenciadores de filas de várias instâncias para que, se a instância ativa do gerenciador de filas falhar, o diário usado esteja disponível para a instância em espera que assume. É possível projetar suas próprias tarefas de configuração e administração para tornar o diário da instância ativa disponível para a instância que assumir. Se você não deseja perder mensagens, seu design deve assegurar que o diário standby é consistente com o diário ativo no ponto de falha. É possível adaptar seu design a partir de uma das duas configurações que são descritas com exemplos nos tópicos subsequentes que mantêm a consistência.

1. Espelhe o diário do sistema que está executando a instância ativa do gerenciador de filas para os sistemas que estão executando as instâncias em espera.
2. Coloque o diário em um IASP (Independent Auxiliary Storage Pool) que é transferível do sistema que está executando a instância ativa para uma instância em espera.

A primeira solução não exige hardware ou software adicionais como ele utiliza ASPs básicos. A segunda solução requer IASPs alternável que precisa do IBM i em cluster suportam que está disponível como um vendido separadamente do IBM i License Produto 5761-SS1 Opção 41.

Confiabilidade e disponibilidade no IBM i

Gerenciadores de Filas de Várias Instâncias visam melhorar a disponibilidade dos aplicativos. As restrições tecnológicas e físicas significam que você precisa de diferentes soluções para atender às demandas de recuperação de desastre, fazendo backup de gerenciadores de filas e operação contínua.

No comércio que você configurar para a confiabilidade e a disponibilidade de um grande número de fatores, resultando em quatro pontos de design distintos:

Recuperação de Desastre

Otimizado para recuperação após um desastre maior que destroem todos os locais ativos.

recuperação de desastre com espelhamento geográfico é geralmente baseada no IBM i do IASP.

Backup

Otimizado para a recuperação após uma falha localizada, geralmente um erro humano ou algum problema técnico inesperado.

IBM MQ fornece gerenciadores de filas de backup para fazer backup os gerenciadores de filas periodicamente. Também é possível utilizar replicação assíncrona de diários do gerenciador de filas para melhorar a moeda do backup.

Disponibilidade

Otimizado para restaurar as operações rapidamente fornecendo a aparência de um serviço ininterrupto quase previsível falhas técnicas a seguir como uma falha do servidor ou do disco.

A recuperação geralmente é medida em minutos, com a detecção, às vezes, demorando mais do que o processo de recuperação. Um gerenciador de filas de várias instâncias ajuda você a configurar para *disponibilidade*.

Operação Contínua

Otimizado para fornecer um serviço ininterrupto.

As soluções de operação contínua têm que resolver o problema de detecção e quase sempre envolvem submeter o mesmo trabalho por meio de mais de um sistema e utilizar o primeiro resultado ou se a exatidão for uma consideração importante, comparar pelo menos dois resultados.

Um gerenciador de filas de várias instâncias ajuda você a configurar para *disponibilidade*. Uma instância do gerenciador de filas está ativo por vez. Alternar para uma instância em espera leva pouco mais de

dez segundos até quinze minutos ou mais, dependendo de como o sistema está configurado, carregado e ajustado.

Um gerenciador de filas de várias instâncias pode dar a aparência de um serviço quase ininterrupto se usado com IBM MQ MQI clients, reconectável, que pode continuar o processamento sem que o programa aplicativo esteja necessariamente ciente de uma indisponibilidade do gerenciador de filas; veja o tópico [Reconexão automatizada de cliente](#).

Componentes de uma solução de alta disponibilidade no IBM i

Construa uma solução de alta disponibilidade utilizando gerenciadores de filas de várias instâncias, fornecendo armazenamento em rede robusto para dados do gerenciador de filas, diário ou diários IASP armazenamento robusto para replicação do gerenciador de filas e o utilizando os clientes reconectáveis, de aplicativos configurados como serviços do gerenciador de filas reiniciável.

Um gerenciador de filas de várias instâncias reage à detecção de falha do gerenciador de filas, retomando a inicialização de outra instância do gerenciador de filas em outro servidor. Para concluir sua inicialização, a instância precisa de acesso aos dados do gerenciador de filas compartilhadas em armazenamento em rede e a sua cópia do diário do gerenciador de filas locais.

Para criar uma solução de alta disponibilidade, você precisa gerenciar a disponibilidade dos dados do gerenciador de fila, a moeda do diário do gerenciador de filas locais e qualquer construção reconectáveis aplicativos cliente ou implementar seus aplicativos como serviços do gerenciador de filas a reiniciar automaticamente quando o gerenciador de filas é retomado. A reconexão do cliente automática não é suportada pelo IBM MQ classes for Java.

Dados do gerenciador de filas

Coloque dados do gerenciador de filas no armazenamento em rede que seja compartilhado, altamente disponível e confiável, possivelmente utilizando os discos RAID de nível 1 ou superior. O sistema de arquivos precisa para atender os requisitos para um sistema de arquivo compartilhado para gerenciadores de filas de várias instâncias; para obter informações adicionais sobre os requisitos para sistemas de arquivos compartilhados, consulte [Requisitos para sistemas de arquivos compartilhados](#). O Network File System 4 (NFS4) é um protocolo que atende a esses requisitos.

Diários do gerenciador de filas

Você também precisa configurar os diários do IBM i usados pelas instâncias do gerenciador de filas para que a instância em espera seja capaz de restaurar seus dados do gerenciador de filas para um estado consistente. Para o serviço ininterrupto, isso significa que deve-se restaurar os diários para seu estado quando a instância ativa falhou. Ao contrário de soluções de recuperação de desastre ou de backup, restaurando diários para um ponto de verificação anterior não é suficiente.

Não é possível compartilhar fisicamente diários entre vários sistemas IBM i no armazenamento em rede. Para restaurar diários do gerenciador de filas para o estado consistente no ponto de falha, será necessário transferir o diário físico que era local para a instância ativa do gerenciador de filas no momento da falha para a nova instância que foi ativada ou manter espelhos do diário em execução em instâncias de espera. O diário espelhado é uma réplica de diário remoto que foi mantida exatamente em sincronia com o diário local pertencente à instância que falhou.

Três configurações são pontos de partida para projetar como gerenciar os diários para um gerenciador de filas de várias instâncias,


1. Usando de replicação sincronizada diário (espelhamento diário) da instância ativa ASP, para as instâncias em espera ASPs.
2. Transferindo um IASP que você configurou para conter o diário do gerenciador de filas a partir da instância ativa para a instância em espera que está assumindo como a instância ativa.
3. Usando secundário IASP espelhos sincronizados.

Consulte as opções de [ASP](#) para obter mais informações sobre como colocar dados do gerenciador de filas em um iASP, no comando CRTMQM do IBM MQ IBM i.

Consulte também [Alta disponibilidade](#) nas informações do IBM i em IBM Documentation.

Aplicativos

Para construir um cliente para reconectar automaticamente ao gerenciador de filas quando o gerenciador de filas de espera for retomado, conecte seu aplicativo para o gerenciador de filas usando MQCONNX e especifique MQCNO_RECONNECT_Q_MGR no campo **MQCNO** Opções. Consulte [de Alta Disponibilidade programas de amostra para três programas de amostra utilizando os clientes reconectáveis](#) e [Recuperação de Aplicativo](#) para obter informações sobre como projetar aplicativos cliente para recuperação.

 *Criando um compartilhamento de rede para os dados do gerenciador de filas usando o NetServer no IBM i*

Criar um compartilhamento de rede em um servidor IBM i para armazenar dados do gerenciador de filas. Configure conexões a partir de dois servidores, que vão para o host instâncias do gerenciador de filas, para acessar o compartilhamento de rede.

Antes de começar

- Você requer três servidores IBM i para esta tarefa. O compartilhamento de rede está definido em um dos servidores, GAMMA. Os outros dois servidores, ALPHA e BETA, são para se conectar a GAMMA.
- Instale o IBM MQ nos três servidores.
- Instale o System i Navigator. Consulte [System i Navegador](#).

Sobre esta tarefa

- Crie o diretório do gerenciador de filas em GAMMA e configure a propriedade e as permissões corretas para os perfis de usuário QMQM e QMQMADM. O diretório e permissão são facilmente criadas pela instalação do IBM MQ em GAMMA.
- Use o System i Navigator para criar um compartilhamento para o diretório de dados do gerenciador de filas em GAMMA.
- Crie diretórios em ALPHA e BETA que aponte para o compartilhamento.

Procedimento

1. No GAMMA, crie o diretório para hospedar os dados do gerenciador de filas com o perfil de usuário do QMQM como o proprietário e QMQMADM como o grupo principal.

Sugestão:

Uma maneira rápida e confiável de criar o diretório com as permissões corretas é instalar o IBM MQ em GAMMA.

Posteriormente, se você não deseja executar o IBM MQ em GAMMA, desinstale IBM MQ. Após a desinstalação, o diretório permanece no /QIBM/UserData/mqm/qmgrs GAMMA com o proprietário do perfil de usuário do QMQM e QMQMADM o grupo principal.

A tarefa usa o diretório /QIBM/UserData/mqm/qmgrs no GAMMA para o compartilhamento.

2. Inicie o System i Navigator assistente **Incluir conexão** e se conectar ao sistema GAMMA.
 - a) Dê um clique duplo no ícone **System i Navigator** na área de trabalho do seu Windows.
 - b) Clique em **Sim** para criar uma conexão.
 - c) Siga as instruções no assistente **Incluir Conexão** e criar uma conexão do sistema para IBM i GAMMA.

A conexão com GAMMA é incluída em **Meu Conexões**.

3. Inclua um novo compartilhamento de arquivo em GAMMA.

- a) Na janela **System i Navigator**, clique na pasta File Shares em My Connections/GAMMA/File Systems.
 - b) Na janela **Minhas tarefas**, clique em **Gerenciar IBM i Compartilhamentos do NetServer**.
Uma nova janela, **IBM i NetServer - GAMMA**, se abre na sua área de trabalho e mostra objetos compartilhados.
 - c) Clique com o botão direito na pasta Shared Objects > **Arquivo** > **Novo** > **Arquivo**.
Uma nova janela, **IBM i NetServer File Share - GAMMA**, se abre.
 - d) Dê ao compartilharem um nome, WMQ, por exemplo.
 - e) Configure o controle de acesso para Read/Write
 - f) Selecione o **Nome do caminho** navegando para o diretório /QIBM/UserData/mqm/qmgrs criado anteriormente, e clique em **OK**.
A janela **IBM i NetServer File Share - GAMMA** fecha, e WMQ é listado na janela de objetos compartilhados.
4. Clique com o botão direito em **WMQ** na janela de objetos compartilhados. Clique em **Arquivo** > **Permissões**.

Uma janela é aberta, **Permissões Qmgrs - GAMMA**, para o objeto /QIBM/UserData/mqm/qmgrs.

- a) Verifique as seguintes permissões para QMQM, se eles ainda não estiverem configurados:
 - Read
 - Write
 - Execute
 - Management
 - Existence
 - Alter
 - Reference
 - b) Verifique as seguintes permissões para QMQMADM, se eles ainda não estiverem configurados:
 - Read
 - Write
 - Execute
 - Reference
 - c) Inclua outros perfis de usuários que você deseja dar permissões para /QIBM/UserData/mqm/qmgrs.
Por exemplo, você pode fornecer ao perfil do usuário padrão (Público) permissões Read e Execute para /QIBM/UserData/mqm/qmgrs.
5. Verifique se todos os perfis de usuário aos quais é concedido acesso ao /QIBM/UserData/mqm/qmgrs no GAMMA têm a mesma senha dos servidores que acessam o GAMMA.
- Em particular, assegure-se de que o QMQM perfis de usuário em outros servidores, que vão para acessar o compartilhamento, possuem a mesma senha que o perfil do usuário no QMQM GAMMA.
- Sugestão:** Clique na pasta My Connections/GAMMA/Users and Groups no System i Navigator para configurar as senhas. Alternativamente, use os comandos **CHFUSRPRF** e **CHGPWD**.

Resultados

Verifique se é possível acessar GAMMA a partir de outros servidores utilizando o compartilhamento. Se você estiver fazendo as outras tarefas, verifique se pode acessar o GAMMA do ALPHA e do BETA usando o caminho /QNTC/GAMMA/WMQ. Se o diretório /QNTC/GAMMA não existir no ALPHA ou no BETA então você

deve criar o diretório. Dependendo do domínio NetServer, é possível ter de IPL ALPHA ou BETA antes de criar o diretório.

```
CRTDIR DIR('/QNTC/GAMMA')
```

Após verificar isso, você terá acesso a /QNTC/GAMMA/WMQ de ALPHA ou BETA, emitir o comando, CRTMQM MQMNAME('QM1') MQMDIRP('/QNTC/GAMMA/WMQ') cria /QIBM/UserData/mqm/qmgrs/QM1 em GAMMA.

Como proceder a seguir

Crie um gerenciador de filas de várias instâncias seguindo as etapas em qualquer uma das tarefas, “Criando um gerenciador de filas de várias instâncias usando o espelhamento de diário e o NetServer no IBM i” na página 434 ou “Convertendo um gerenciador de filas de instância única para um gerenciador de filas de várias instâncias usando NetServer e espelhamento diário no IBM i” na página 438.

IBM i **Desempenho de failover no IBM i**

O tempo que leva para detectar uma instância do gerenciador de filas falhou e, então, para retomar o processamento em uma espera pode variar entre dezenas de segundos até quinze minutos ou mais dependendo da configuração. Desempenho precisa ser uma importante consideração de projetar e testar uma solução de alta disponibilidade.

Há vantagens e desvantagens para ponderar de decidir se deve configurar um gerenciador de filas de várias instâncias para utilizar replicação de diário ou utilizar um IASP. O espelhamento requer que o gerenciador de filas grave assincronicamente em um diário remoto. Do ponto de vista de hardware, esse não precisa afetar o desempenho, mas a partir de uma perspectiva de software há um comprimento maior envolvido na gravação em um diário remoto do que somente a um diário local e isso pode ser esperado para reduzir o desempenho de um gerenciador de filas em execução até certo ponto. No entanto, quando o gerenciador de filas de espera assume, o atraso na sincronização de seu diário local a partir do diário remoto mantidos pela instância ativa antes de falhar, geralmente é pequeno em comparação com o tempo que leva para o IBM i para detectar e transferir o IASP para o servidor que está executando a instância de espera do gerenciador de filas. Os tempos de transferência de IASP podem ser de dez a quinze minutos em vez de serem concluídos em segundos. O tempo de transferência IASP depende do número de objetos que precisam ser *desativado no* quando o IASP é transferido para o sistema em espera e o tamanho dos caminhos de acesso ou índices, que precisam ser mesclados.

Quando o gerenciador de filas de espera assumir, o atraso na sincronização de seu diário local do diário remoto, mantido pela instância ativa antes de ele falhar, geralmente será pequeno em comparação com o tempo que leva para o IBM i detectar e transferir o ASP independente para o servidor executando a instância em espera do gerenciador de filas. ASP tempos de transferência pode ser até dez a quinze minutos em vez de ser concluída em segundos. O tempo de transferência ASP independente depende do número de objetos que precisam ser *desativado em* quando o ASP independente é transferido para o sistema em espera e o tamanho dos caminhos de acesso ou índices, que precisam ser mescladas.

No entanto, transferir o diário não é o único fator que influencia o tempo gasto para a instância em espera para retomar totalmente. Você também precisa considerar o tempo que leva para o sistema de arquivos de rede para liberar o bloqueio nos dados do gerenciador de filas que assinala para a instância em espera para tentar continuar com sua inicialização e também o tempo que leva para recuperar filas do diário para que a instância é capaz de iniciar o processamento de mensagens novamente. Estas outras fontes de atraso todos incluir ao tempo que leva para iniciar uma instância em espera. O tempo total para a comutação consiste nos componentes a seguir,

Tempo de detecção de falha

O tempo que leva para NFS para liberar o bloqueio nos dados do gerenciador de filas e a instância em espera para continuar seu processo de inicialização.

Tempo de transferência

No caso de um cluster HA, o tempo gasto para IBM i para transferir o IASP do sistema que hospeda a instância ativa para a instância em espera e, no caso de replicação de diário, o tempo que leva para atualizar o diário local em espera com os dados da réplica remota.

Tempo de reinicialização

O tempo que leva para o gerenciador de filas recém-criado instância ativa para reconstruir suas filas do último ponto de verificação em seu diário restaurados e retomar o processamento de mensagens.

Nota:

Se a instância em espera que assumiu está configurada para replicação síncrona para a instância anteriormente ativa, a inicialização poderia ser atrasada. A nova instância ativada pode ser incapaz de replicar para seu diário remoto, se o diário remoto estiver no servidor que hospedou a instância previamente ativa e o servidor falhou.

O tempo padrão para aguardar por uma resposta síncrona é um minuto. É possível configurar o atraso máximo antes que o tempo limite de replicação. Alternativamente, é possível configurar instâncias em espera para iniciar utilizando replicação assíncrona para a instância ativa que falhou. Posteriormente você alternar a replicação síncrona para, quando a instância com falha está em execução em espera novamente. A mesma consideração se aplica ao uso de espelhos do ASP independentes síncronos.

É possível fazer medidas de nível base separado para esses componentes para ajudar a avaliar o tempo geral para failover e fator em sua decisão que a abordagem de configuração a ser utilizado. Ao apresentar a melhor decisão de configuração, também é necessário considerar como outros aplicativos no mesmo servidor irão efetuar failover e se há processos de backup ou recuperação de desastre que já utilizam IASP.

Tempos de transferência de IASP podem ser reduzidos por meio de ajuste de sua configuração de cluster:

1. Perfis de usuários nos sistemas no cluster devem ter os mesmos GID e UID para eliminar a necessidade do processo de ativação para mudar UIDs e GIDs.
2. Minimizar o número de objetos de banco de dados no sistema e os conjuntos de discos básicos, como esses precisam ser mescladas para criar a tabela de referência cruzada para o grupo de discos do conjunto.
3. Dicas de desempenho adicionais podem ser encontradas no IBM Redbook, *Implementando o PowerHA for IBM i*, SG24-7405.

Uma configuração utilizando os ASPs básicos, espelhamento de diário e uma configuração pequena deve comutar na ordem de dezenas de segundos.

Visão geral da combinação de recursos de clusterização do IBM i com o armazenamento em cluster do IBM MQ

A execução do IBM MQ no IBM i e a exploração dos recursos de clusterização do IBM i podem fornecer uma solução mais abrangente de alta disponibilidade do que usar apenas o armazenamento em cluster do IBM MQ.

Ter esse recurso, você precisa configurar:

1. Clusters na máquina do IBM i; consulte [“Clusters do IBM i”](#) na página 424
2. Um conjunto de armazenamentos auxiliares independentes (IASP), para o qual mover o gerenciador de filas; consulte [“Conjuntos de armazenamentos auxiliares independentes \(IASPs\)”](#) na página 425
3. Um grupo de recursos de cluster (CRG), consulte [“Grupos de recursos de cluster de dispositivo”](#) na página 425, no qual você define:
 - Domínio de Recuperação
 - IASP
 - Programa de saída; consulte [“Programa de saída do CRG do dispositivo”](#) na página 425

Clusters do IBM i

Um cluster do IBM i é uma coleção de instâncias, isto é, computadores ou partições do IBM i, que estão logicamente vinculados.

O propósito desse agrupamento é permitir o backup de cada instância, eliminando um ponto único de falha e aumentando a resiliência de dados e aplicativos. Com um cluster criado, os vários tipos de grupos

de recursos de cluster (CRG) podem ser configurados para gerenciar aplicativos, dados e dispositivos no cluster.

Consulte [Criando um cluster](#) e o comando [Create Cluster \(CRTCLU\)](#) para obter informações adicionais.

Conjuntos de armazenamentos auxiliares independentes (IASPs)

Um IASP é um tipo de ASP do usuário que serve de extensão de armazenamento de nível único. É uma parte do armazenamento que, devido à sua independência do armazenamento do sistema, pode ser facilmente manipulada sem precisar do IPL do sistema.

Um IASP pode ser facilmente alternado para outra instância do sistema operacional ou replicado para um IASP de destino em outra instância do sistema operacional. Dois métodos podem ser usados para alternar um IASP entre instâncias:

- O primeiro método requer que todos os computadores no cluster, assim como a torre de disco alternável que contém o IASP, sejam conectados usando um loop de link de alta velocidade (HSL).
- O segundo método requer que as instâncias do sistema operacional sejam partições no mesmo computador do IBM i em que os processadores de entrada/saída (IOPs) podem ser alternados entre as partições. Nenhum hardware especial é necessário para poder replicar um IASP. A replicação é executada usando TCP/IP sobre a rede.

Consulte o comando [Configure Device ASP \(CFGDEVASP\)](#) para obter mais informações.

Grupos de recursos de cluster de dispositivo

Há vários tipos de grupos de recursos de cluster (CRGs). Para obter mais informações sobre os diferentes tipos de CRGs disponíveis, consulte [Grupo de recursos de cluster](#).

Este tópico concentra-se em um CRG de dispositivo. Um CRG de dispositivo:

- Descreve e gerencia recursos do dispositivo, como conjuntos de armazenamentos auxiliares independentes (IASPs).
- Define o domínio de recuperação dos nós do cluster
- Designa um dispositivo e
- Designa o programa de saída que manipulará eventos de cluster.

O domínio de recuperação indica qual nó do cluster será considerado como o nó primário. O restante dos nós é considerado backups. Os nós de backup também são ordenados no domínio de recuperação, especificando qual nó é o primeiro backup, o segundo backup e assim por diante, dependendo de quantos nós existem no domínio de recuperação.

No caso de uma falha do nó primário, o programa de saída é executado em todos os nós no domínio de recuperação. O programa de saída em execução no primeiro backup pode então fazer as inicializações necessárias para tornar esse nó o novo nó primário.

Consulte [Criando CRGs de dispositivo](#) e o comando [Create Cluster Resource Group \(CRTCRG\)](#) para obter mais informações.

Programa de saída do CRG do dispositivo

O serviço de recurso de cluster do sistema operacional chama um programa de saída do CRG do dispositivo quando ocorre um evento em um dos nós que o domínio de recuperação define; por exemplo, um evento de comutação ou failover.

Um evento de failover ocorre quando o nó primário do cluster falha e os CRGs são alternados com todos os recursos que eles gerenciam, enquanto um evento de comutação ocorre quando um CRG específico é alternado manualmente do nó primário para o nó de backup.

De qualquer maneira, o programa de saída é responsável por inicializar e iniciar todos os programas que estavam em execução no nó primário anterior, que converte o primeiro nó de backup no novo nó primário.

Por exemplo, com o IBM MQ, o programa de saída deve ser responsável por iniciar o subsistema (QMQM) e os gerenciadores de filas do IBM MQ. Os gerenciadores de filas devem ser configurados para iniciar automaticamente listeners e serviços, como monitores acionadores.

Um programa de saída de amostra, AMQSCRG4 está disponível no IBM i

Configuração IASP alternável

O IBM MQ pode ser configurado para aproveitar os recursos de clusterização do IBM i. Para fazer isso:

1. Crie um cluster do IBM i entre os sistemas do data center
2. Mova o gerenciador de filas para um IASP.

“Movendo ou removendo um gerenciador de filas para/de um conjunto de armazenamentos auxiliares independentes” na página 427 contém um código de amostra para ajudar a realizar essa operação.


3. É necessário criar um CRG definindo o domínio de recuperação, o IASP e o programa de saída.

“Configurando um grupo de recursos de cluster do dispositivo” na página 426 contém um código de amostra para ajudar a realizar essa operação.

Conceitos relacionados

“ASPs independentes e alta disponibilidade” na página 447

ASPs independentes permitem que os aplicativos e os dados a serem movidos entre servidores. A flexibilidade de ASPs independentes significa que eles são a base para alguns IBM i soluções de alta disponibilidade. Ao ponderar se utilizar um ASP ou ASP independente para o diário do gerenciador de filas, é necessário considerar a configuração de alta disponibilidade no outro baseado em ASPs independentes.

 *Configurando um grupo de recursos de cluster do dispositivo*

Um programa de exemplo para configurar um grupo de recursos de cluster (CRG) do dispositivo.

Sobre esta tarefa

No exemplo a seguir, observe que:

- [PRIMARY SITE NAME] e [BACKUP SITE NAME] poderiam ser duas sequências distintas de oito caracteres.
- [PRIMARY IP] e [BACKUP IP] são os IPs a serem usados para espelhamento.

Procedimento

1. Identifique o nome do cluster.
2. Identifique o nome e a biblioteca do programa de saída de CRG.
3. Determine o nome do nó primário e os nós de backup a serem definidos por esse CRG.
4. Identifique o IASP a ser gerenciado por esse CRG e certifique-se de que ele tenha sido criado sob o nó primário.
5. Crie uma descrição do dispositivo nos nós de backup usando o comando:

```
CRTDEVASP DEVD([IASP NAME]) RSRNAME([IASP NAME])
```

6. Inclua o endereço IP de controle em todos os nós usando o comando:

```
ADDCPIFC INTNETADR(' [TAKEOVER IP]') LIND([LINE DESC])  
SUBNETMASK(' [SUBNET MASK]') AUTOSTART(*NO)
```

7. Inicie o endereço IP de controle apenas no nó primário usando o comando:

```
STRTCPIFC INTNETADR(' [TAKEOVER IP]')
```

8. Opcional: Se o IASP for comutável, chame este comando:

```
CRTCRCG CLUSTER([CLUSTER NAME]) CRG( [CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT NAME])
USRPRF([EXIT PROFILE]) RCYDMN(( [PRIMARY NODE] *PRIMARY) ([BACKUP NAME] *BACKUP))
EXITPGMFMT(EXTP0200) CFGOBJ([IAPS NAME] *DEVD *ONLINE '[TAKEOVER IP]')
```

9. Opcional: Se o IASP deve ser espelhado, chame este comando:

```
CRTCRCG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT NAME])
USRPRF([EXIT PROFILE]) RCYDMN(( [PRIMARY NODE] *PRIMARY *LAST [PRIMARY SITE NAME] ('[PRIMARY IP]'))
[BACKUP NAME] *BACKUP *LAST [BACKUP SITE NAME] ('[BACKUP IP]')) EXITPGMFMT(EXTP0200)
CFGOBJ([IAPS NAME] *DEVD *ONLINE '[TAKEOVER IP]')
```



Movendo ou removendo um gerenciador de filas para/de um conjunto de armazenamentos auxiliares independentes

Um programa de exemplo para mover um gerenciador de filas para um conjunto de armazenamentos auxiliares independentes (IASP) e comandos para remover um gerenciador de filas de um IASP.

Sobre esta tarefa

No exemplo a seguir, observe que:

- [MANAGER NAME] é o nome do seu gerenciador de filas.
- [IASP NAME] é o nome do IASP.
- [MANAGER LIBRARY] é o nome da biblioteca de gerenciadores de filas.
- [MANAGER DIRECTORY] é o nome do diretório de gerenciadores de filas.

Procedimento

1. Identifique o nó primário e os nós de backup.
2. Realize o procedimento a seguir no nó primário:
 - a) Certifique-se de que o gerenciador de filas tenha sido terminado.
 - b) Certifique-se de que o IASP seja vary on usando o comando

```
VRYCFG CFGOBJ([IASP NAME]) CFGTYPE(*DEV) STATUS(*ON)
```

- c) Crie o diretório de gerenciadores de filas no IASP.
Haverá um diretório sob raiz com o nome do IASP, que é:

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

- d) Mova os objetos IFS do gerenciador de filas para o diretório de gerenciadores de filas que você acabou de criar no IASP usando o comando a seguir:

```
QSH CMD('mv /QIBM/UserData/mqm/qmgrs/[MANAGER NAME]
/[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

- e) Crie um arquivo de salvamento temporário chamado MGRLIB usando o comando:

```
CRTSAVF QGPL/MGRLIB
```

- f) Salve sua biblioteca de gerenciadores de filas no arquivo de salvamento MGRLIB, usando o comando a seguir:

```
SAVLIB LIB([MANGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)
```

- g) Exclua a biblioteca de gerenciadores de filas usando o comando a seguir e ignore todas as mensagens de consulta:

```
DLTLIB [MANAGER LIBRARY]
```

- h) Restaure a biblioteca de gerenciadores de filas no IASP usando o comando a seguir:

```
RSTLIB SAVLIB([MANAGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)  
RSTASPDEV([IASP NAME])
```

- i) Exclua o arquivo de salvamento temporário usando o comando a seguir:

```
DLTF FILE(QGPL/MGRLIB)
```

- j) Crie um link simbólico para os objetos IFS do gerenciador de filas no IASP, usando o comando a seguir:

```
ADDLNK OBJ('/[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

- k) Conecte ao IASP usando o comando a seguir:

```
SETASPGRP [IASP NAME]
```

- l) Inicie o gerenciador de filas usando o comando:

```
STRMQM [MANAGER NAME]
```

3. Realize o procedimento a seguir nos nós de backup:

- a) Crie um diretório de gerenciadores de filas temporário usando o comando a seguir:

```
QSH CMD('mkdir -p /[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

- b) Crie um link simbólico para o diretório de gerenciadores de filas temporário usando o comando a seguir:

```
ADDLNK OBJ('/[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

- c) Exclua o diretório temporário usando o comando a seguir:

```
QSH CMD('rm -r /[IASP NAME]')
```

- d) Inclua o seguinte no final do arquivo /QIBM/UserData/mqm/mqs.ini:

```
QueueManager:  
Name=[MANAGER NAME]  
Prefix=/QIBM/UserData/mqm  
Library=[MANAGER LIBRARY]  
Directory=[MANAGER DIRECTORY]
```

4. Para remover um gerenciador de filas de um IASP, emita os comandos a seguir:

- VRYCFG CFGOBJ([IASP NAME]) CFGTYPE(*DEV) STATUS(*ON)
- SETASPGRP [IASP NAME]
- ENDMQM [MANAGER NAME]
- DLTMQM [MANAGER NAME]

Configuração de diário espelhado para ASP no IBM i

Configure um gerenciador de filas de várias instâncias robusto usando replicação síncrona entre diários espelhados.

Uma configuração de gerenciador de filas espelhado usa diários criados em conjuntos de armazenamento auxiliares (ASP) independentes ou básico.s

No IBM i, os dados do gerenciador de filas são gravados em diários e em um sistema de arquivos. Diários contêm a cópia principal dos dados do gerenciador de filas. Diários são compartilhados entre sistemas usando a replicação de diário síncrona ou assíncrona. Uma mistura de diários locais e remotos é necessária para reiniciar uma instância do gerenciador de filas. A reinicialização do gerenciador de filas lê os registros de diário a partir da mistura de diários locais e remotos no servidor e os dados do gerenciador de filas no sistema de arquivos de rede compartilhada. Os dados no sistema de arquivos se apressam em reiniciar o gerenciador de filas. Pontos de verificação são armazenados no sistema de arquivos, marcando pontos de sincronização entre o sistema de arquivos e os diários. Registros de diário armazenados antes do ponto de verificação não são necessários para que o gerenciador de filas típico seja reiniciado. No entanto, os dados no sistema de arquivos podem não estar atualizados e registros de diário após o ponto de verificação são usados para concluir a reinicialização do gerenciador de filas. Os dados nos diários anexados à instância são mantidos atualizados para que a reinicialização possa ser concluída com êxito.

No entanto, mesmo os registros de diário podem não estar atualizados, se o diário remoto no servidor de espera estava sendo replicado assincronicamente e a falha ocorreu antes de ele ter sido sincronizado. Caso você decida reiniciar um gerenciador de filas usando um diário remoto que não esteja sincronizado, a instância do gerenciador de filas de espera pode reprocessar as mensagens excluídas antes de a instância ativa falhar ou não processar as mensagens recebidas antes de a instância ativa falhar.

Outra possibilidade rara é o sistema de arquivos conter o registro de ponto de verificação mais recente e um diário remoto não sincronizado na espera não conter. Nesse caso, o gerenciador de filas não é reiniciado automaticamente. Você tem a opção de esperar até que o diário remoto seja sincronizado ou fazer cold start do gerenciador de filas em espera a partir do sistema de arquivos. Embora, nesse caso, o sistema de arquivos contenha um ponto de verificação mais recente dos dados do gerenciador de filas que o diário remoto, ele pode não conter todas as mensagens processadas antes de a instância ativa falhar. Algumas mensagens podem ser reprocessadas e algumas não processadas após um cold restart que não esteja sincronizado com os diários.

Com um gerenciador de filas de várias instâncias, o sistema de arquivos também é usado para controlar qual instância de um gerenciador de filas está ativa e qual está em espera. A instância ativa adquire um bloqueio para os dados do gerenciador de filas. A que está em espera aguarda para adquirir o bloqueio e quando o faz, ela se torna a instância ativa. O bloqueio é liberado pela instância ativa, se ela terminar normalmente. O bloqueio será liberado pelo sistema de arquivos se este detectar que a instância ativa falhou ou não pode acessar o sistema de arquivos. O sistema de arquivos deve atender aos requisitos para detectar falha; consulte [Requisitos para sistemas de arquivos compartilhados](#).

A arquitetura de gerenciadores de filas de várias instâncias no IBM i fornece reinicialização automática após uma falha no gerenciador de filas ou no servidor. Ele também suporta a restauração de dados do gerenciador de filas após uma falha do sistema de arquivos no qual os dados do gerenciador de filas são armazenados.

No [Figura 24 na página 430](#), se ALPHA falhar, será possível reiniciar manualmente QM1 em BETA, usando o diário espelhado. Ao incluir o recurso de gerenciador de filas de várias instâncias no QM1, a instância em espera de QM1 continuará automaticamente no BETA se a instância ativa no ALPHA falhar. QM1 também pode continuar automaticamente se ele for o servidor ALPHA que falha, não apenas a instância ativa do QM1. Depois que BETA se torna o host da instância ativa do gerenciador de filas, a instância de espera pode ser iniciada em ALPHA.

[Figura 24 na página 430](#) mostra uma configuração que espelha diários entre duas instâncias de um gerenciador de filas usando o NetServer para armazenar dados do gerenciador de filas. É possível expandir o padrão para incluir mais diários e, portanto, mais instâncias. Siga as regras de nomenclatura do diário explicadas no tópico, [“Diários do gerenciador de filas no IBM i” na página 409](#). Atualmente o número de instâncias em execução de um gerenciador de filas é limitado a dois, um está ativo e um está em espera.

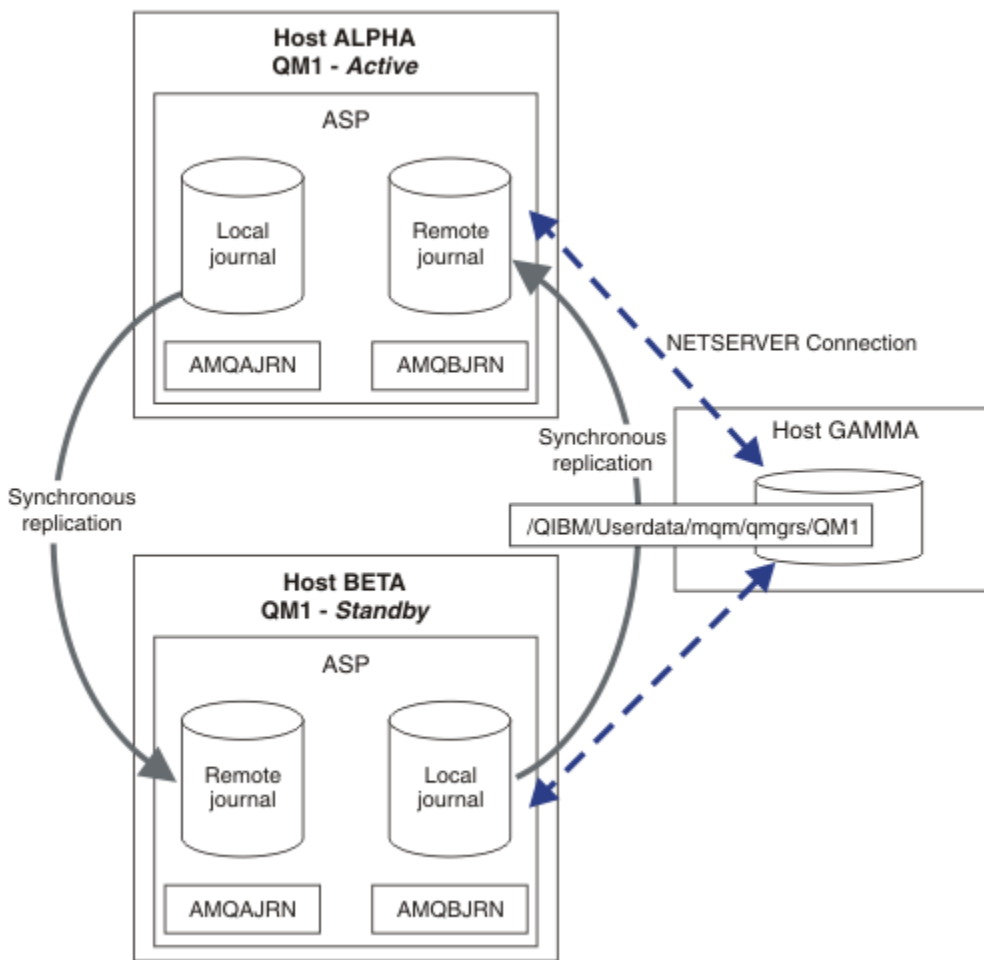


Figura 24. Espelhar um diário do gerenciador de filas

O diário local para QM1 no host ALPHA é chamado de AMQAJRN (ou mais, de forma mais completa, de QMQM1/AMQAJRN) e no BETA o diário é QMQM1/AMQBJRN. Cada diário local replica para diários remotos em todas as outras instâncias do gerenciador de filas. Se o gerenciador de filas estiver configurado com duas instâncias, um diário local será replicado para um diário remoto.

Replicação de diário remoto *SYNC ou *ASync

Diários do IBM i são espelhados usando qualquer um dos registros no diário síncronos (*SYNC) ou assíncronos (*ASync); consulte [Gerenciamento de diário remoto](#).

O modo de replicação em [Figura 24 na página 430](#) é *SYNC, não *ASync. *ASync é mais rápido, mas se ocorrer uma falha quando o estado do diário remoto for *ASyncPEND, o diário local e remoto não são consistentes. O diário remoto deve alcançar o diário local. Se você escolher *SYNC, o sistema local esperará o diário remoto antes de retornar de uma chamada que requer uma gravação concluída. Os diários locais e remotos geralmente permanecem consistentes entre si. Somente se a operação *SYNC demorar mais do que um tempo designado¹ e o registro no diário remoto estiver desativado, faça os diários saírem da sincronização. Um erro é registrado na fila de mensagens do diário e em QSYSOPR. O gerenciador de filas detecta essa mensagem, grava um erro no log de erros do gerenciador de filas e desativa a replicação remota do diário do gerenciador de filas. A instância ativa do gerenciador de filas continua sem o registro no diário remoto para este diário. Quando o servidor remoto está disponível novamente, deve-se reativar manualmente a replicação de diário remoto síncrona. Os diários são então resincronizados.

¹ O tempo designado é de 60 segundos no IBM i 5 e na faixa de 1 a 3600 segundos no IBM i 6.1 em diante.

Um problema com a configuração *SYNC / *SYNC ilustrada na Figura 24 na página 430 é como a instância do gerenciador de filas em espera em BETA assume o controle. Assim que a instância do gerenciador de filas em BETA grava sua primeira mensagem persistente, ela tenta atualizar o diário remoto no ALPHA. Se a causa de o controle ser passado de ALPHA para BETA foi a falha de ALPHA e ALPHA ainda está inativo, o registro no diário remoto em ALPHA falhará. BETA aguarda a resposta de ALPHA e, em seguida, desativa o registro no diário remoto e continua o processamento de mensagens com apenas o registro no diário local. BETA tem que esperar um pouco para detectar que ALPHA está inativo, causando um período de inatividade.

A opção entre configurar o registro no diário remoto para *SYNC ou *ASYNC é uma compensação. O Tabela 24 na página 431 resume as compensações entre o registro no diário *SYNC e *ASYNC entre um par de gerenciadores de filas:

<i>Tabela 24. Opções de registro no diário remoto</i>			
Ativo	Espera	*SYNC	*ASYNC
*SYNC		<ol style="list-style-type: none"> 1. Comutação e failover consistentes 2. A instância em espera não continua imediatamente após o failover. 3. O registro no diário remoto deve estar disponível todo o tempo 4. O desempenho do gerenciador de filas depende do registro no diário remoto 	<ol style="list-style-type: none"> 1. Comutação e failover consistentes 2. O registro no diário remoto deverá ser alternado para *SYNC quando o servidor de espera estiver disponível 3. O registro no diário remoto deverá permanecer disponível depois que ele tiver sido reiniciado 4. O desempenho do gerenciador de filas depende do registro no diário remoto
*ASYNC		<ol style="list-style-type: none"> 1. Não é uma combinação sensata 	<ol style="list-style-type: none"> 1. Algumas mensagens podem ser perdidas ou duplicadas após um failover ou uma comutação 2. A instância em espera não precisa estar disponível todo o tempo para a instância ativa para continuar sem demora. 3. O desempenho não depende do registro no diário remoto

***SYNC / *SYNC**

A instância ativa do gerenciador de filas usa o registro no diário *SYNC e, quando a instância do gerenciador de filas em espera é iniciada, ela tenta imediatamente usar o registro no diário *SYNC.

1. O diário remoto está consistente transacionalmente com o diário local do gerenciador de filas ativo. Se o gerenciador de filas é alternado para a instância em espera, ele poderá continuar imediatamente. A instância de espera normalmente é retomada sem perda ou duplicação de mensagens. As mensagens só serão perdidas ou duplicadas se o registro no diário remoto tiver falhado desde o último ponto de verificação e o gerenciador de filas ativo anteriormente não puder ser reiniciado.
2. Se o gerenciador de filas faz failover para a instância em espera, ele pode não ser capaz de iniciar imediatamente. A instância do gerenciador de filas em espera está ativada com o registro no diário *SYNC. A causa do failover pode impedir registro no diário remoto no servidor que hospeda a instância em espera. O gerenciador de filas aguarda até que o problema seja detectado antes de processar qualquer mensagem persistente. Um erro é registrado na fila de mensagens do diário e em QSYSOPR. O gerenciador de filas detecta essa mensagem, grava um erro no log de erros do gerenciador de filas e desativa a replicação remota do diário do gerenciador de filas. A instância ativa do gerenciador de filas continua sem o registro no diário remoto para este diário. Quando o

servidor remoto está disponível novamente, deve-se reativar manualmente a replicação de diário remoto síncrona. Os diários são então ressincronizados.

3. O servidor no qual o diário remoto é replicado deve estar sempre disponível para manter o diário remoto. O diário remoto é geralmente replicado para o mesmo servidor que hospeda o gerenciador de filas em espera. O servidor pode ficar indisponível. Um erro é registrado na fila de mensagens do diário e em QSYSOPR. O gerenciador de filas detecta essa mensagem, grava um erro no log de erros do gerenciador de filas e desativa a replicação remota do diário do gerenciador de filas. A instância ativa do gerenciador de filas continua sem o registro no diário remoto para este diário. Quando o servidor remoto está disponível novamente, deve-se reativar manualmente a replicação de diário remoto síncrona. Os diários são então ressincronizados.
4. O registro no diário remoto é mais lento do que o registro no diário local e substancialmente mais lento se os servidores são separados por uma distância grande. O gerenciador de filas deve aguardar pelo registro no diário remoto, o que reduz o desempenho do gerenciador de filas.

A configuração *SYNC / *SYNC entre um par de servidores tem a desvantagem de um atraso em continuar a instância em espera após o failover. A configuração *SYNC / *ASYNCR não tem este problema.

*SYNC / *ASYNCR garante nenhuma perda de mensagens depois da comutação ou do failover, contanto que um diário remoto esteja disponível. Se você deseja reduzir o risco de perda de mensagens após failover ou comutação, você tem duas opções. Parar a instância ativa se o diário remoto se tornar inativo ou criar diários remotos em mais de um servidor.

***SYNC / *ASYNCR**

A instância ativa do gerenciador de filas usa o registro no diário *SYNC e, quando a instância do gerenciador de filas em espera é iniciada, ele utiliza o registro no diário *ASYNCR. Pouco depois que o servidor que hospeda a nova instância em espera se torna disponível, o operador do sistema deve alternar o diário remoto no instância ativa para *SYNC. Quando o operador alterna o registro no diário remoto de *ASYNCR para *SYNC, a instância ativa será pausada se o status do diário remoto for *ASYNCRPEND. A instância ativa do gerenciador de filas aguarda até que as entradas de diário restantes sejam transferidas para o diário remoto. Quando o diário remoto for sincronizado com o diário local, a nova em espera será transacionalmente consistente de novo com a nova instância ativa. Da perspectiva do gerenciamento de gerenciadores de filas de várias instâncias, em uma configuração *SYNC / *ASYNCR, o operador do sistema IBM i possui uma tarefa adicional. O operador deve alternar o registro no diário remoto para *SYNC, além de reiniciar a instância do gerenciador de filas com falha.

1. O diário remoto está consistente transacionalmente com o diário local do gerenciador de filas ativo. Se a instância do gerenciador de filas ativa for comutada ou fizer failover para a instância em espera, a instância em espera poderá continuar imediatamente. A instância de espera normalmente é retomada sem perda ou duplicação de mensagens. As mensagens só serão perdidas ou duplicadas se o registro no diário remoto tiver falhado desde o último ponto de verificação e o gerenciador de filas ativo anteriormente não puder ser reiniciado.
2. O operador do sistema deve alternar diário remoto a partir do *ASYNCR para *SYNC logo após o sistema que está hospedando a instância ativa se torna disponível novamente. O operador pode aguardar o diário remoto alcançar antes de alternar o diário remoto para *SYNC. Como alternativa, o operador pode alternar a instância remota para *SYNC imediatamente e forçar a instância ativa a aguardar até que o diário da instância em espera tenha alcançado. Quando o registro no diário remoto é definido como *SYNC, a instância de espera geralmente é consistente de forma transacional com a instância ativa. As mensagens só serão perdidas ou duplicadas se o registro no diário remoto tiver falhado desde o último ponto de verificação e o gerenciador de filas ativo anteriormente não puder ser reiniciado.
3. Quando a configuração foi restaurada a partir de uma comutação ou um failover, o servidor no qual o diário remoto está hospedado deve estar disponível todo o tempo.

Escolha *SYNC / *ASYNCR quando você deseja que o gerenciador de filas em espera continue rapidamente após um failover. Deve-se restaurar a configuração do diário remoto para *SYNC na nova instância ativa manualmente. A configuração *SYNC / *ASYNCR corresponde ao padrão normal de administração de um par de gerenciadores de filas de várias instâncias. Após uma instância ter

falhado, há um tempo antes de a instância em espera ser reiniciada, durante o qual a instância ativa não pode executar failover.

***ASYNC / *ASYNC**

Ambos os servidores que hospedam os gerenciadores de filas ativos e em espera são configurados para usar o registro no diário *ASYNC remoto.

1. Quando a comutação ou o failover ocorrer, o gerenciador de filas continuará com o diário no novo servidor. O diário pode não ser sincronizado quando a comutação ou o failover ocorre. Conseqüentemente as mensagens poderão ser perdidas ou duplicadas.
2. A instância ativa será executada, mesmo se o servidor que hospeda o gerenciador de filas em espera não estiver disponível. O diário local é replicado assincronicamente com o servidor em espera quando ele está disponível.
3. O desempenho do gerenciador de filas locais não é afetado pelo registro no diário remoto.

Escolha *ASYNC / *ASYNC se o desempenho for seu proprietário requisito e se você estiver preparado para perder ou duplicar algumas mensagens após o failover ou a comutação.

***ASYNC / *SYNC**

Não há razão para se utilizar essa combinação de opções.

Ativação do gerenciador de filas a partir de um diário remoto

Diários são replicados de maneira síncrona ou assíncrona. O diário remoto pode não estar ativo ou pode estar alcançando o diário local. O diário remoto pode estar alcançando, mesmo se ele for replicado sincronicamente, porque ele pode ter sido recentemente ativado. As regras que o gerenciador de filas aplica ao estado do diário remoto são usadas durante a inicialização a seguir.

1. A inicialização em espera falhará se ela dever ser reproduzida a partir do diário remoto na espera e o status do diário for *FAILED ou *INACTPEND.
2. Quando a ativação da em espera se inicia, o status do diário remoto na espera deve ser *ACTIVE ou *INACTIVE. Se o estado for *INACTIVE, é possível que a ativação falhe, se nem todos os dados do diário foram replicados.

A falha ocorrerá se os dados do gerenciador de filas no sistema de arquivos de rede tiver um registro de ponto de verificação mais recente que o presente no diário remoto. A falha é improvável de acontecer, contanto que o diário remoto esteja bem ativado dentro do intervalo máximo padrão de 30 minutos entre pontos de verificação. Se o gerenciador de filas em espera ler um registro de ponto de verificação mais recente no sistema de arquivos, ele não será iniciado.

Você tem uma opção: aguarde até que o diário local no servidor ativo possa ser restaurado ou inicie a frio o gerenciador de filas em espera. Se você optar por iniciar a frio, o gerenciador de filas é iniciado sem dados de diário e conta com a consistência e a integridade dos dados do gerenciador de filas no sistema de arquivos.

Nota: Se você iniciar a frio um gerenciador de filas, você corre o risco de perder ou duplicar mensagens após o último ponto de verificação. As transações de mensagem foram gravadas no diário, mas algumas das transações podem não ter sido gravadas nos dados do gerenciador de filas no sistema de arquivos. Ao iniciar a frio um gerenciador de filas, um novo diário é iniciado e as transações não gravadas nos dados do gerenciador de filas no sistema de arquivos serão perdidas.

3. A ativação do gerenciador de filas em espera aguarda o status do diário remoto na espera para mudar de *ASYNCPEND ou *SYNCPEND para *ASYNC ou *SYNC. As mensagens são gravadas no log da tarefa do controlador de execução periodicamente.

Nota: Neste caso, a ativação está aguardando no local do diário remoto que o gerenciador de filas em espera seja ativado. O gerenciador de filas também aguarda por um momento antes de continuar sem um diário remoto. Ele aguarda quando ele tenta gravar de forma síncrona em seu diário (ou diários) remoto e o diário não está disponível.

4. A ativação para se o status do diário muda para *FAILED ou *INACTPEND.

Os nomes e estados dos diários locais e remotos a serem usados na ativação são gravados no log de erros do gerenciador de filas.

IBM i Criando um gerenciador de filas de várias instâncias usando o espelhamento de diário e o NetServer no IBM i

Crie um gerenciador de filas de várias instâncias para ser executado em dois servidores do IBM i. Os dados do gerenciador de filas são armazenados em um terceiro servidor do IBM i usando NetServer. O diário do gerenciador de filas é espelhado entre os dois servidores usando o registro no diário remoto. O comando **ADDQMJRN** é utilizado para simplificar a criação de diários remotos.

Antes de começar

1. A tarefa requer três servidores IBM i. Instale o IBM MQ em dois deles, ALPHA e BETA no exemplo. O produto deve estar pelo menos no IBM WebSphere MQ 7.0.1 Fix Pack 1.
2. O terceiro servidor é um servidor IBM i, conectado pelo NetServer para ALPHA e BETA. Ele é utilizado para compartilhar os dados do gerenciador de filas. Ele não precisa ter uma instalação do IBM MQ. É útil para instalar o IBM MQ no servidor como uma etapa temporária para configurar os diretórios e as permissões do gerenciador de filas.
3. Certifique-se de que o perfil do usuário QMQM tenha a mesma senha em todos os três servidores.
4. Instale o IBM i NetServer; consulte [i5/OS NetServer](#).

Sobre esta tarefa

Execute as etapas a seguir para criar a configuração mostrada em [Figura 25 na página 437](#). Os dados do gerenciador de filas são conectados usando o IBM i NetServer.

- Crie conexões a partir de ALPHA e BETA para o compartilhamento do diretório em GAMMA que deve armazenar os dados do gerenciador de filas. A tarefa também configura as permissões necessárias, os perfis de usuário e as senhas.
- Inclua Entradas do Banco de Dados Relacional (RDBE) em sistemas IBM i que vão executar instâncias do gerenciador de filas. As entradas RDBE são usadas para se conectar aos sistemas IBM i utilizados para registro no diário remoto.
- Crie o gerenciador de filas QM1 no servidor IBM i, ALPHA.
- Inclua as informações de controle do gerenciador de filas para QM1 no outro servidor IBM i, BETA.
- Crie diários remotos em ambos os servidores IBM i para ambas as instâncias do gerenciador de filas. Cada gerenciador de filas grava no diário local. O diário local é replicado para o diário remoto. O comando, **ADDQMJRN**, simplifica a inclusão dos diários e das conexões.
- Inicie o gerenciador de filas, permitindo uma instância em espera.

Procedimento

1. Execute a tarefa “[Criando um compartilhamento de rede para os dados do gerenciador de filas usando o NetServer no IBM i](#)” na página 421.

Como resultado, o ALPHA e o BETA têm uma participação, /QNTC/GAMMA/WMQ, que aponta para /QIBM/UserData/mqm/qmgrs no GAMMA. Os perfis do usuário QMQM e QMQMADM têm as permissões necessárias e QMQM tem senhas correspondentes em todos os três sistemas.

2. Inclua Entradas do Banco de Dados Relacional (RDBE) em sistemas IBM i que vão hospedar instâncias do gerenciador de filas.
 - a) Em ALPHA, crie a conexão para BETA.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) Em BETA, crie as conexões para ALPHA.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Crie o gerenciador de filas QM1 em ALPHA, salvando os dados do gerenciador de filas em GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP(' /QNTC/GAMMA/WMQ')
```

O caminho,, usa NetServer para criar os dados do gerenciador de filas.

4. Executar em ALPHA O comando inclui um diário remoto em BETA para.

```
ADDMQJRN MQMNAME(QM1) RMTJRN RDB(BETA)
```

cria entradas de diário em seu diário local em ALPHA quando a instância ativa de está em ALPHA. O diário local em ALPHA é replicado para o diário remoto no BETA.

5. Use o comando,, para inspecionar os dados de configuração IBM MQ criados por para ALPHA.

As informações são necessárias na próxima etapa.

Neste exemplo, a configuração a seguir é criada em ALPHA para:

```
Name=QM1  
Prefix=/QIBM/UserData/mqm  
Library=QMOM1  
Directory=QM1  
DataPath= /QNTC/GAMMA/WMQ /QM1
```

6. Crie uma instância do gerenciador de filas de QM1 no BETA usando o comando. Execute o comando a seguir em BETA para modificar as informações de controle do gerenciador de filas em BETA...

```
ADDMQMINF MQMNAME(QM1)  
PREFIX('/QIBM/UserData/mqm')  
MQMDIR(QM1)  
MQMLIB(QMOM1)  
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

Sugestão: Copie e cole as informações de configuração. A sub-rotina do gerenciador de filas é a mesmo em ALPHA e BETA.

7. Executar em BETA. O comando inclui um diário local em BETA e um diário remoto em ALPHA para.

```
ADDMQJRN MQMNAME(QM1) RMTJRN RDB(ALPHA)
```

cria entradas de diário em seu diário local em BETA quando a instância ativa de está em BETA. O diário local em BETA é replicado para o diário remoto em ALPHA.

Nota: Como alternativa, é possível desejar configurar o registro no diário remoto de BETA para ALPHA usando o diário assíncrono.

Use este comando para configurar o registro no diário assíncrono de BETA para ALPHA, em vez de usar o comando na etapa [“7”](#) na página 435.

```
ADDMQJRN MQMNAME(QM1) RMTJRN RDB(ALPHA) RMTJRN DLV(*ASYNCR)
```

Se o servidor ou o registro no diário em ALPHA é a origem da falha, BETA é iniciado sem esperar por novas entradas de diário serem replicadas para ALPHA.

Altere o modo de réplica para *SYNC, usando o comando, quando ALPHA estiver online novamente

Use as informações em [“Configuração de diário espelhado para ASP no IBM i”](#) na página 428 para decidir se deseja espelhar os diários de maneira síncrona, assíncrona ou uma mistura de ambos. O

padrão é replicar de maneira síncrona, com um período de espera de 60 segundos para uma resposta a partir do diário remoto.

8. Verifique se os diários em ALPHA e BETA estão ativados e se o status da replicação de diário remoto está.

a) Em ALPHA:

```
WRKMQMJRN MQMNAME(QM1)
```

b) Em BETA:

```
WRKMQMJRN MQMNAME(QM1)
```

9. Inicie as instâncias do gerenciador de filas em ALPHA e BETA.

a) Inicie a primeira instância em ALPHA, tornando-a a instância ativa. Ative a comutação para uma instância em espera.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) Inicie a segunda instância em BETA, tornando-a a instância em espera.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Resultados

Use para verificar o status do gerenciador de filas:

1. O status da instância do gerenciador de filas em ALPHA deve ser.
2. O status da instância do gerenciador de filas em BETA deve ser..

Exemplo

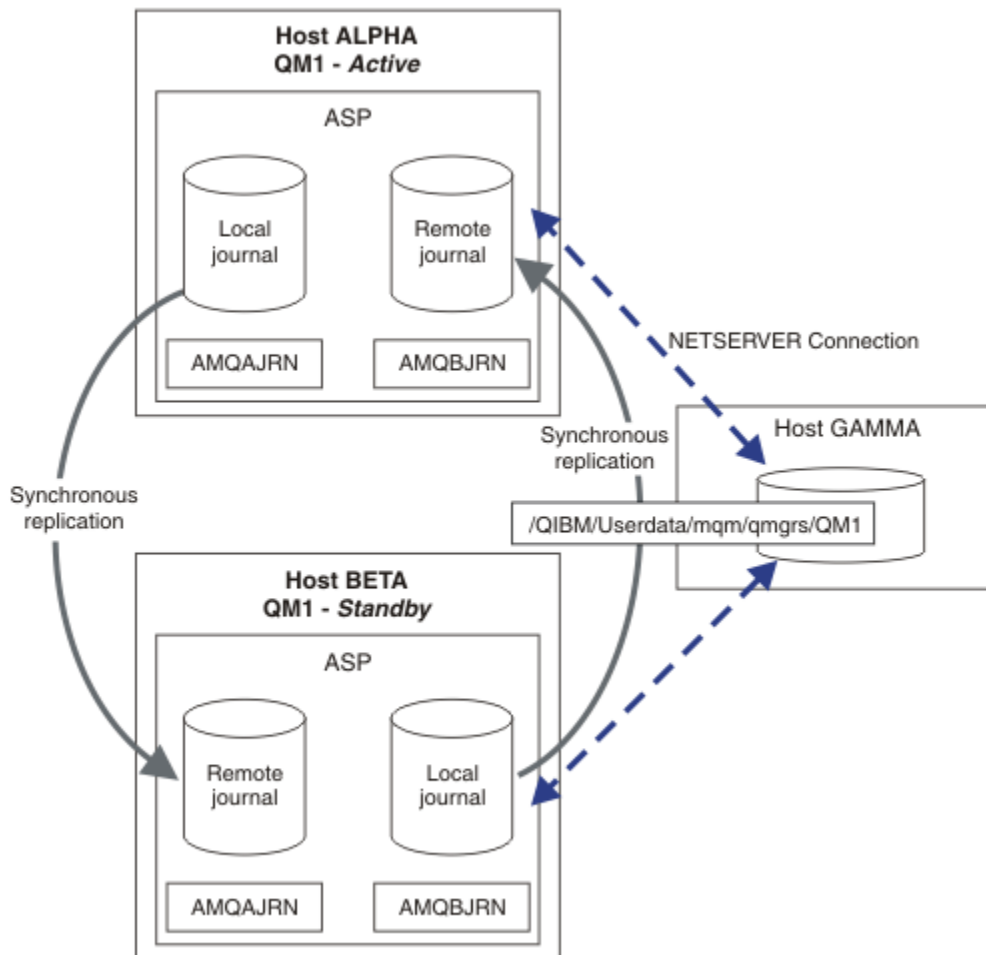


Figura 25. Configuração do diário espelhado

Como proceder a seguir

- Verifique se as instâncias ativas e em espera alternam automaticamente. É possível executar os programas de amostra de alta disponibilidade para testar a comutação; consulte [Programas de amostra de alta disponibilidade](#). Os programas de amostra são clientes 'C'. É possível executá-los a partir de uma plataforma Windows ou Unix.
 1. Inicie os programas de amostra de alta disponibilidade.
 2. Em ALPHA, termine o gerenciador de filas que solicita a comutação:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```
 3. Verifique se a instância de em BETA está ativa
 4. Reiniciar em ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```
- Veja as configurações de alta disponibilidade alternativas:
 1. Use o NetServer para colocar os dados do gerenciador de filas em um servidor Windows.

2. Em vez de utilizar o registro no diário remoto para espelhar o diário do gerenciador de filas, armazene o diário em um ASP independente. Use o armazenamento em cluster do IBM i para transferir o ASP independente de ALPHA para BETA.

IBM i *Convertendo um gerenciador de filas de instância única para um gerenciador de filas de várias instâncias usando NetServer e espelhamento diário no IBM i*

Converter um gerenciador de filas de instância única para um gerenciador de filas de várias instâncias. Mova os dados do gerenciador de filas para um compartilhamento de rede conectados por NetServer. Espelhe o diário do gerenciador de filas para um segundo servidor IBM i usando registro no diário remoto.

Antes de começar

1. A tarefa requer três servidores IBM i. A instalação existente do IBM MQ no servidor ALPHA no exemplo deve estar pelo menos no IBM WebSphere MQ 7.0.1 Fix Pack 1. ALPHA está executando um gerenciador de filas chamado QM1 no exemplo.
2. Instale o IBM MQ no segundo servidor IBM i, BETA no exemplo.
3. O terceiro servidor é um servidor IBM i, conectado pelo NetServer para ALPHA e BETA. Ele é utilizado para compartilhar os dados do gerenciador de filas. Ele não precisa ter uma instalação do IBM MQ. É útil para instalar o IBM MQ no servidor como uma etapa temporária para configurar os diretórios e as permissões do gerenciador de filas.
4. Certifique-se de que o perfil do usuário QMQM tenha a mesma senha em todos os três servidores.
5. Instale o IBM i NetServer; consulte [i5/OS NetServer](#).

Sobre esta tarefa

Execute as seguintes etapas para converter um gerenciador de filas de instância única com o gerenciador de filas de múltiplas instâncias mostrado em [Figura 26 na página 442](#). O gerenciador de filas de instância única é excluído na tarefa e, em seguida, recriado, armazenando os dados do gerenciador de filas no compartilhamento de rede conectado pelo NetServer. Este procedimento é mais confiável que a movimentação do gerenciador de diretórios e arquivos para o compartilhamento de rede usando o comando **CPY**.

- Crie conexões a partir de ALPHA e BETA para o compartilhamento do diretório em GAMMA que deve armazenar os dados do gerenciador de filas. A tarefa também configura as permissões necessárias, os perfis de usuário e as senhas.
- Inclua Entradas do Banco de Dados Relacional (RDBE) em sistemas IBM i que vão executar instâncias do gerenciador de filas. As entradas RDBE são usadas para se conectar aos sistemas IBM i utilizados para registro no diário remoto.
- Salve os logs do gerenciador de filas e definições, parar o gerenciador de filas e exclua-o.
- Recrie o gerenciador de filas, armazenando os dados do gerenciador de filas no compartilhamento de rede em GAMMA.
- Inclua a segunda instância do gerenciador de filas para o outro servidor.
- Crie diários remotos em ambos os servidores IBM i para ambas as instâncias do gerenciador de filas. Cada gerenciador de filas grava no diário local. O diário local é replicado para o diário remoto. O comando, **ADDMQMJRN**, simplifica a inclusão dos diários e das conexões.
- Inicie o gerenciador de filas, permitindo uma instância em espera.

Nota:

Na etapa “4” na [página 439](#) da tarefa, você exclui o gerenciador de filas de instância única, QM1. Excluindo o gerenciador de filas exclui todas as mensagens persistentes em filas. Por essa razão, conclua o processamento de todas as mensagens armazenadas pelo gerenciador de filas, antes de converter o gerenciador de filas. Se o processamento de todas as mensagens não for possível, faça backup da biblioteca do gerenciador de filas antes da etapa “4” na [página 439](#). Restaure a biblioteca do gerenciador de filas após a etapa “5” na [página 439](#).

Nota:

Na etapa “5” na página 439 da tarefa, você recria o QM1. Embora o gerenciador de filas tenha o mesmo nome, ele possui um identificador de gerenciador de filas diferente. O armazenamento em cluster do gerenciador de filas usa o identificador do gerenciador de filas. Para excluir e recriar um gerenciador de filas em um cluster, deve-se primeiro remover o gerenciador de filas do cluster; veja [Removendo um gerenciador de filas de um cluster: Método alternativo](#) ou [Removendo um gerenciador de filas de um cluster](#). Quando você tiver recriado o gerenciador de filas, inclua-o no cluster. Embora ele possua o mesmo nome de antes, parece ser um gerenciador de filas novo aos outros gerenciadores de filas no cluster.

Procedimento

1. Execute a tarefa “[Criando um compartilhamento de rede para os dados do gerenciador de filas usando o NetServer no IBM i](#)” na página 421.

Como resultado, o ALPHA e o BETA têm uma participação, /QNTC/GAMMA/WMQ, que aponta para /QIBM/UserData/mqm/qmgrs no GAMMA. Os perfis do usuário QMQM e QMQMADM têm as permissões necessárias e QMQM tem senhas correspondentes em todos os três sistemas.

2. Inclua Entradas do Banco de Dados Relacional (RDBE) em sistemas IBM i que vão hospedar instâncias do gerenciador de filas.
 - a) Em ALPHA, crie a conexão para BETA.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) Em BETA, crie as conexões para ALPHA.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Crie os scripts que recriam os objetos do gerenciador de filas.

```
QSAVEQMGR LCLQMGRNAM(QM1) FILENAME(' *CURLIB/QMQSC(QM1) ')  
OUTPUT(*REPLACE) MAKEAUTH(*YES) AUTHFN(' *CURLIB/QMAUT(QM1) ')
```

4. Pare o gerenciador de filas e exclua-o.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ENDCCTJOB(*YES) RCDMQMIMG(*YES) TIMEOUT(15)  
DLTMQM MQMNAME(QM1)
```

5. Crie o gerenciador de filas QM1 em ALPHA, salvando os dados do gerenciador de filas em GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

O caminho,, usa NetServer para criar os dados do gerenciador de filas.

6. Recrie os objetos do gerenciador de filas para QM1 das definições salvas.

```
STRMQMQSC SRCMBR(QM1) SRCFILE(*CURLIB/QMQSC) MQMNAME(QM1)
```

7. Aplique as autorizações a partir as informações salvas.

- a) Compile o programa de autorização salvo.

```
CRTCLPGM PGM(*CURLIB/QM1) SRCFILE(*CURLIB/QMAUT)  
SRCMBR(QM1) REPLACE(*YES)
```

- b) Execute o programa para aplicar as autorizações.

```
CALL PGM(*CURLIB/QM1)
```

- c) Atualizar as informações de segurança para QM1.

```
RFRMQMAUT MQMNAME(QM1)
```

8. Executar em ALPHA O comando inclui um diário remoto em BETA para.

```
ADDQMJRN MQMNAME(QM1) RMTJRNRDB(BETA)
```

cria entradas de diário em seu diário local em ALPHA quando a instância ativa de está em ALPHA. O diário local em ALPHA é replicado para o diário remoto no BETA.

9. Use o comando,, para inspecionar os dados de configuração IBM MQ criados por para ALPHA.

As informações são necessárias na próxima etapa.

Neste exemplo, a configuração a seguir é criada em ALPHA para:

```
Name=QM1  
Prefix=/QIBM/UserData/mqm  
Library=QMOM1  
Directory=QM1  
DataPath= /QNTC/GAMMA/WMQ /QM1
```

10. Crie uma instância do gerenciador de filas de QM1 no BETA usando o comando. Execute o comando a seguir em BETA para modificar as informações de controle do gerenciador de filas em BETA...

```
ADDQMINF MQMNAME(QM1)  
PREFIX('/QIBM/UserData/mqm')  
MQMDIR(QM1)  
MQMLIB(QMOM1)  
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

Sugestão: Copie e cole as informações de configuração. A sub-rotina do gerenciador de filas é a mesmo em ALPHA e BETA.

11. Executar em BETA. O comando inclui um diário local em BETA e um diário remoto em ALPHA para.

```
ADDQMJRN MQMNAME(QM1) RMTJRNRDB(ALPHA)
```

cria entradas de diário em seu diário local em BETA quando a instância ativa de está em BETA. O diário local em BETA é replicado para o diário remoto em ALPHA.

Nota: Como alternativa, é possível desejar configurar o registro no diário remoto de BETA para ALPHA usando o diário assíncrono.

Use este comando para configurar o registro no diário assíncrono de BETA para ALPHA, em vez de usar o comando na etapa “7” na página 435.

```
ADDQMJRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*ASYNC)
```

Se o servidor ou o registro no diário em ALPHA é a origem da falha, BETA é iniciado sem esperar por novas entradas de diário serem replicadas para ALPHA.

Altere o modo de réplica para *SYNC, usando o comando, quando ALPHA estiver online novamente

Use as informações em “[Configuração de diário espelhado para ASP no IBM i](#)” na página 428 para decidir se deseja espelhar os diários de maneira síncrona, assíncrona ou uma mistura de ambos. O padrão é replicar de maneira síncrona, com um período de espera de 60 segundos para uma resposta a partir do diário remoto.

12. Verifique se os diários em ALPHA e BETA estão ativados e se o status da replicação de diário remoto está.

a) Em ALPHA:

```
WRKMQMJRN MQMNAME(QM1)
```

b) Em BETA:

```
WRKMQMJRN MQMNAME(QM1)
```

13. Inicie as instâncias do gerenciador de filas em ALPHA e BETA.

a) Inicie a primeira instância em ALPHA, tornando-a a instância ativa. Ative a comutação para uma instância em espera.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) Inicie a segunda instância em BETA, tornando-a a instância em espera.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Resultados

Use para verificar o status do gerenciador de filas:

1. O status da instância do gerenciador de filas em ALPHA deve ser.
2. O status da instância do gerenciador de filas em BETA deve ser..

Exemplo

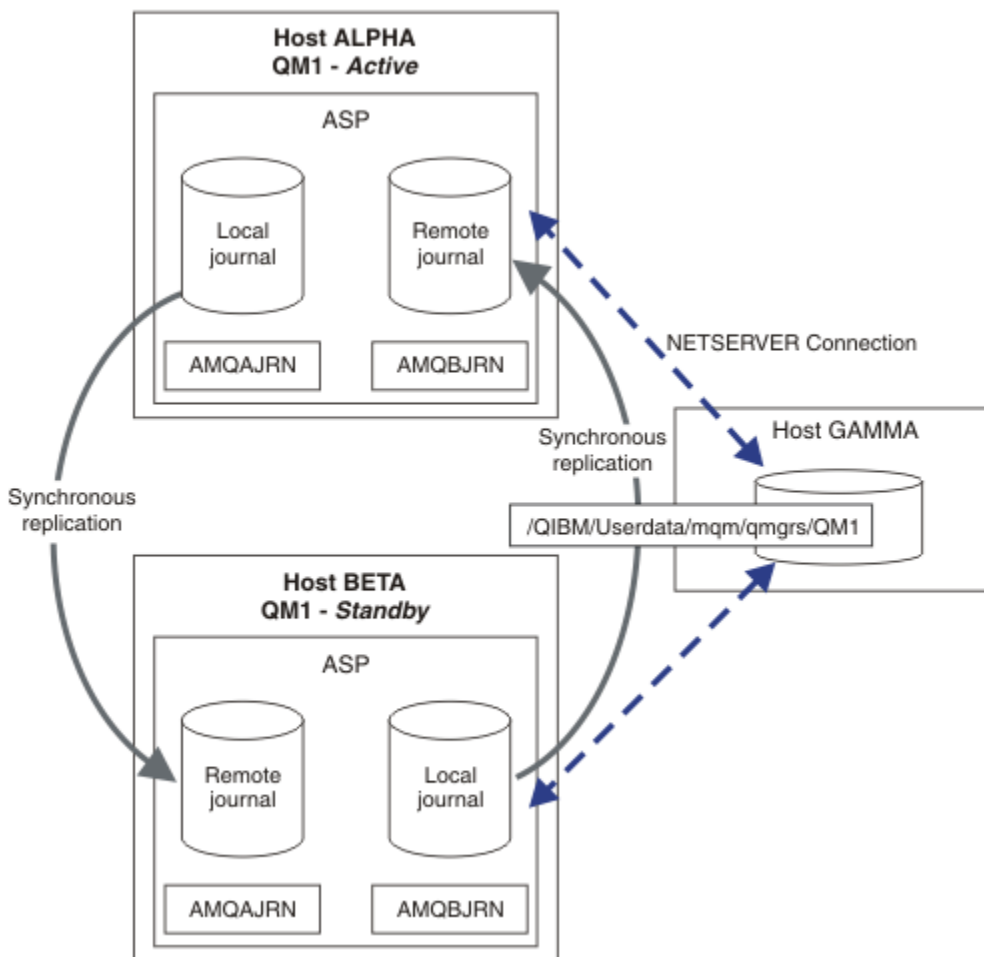


Figura 26. Configuração do diário espelhado

Como proceder a seguir

- Verifique se as instâncias ativas e em espera alternam automaticamente. É possível executar os programas de amostra de alta disponibilidade para testar a comutação; consulte [Programas de amostra de alta disponibilidade](#). Os programas de amostra são clientes 'C'. É possível executá-los a partir de uma plataforma Windows ou Unix.

1. Inicie os programas de amostra de alta disponibilidade.
2. Em ALPHA, termine o gerenciador de filas que solicita a comutação:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Verifique se a instância de em BETA está ativa
4. Reiniciar em ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Veja as configurações de alta disponibilidade alternativas:

1. Use o NetServer para colocar os dados do gerenciador de filas em um servidor Windows.

2. Em vez de utilizar o registro no diário remoto para espelhar o diário do gerenciador de filas, armazene o diário em um ASP independente. Use o armazenamento em cluster do IBM i para transferir o ASP independente de ALPHA para BETA.

IBM i **Configuração de diário ASP independente alternado no IBM i**

Você não precisa replicar um diário ASP independente para criar uma configuração do gerenciador de filas de várias instâncias. Você precisa automatizar um meio para transferir o ASP independente do gerenciador de filas ativo para o gerenciador de filas de espera. Existem alternativas possíveis soluções de alta disponibilidade utilizando um ASP independente, nem todos que requerem o uso de um gerenciador de filas de várias instâncias.

Ao usar um ASP independente, não é necessário espelhar o diário do gerenciador de filas. Se você tiver instalado o gerenciamento de cluster e os servidores que hospedam as instâncias do gerenciador de filas estão no mesmo grupo de recurso de cluster, então o diário do gerenciador de filas podem ser transferidos automaticamente para outro servidor em uma curta distância do servidor ativo, se o host que está executando a instância ativa falhar. Também é possível transferir o diário manualmente, como parte de uma comutação planejada ou é possível gravar um procedimento de comando para transferir o ASP independente programaticamente.

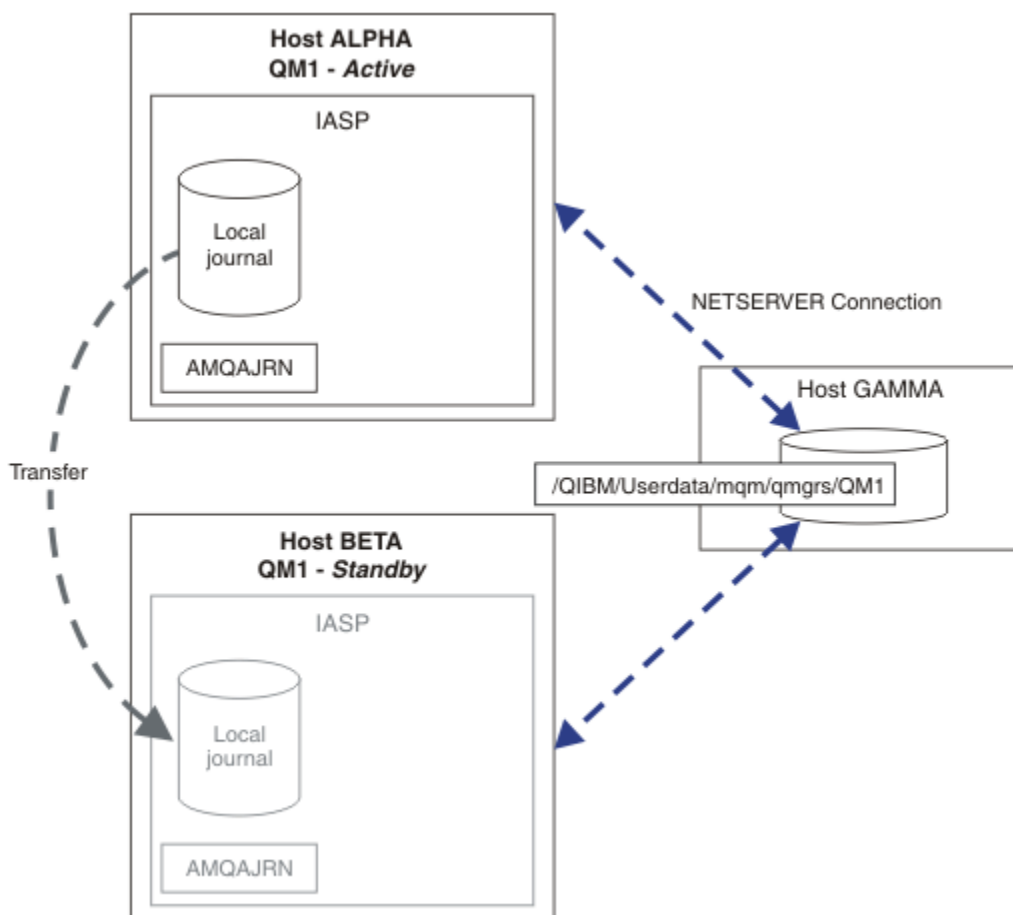


Figura 27. Transferir um diário do gerenciador de filas utilizando um ASP independente

Para a operação do gerenciador de filas de várias instâncias, os dados do gerenciador de filas devem ser armazenados em um sistema de arquivo compartilhado. O sistema de arquivos pode ser hospedado em uma variedade de plataformas diferentes. Não é possível armazenar gerenciador de filas de várias instâncias de dados em um ASP ou ASP independente.

O sistema de arquivo compartilhado desempenha duas funções na configuração: O dados do mesmo gerenciador de filas é compartilhada entre todas as instâncias do gerenciador de filas. O sistema de

arquivos deve ter um protocolo de bloqueio robusto que assegura que somente uma instância do gerenciador de filas tem acesso aos dados do gerenciador de filas após ele ter sido iniciado. Se o gerenciador de filas falhar ou a comunicação com o servidor divide o arquivo e, em seguida, o sistema de arquivo deve liberar o bloqueio para os dados do gerenciador de filas mantidos pela instância ativa que não está mais se comunicando com o sistema de arquivo. A instância do gerenciador de filas em espera, é possível obter acesso de leitura/gravação para os dados do gerenciador de filas. O protocolo de sistema de arquivos deve estar de acordo com um conjunto de regras para funcionar corretamente com gerenciadores de filas de várias instâncias; consulte [“Componentes de uma solução de alta disponibilidade no IBM i”](#) na página 420.

O mecanismo de bloqueio serializa o início do gerenciador de filas de comando e controla qual instância do gerenciador de filas está ativo. Depois que um gerenciador de filas se torna ativo, ele reconstruirá suas filas do diário local que você ou o cluster de HA, foi transferido para o servidor em espera. Os clientes reconectáveis que estão aguardando reconexão com o mesmo gerenciador de filas são reconectados e quaisquer transações em andamento são desfeitas. Os aplicativos que estão configurados para iniciar como serviços do gerenciador de filas são iniciados.

Você precisa assegurar que o diário local do gerenciador de filas de instância ativa no ASP independente é transferido para o servidor que hospeda a instância do gerenciador de filas de espera recentemente ativada, configurando o gerenciador de recursos do cluster ou transferir o ASP independente manualmente. Usando ASPs independentes não exclui a configuração de diários remotos e espelhamento, se você decidir utilizar ASP independente para backup e recuperação de desastre e utilizar o espelhamento diário remoto para configuração do gerenciador de filas de várias instâncias.

Se você tiver optado por utilizar um ASP independente, há configurações altamente disponíveis que é possível considerar a alternativa. O plano de fundo para essas soluções são descritos em [“ASPs independentes e alta disponibilidade”](#) na página 447.

1. Em vez de usar gerenciadores de filas de várias instâncias, instale e configure um gerenciador de filas de instância única inteiramente em um ASP independente e use os serviços de alta disponibilidade do IBM i para efetuar failover do gerenciador de filas. Você, provavelmente precisará aumentar a solução com um monitor do gerenciador de filas para detectar se o gerenciador de filas tiver falhado independentemente do servidor. Esta é a base da solução fornecida em, *Supportpac MC41: Configurando o IBM MQ para iSeries para Alta Disponibilidade*.
2. Utilize ASPs independentes e o espelhamento de sites cruzados (XSM) para espelhar o ASP independente em vez de comutar o ASP independente no barramento local. Isso estende a faixa geográfica do ASP independente a solução até que o tempo gasto para gravar registros de log a uma grande distância permite.

Criando um gerenciador de filas de várias instâncias usando um ASP independente e o NetServer no IBM i

Crie um gerenciador de filas de várias instâncias para ser executado em dois servidores do IBM i. Os dados do gerenciador de filas são armazenados em um servidor IBM i usando NetServer. O diário do gerenciador de filas é armazenado em um ASP independente. Use o procedimento de armazenamento em cluster ou manual do IBM i para transferir o ASP independente que contém o diário do gerenciador de filas para o outro servidor IBM i.

Antes de começar

1. A tarefa requer três servidores IBM i. Instale o IBM MQ em dois deles, ALPHA e BETA no exemplo. O produto deve estar pelo menos no IBM WebSphere MQ 7.0.1 Fix Pack 1.
2. O terceiro servidor é um servidor IBM i, conectado pelo NetServer para ALPHA e BETA. Ele é utilizado para compartilhar os dados do gerenciador de filas. Ele não precisa ter uma instalação do IBM MQ. É útil para instalar o IBM MQ no servidor como uma etapa temporária para configurar os diretórios e as permissões do gerenciador de filas.
3. Certifique-se de que o perfil do usuário QMQM tenha a mesma senha em todos os três servidores.
4. Instale o IBM i NetServer; consulte [i5/OS NetServer](#).

5. Criar procedimentos para transferir o ASP independente do gerenciador de filas com falha na espera que está ocorrendo. É possível localizar algumas das técnicas em *SupportPac MC41: Configurando o IBM MQ para iSeries para Alta Disponibilidade* útil em seu projeto de ASP independente de transferência de procedimentos.

Sobre esta tarefa

Execute as etapas a seguir para criar a configuração mostrada em [Figura 28 na página 446](#). Os dados do gerenciador de filas são conectados usando o IBM i NetServer.

- Crie conexões a partir de ALPHA e BETA para o compartilhamento do diretório em GAMMA que deve armazenar os dados do gerenciador de filas. A tarefa também configura as permissões necessárias, os perfis de usuário e as senhas.
- Crie o gerenciador de filas QM1 no servidor IBM i, ALPHA.
- Inclua as informações de controle do gerenciador de filas para QM1 no outro servidor IBM i, BETA.
- Inicie o gerenciador de filas, permitindo uma instância em espera.

Procedimento

1. Execute a tarefa [“Criando um compartilhamento de rede para os dados do gerenciador de filas usando o NetServer no IBM i”](#) na página 421.

Como resultado, o ALPHA e o BETA têm uma participação, /QNTC/GAMMA/WMQ, que aponta para /QIBM/UserData/mqm/qmgrs no GAMMA. Os perfis do usuário QMQM e QMQMADM têm as permissões necessárias e QMQM tem senhas correspondentes em todos os três sistemas.

2. Crie o gerenciador de filas QM1 em ALPHA, salvando os dados do gerenciador de filas em GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

O caminho,, usa NetServer para criar os dados do gerenciador de filas.

3. Use o comando,, para inspecionar os dados de configuração IBM MQ criados por para ALPHA.

As informações são necessárias na próxima etapa.

Neste exemplo, a configuração a seguir é criada em ALPHA para:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

4. Crie uma instância do gerenciador de filas de QM1 no BETA usando o comando. Execute o comando a seguir em BETA para modificar as informações de controle do gerenciador de filas em BETA...

```
ADDQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH(' /QNTC/GAMMA/WMQ /QM1 ')
```

Sugestão: Copie e cole as informações de configuração. A sub-rotina do gerenciador de filas é a mesmo em ALPHA e BETA.

5. Inicie as instâncias do gerenciador de filas em ALPHA e BETA.

- a) Inicie a primeira instância em ALPHA, tornando-a a instância ativa. Ative a comutação para uma instância em espera.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) Inicie a segunda instância em BETA, tornando-a a instância em espera.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Resultados

Use para verificar o status do gerenciador de filas:

1. O status da instância do gerenciador de filas em ALPHA deve ser.
2. O status da instância do gerenciador de filas em BETA deve ser..

Exemplo

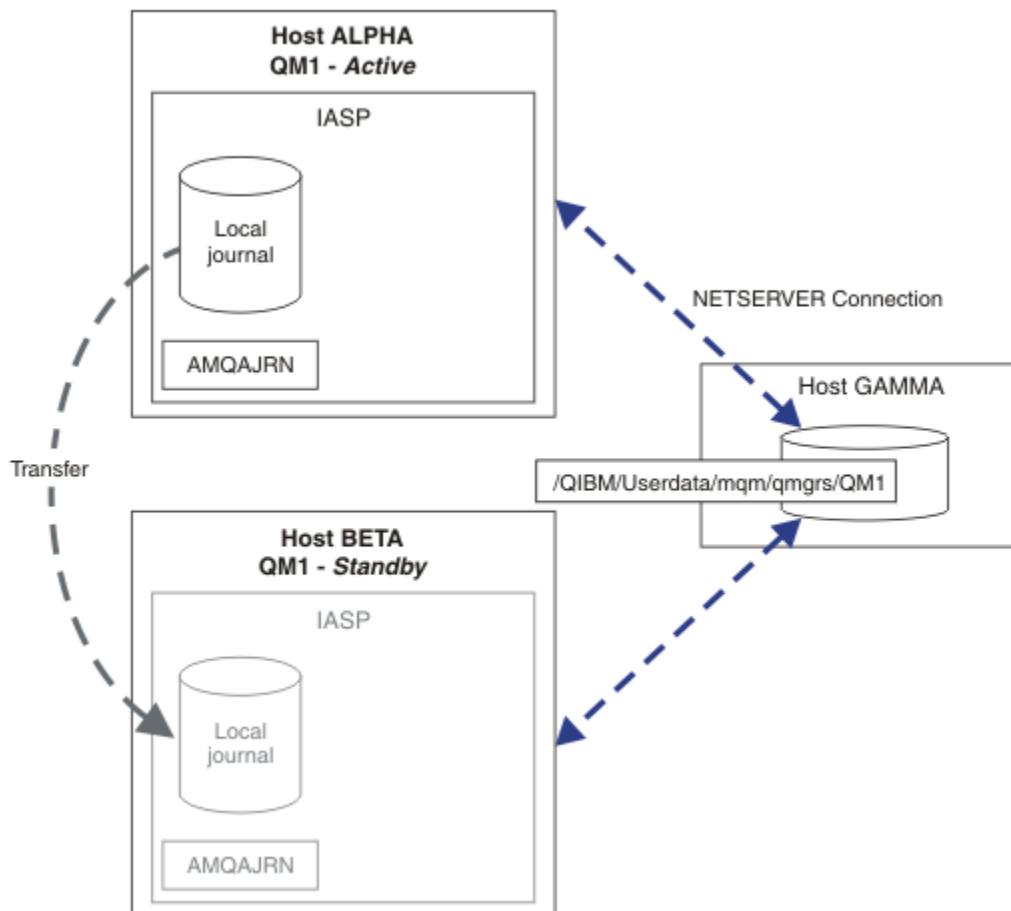


Figura 28. Transferir um diário do gerenciador de filas utilizando um ASP independente

Como proceder a seguir

- Verifique se as instâncias ativas e em espera alternam automaticamente. É possível executar os programas de amostra de alta disponibilidade para testar a comutação; consulte [Programas de amostra de alta disponibilidade](#). Os programas de amostra são clientes 'C'. É possível executá-los a partir de uma plataforma Windows ou Unix.
 1. Inicie os programas de amostra de alta disponibilidade.
 2. Em ALPHA, termine o gerenciador de filas que solicita a comutação:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Verifique se a instância de em BETA está ativa
4. Reiniciar em ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Veja as configurações de alta disponibilidade alternativas:
 1. Use o NetServer para colocar os dados do gerenciador de filas em um servidor IBM i .
 2. Em vez de utilizar um ASP independente para transferir o diário do gerenciador de filas para o servidor em espera, utilize o registro no diário remoto para espelhar o diário para o servidor em espera.

IBM i ASPs independentes e alta disponibilidade

ASPs independentes permitem que os aplicativos e os dados a serem movidos entre servidores. A flexibilidade de ASPs independentes significa que eles são a base para alguns IBM i soluções de alta disponibilidade. Ao ponderar se utilizar um ASP ou ASP independente para o diário do gerenciador de filas, é necessário considerar a configuração de alta disponibilidade no outro baseado em ASPs independentes.

Os conjuntos de armazenamentos auxiliar (ASPs) são um bloco de construção de arquitetura do IBM i. As unidades de disco são agrupados para formar um único ASP. Ao colocar objetos em ASPs diferente, é possível proteger os dados em um ASP de serem afetados por falhas de disco em outro ASP.

Cada servidor IBM i tem pelo menos um ASP *básico*, conhecido como o ASP do sistema. Ele é designado como ASP1 e, às vezes, conhecido como *SYSBAS. É possível configurar até 31 ASPs de *usuário* básicas adicionais que são distinguidas a partir do sistema ASP do ponto de vista do aplicativo, pois eles compartilham o mesmo espaço de nomes. Usando vários ASPs básicos para distribuir aplicativos em vários discos, é possível melhorar o desempenho e reduzir o tempo de recuperação. O uso de vários ASPs básicos também pode fornecer algum grau de isolamento contra falhas no disco, mas isso não melhora a confiabilidade geral.

ASPs independentes são um tipo especial de ASP. Eles são frequentemente chamados de conjuntos de discos independentes. Os conjuntos de discos independentes são componentes chave da alta disponibilidade do IBM i. É possível armazenar dados e aplicativos que se refere como independente do sistema atual aos quais eles estão conectados em unidades de armazenamento em disco independentes. É possível configurar ASPs independentes comutáveis ou não comutáveis. De uma perspectiva geral está somente preocupado com disponibilidade alternável ASPs independentes, que podem ser transferidos automaticamente a partir de servidor para servidor. Como resultado, é possível mover os aplicativos e dados no ASP independente de servidor para servidor.

Diferentemente de ASPs de usuário básico, ASPs independentes não compartilham o mesmo espaço de nomes como o ASP do sistema. Os aplicativos que funcionam com ASPs do usuário requerem mudanças para trabalhar com um ASP independente. Você precisa verificar se seu software e software de terceiros que você utilizar, funciona em um ambiente ASP independente.

Quando o ASP independente está conectado a um servidor diferente do espaço de nomes do ASP independente deve ser combinado com o espaço de nomes do ASP do sistema. Esse processo é denominado *variável em* o ASP independente. É possível variar em um ASP independente sem IPLing o servidor. O suporte de cluster é necessário para transferir os ASPs independentes automaticamente de um servidor para outro.

Construindo confiável com soluções ASPs independentes

Registrar no diário para um ASP independente, em vez registrar no diário para um ASP e utilizar a replicação de diário, fornece um meio alternativo para fornecer o gerenciador de filas de espera com uma cópia do diário local da instância do gerenciador de filas que falhou. Para transferir automaticamente o ASP independente para outro servidor, você precisa ter instalado e configurado o suporte ao clustering. Existem diversas soluções de alta disponibilidade para ASPs independentes com base no suporte de

cluster e espelhamento de disco de nível inferior, que é possível combinar com ou substituir, utilizando gerenciadores de filas de várias instâncias.

A lista a seguir descreve os componentes que são necessários para construir uma solução confiável com base em ASPs independentes.

Registrar alterações

Os gerenciadores de filas e outros aplicativos, utilize diários locais com segurança para disco para gravar dados persistentes para proteger contra perda de dados na memória devido a falha do servidor. Às vezes, isso é referido como consistência de momento. Ela não garante a consistência de várias atualizações que ocorrem durante um período de tempo.

Controle de consolidações

Ao utilizar transações globais, é possível coordenar atualizações para mensagens e bancos de dados para que os dados gravados no diário sejam consistentes. Ele fornece consistência durante um período de tempo, utilizando um protocolo de consolidação de duas fases.

Disco alternado

Discos comutados são gerenciados pelo grupo de recursos do cluster do dispositivo (CRG) em um cluster de HA. CRG ASPs independentes alterna automaticamente para um novo servidor no caso de uma interrupção não planejada. CRGs estão geograficamente limitadas à extensão do barramento de ES local.

Ao configurar o seu diário local em um ASP independente comutável, é possível transferir o diário para um servidor diferente e continuar o processamento de mensagens. Nenhuma mudança para mensagens persistentes feitas sem controle de ponto de sincronização ou confirmados com controle de ponto de sincronização, são perdas, a menos que o ASP independente falhará.

Se você utilizar ambos, o registro no diário e o controle de consolidação em ASPs independentes alternáveis, é possível transferir diários de banco de dados e diários do gerenciador de filas para um servidor diferente e retomar o processamento de transações sem perda de consistência ou transações consolidadas.

Espelhamento de sites cruzados (XSM)

O XSM espelha a ASP independente primário para um ASP independente secundário geograficamente remoto em uma rede TCP/IP e transfere o controle automaticamente em caso de uma falha. Você tem a opção de configurar um espelho síncrono ou assíncrono. O espelhamento síncrono reduz o desempenho do gerenciador de filas porque os dados são espelhados antes das operações de gravação no sistema de produção serem concluídas, mas garante que o ASP independente secundário esteja atualizado. Considerando que se você utilizar o espelhamento assíncrono você não poderá garantir que o ASP independente secundário esteja atualizado. O espelhamento assíncrono mantém a consistência do ASP independente secundário.

Há três tecnologias de XSM.

Geographic Mirror

O espelhamento geográfico é uma extensão do armazenamento em cluster que permite comutar ASPs independentes em uma ampla área. Ele tem ambos os modos síncronos e assíncronos. É possível garantir alta disponibilidade somente no modo síncrono, mas a separação de ASPs independentes pode impactar demais o desempenho. É possível combinar o espelhamento geográfico com disco alternado para fornecer alta disponibilidade localmente e recuperação de desastre remotamente.

Metro Mirror

Espelhamento Metro é um serviço no nível do dispositivo que fornece espelhamento síncrono local em distâncias mais rápido do que o barramento local. É possível combinar isso com um gerenciador de filas de várias instâncias para fornecer alta disponibilidade do gerenciador de filas e por ter duas cópias do ASP independente, alta disponibilidade do diário do gerenciador de filas.

Global Mirror

Espelhamento Global é o nível de dispositivo de serviço que fornece espelhamento assíncrono e é adequado para backup e recuperação de desastre por distâncias maiores, mas não é uma

opção normal para alta disponibilidade, porque somente mantém consistência momento em vez de moeda.

Os principais pontos de decisão que é necessário considerar são,

ASP ou ASP independente?

Não é necessário executar um cluster de alta disponibilidade do IBM i para utilizar gerenciadores de filas de várias instâncias. É possível escolher ASPs independentes, se você já estiver utilizando ASPs independentes ou você tiver requisitos de disponibilidade para outros aplicativos que requerem ASPs independentes. Pode ser útil combinar os ASPs independentes com gerenciadores de filas de várias instâncias para substituir o monitoramento do gerenciador de filas como um meio de detecção de falha do gerenciador de filas.

Disponibilidade?

Qual é o objetivo de tempo de recuperação (RTO)? Se você requerer a aparência do comportamento quase ininterrupto, então qual solução tem o tempo de recuperação mais rápida ?

Disponibilidade do diário?

Como você elimina o diário como um único ponto de falha. É possível adotar uma solução de hardware, utilizando os dispositivos RAID 1 ou melhores ou você pode combinar ou utilizar uma solução de software utilizando espelhamento de disco ou diários de réplica.

Distância?

Como distantes são os ativos e em espera do gerenciador de instâncias. Seus usuários podem tolerar a degradação do desempenho de replicação assincronicamente em distâncias superiores a cerca de 250 metros?

Habilidades?

Há trabalho a ser feito para automatizar as tarefas administrativas envolvidas na manutenção e utilizar a solução regularmente. As habilidades necessárias para fazer a automação são diferentes para as soluções baseadas em ASPs e ASPs independentes.

Excluindo um gerenciador de filas de várias instâncias no IBM i

Antes de excluir um gerenciador de filas de várias instâncias, pare o registro no diário remoto e remova as instâncias do gerenciador de filas.

Antes de começar

1. Neste exemplo, duas instâncias do gerenciador de filas QM1 são definidos nos servidores ALPHA e BETA. ALPHA é a instância ativa e BETA é a espera. Os dados do gerenciador de filas associado com o gerenciador de filas QM1 é armazenado no servidor IBM i GAMMA, utilizando o NetServer. Consulte [“Criando um gerenciador de filas de várias instâncias usando o espelhamento de diário e o NetServer no IBM i”](#) na página 434.
2. ALPHA e BETA devem estar conectados para que quaisquer diários remotos que estejam definidos podem ser excluídos pelo IBM MQ.
3. Verifique se o diretório /QNTC e o compartilhamento de arquivo do diretório do servidor podem ser acessados usando os comandos de sistema **EDTF** ou **WRKLNK**

Sobre esta tarefa

Antes de excluir um gerenciador de filas de várias instâncias de um servidor usando o comando **DLTMQM**, remova quaisquer instâncias do gerenciador de filas em outros servidores usando o comando **RMVMQMINF**.

Quando você remover uma instância do gerenciador de filas utilizando o comando **RMVMQMINF**, diários locais e remotos prefixado com AMQe associado à instância, são excluídos. As informações de configuração sobre a instância do gerenciador de filas, local para o servidor, também será excluído.

Não execute o comando **RMVMQMINF** no servidor que contém o restante da instância do gerenciador de filas. Fazer isso impede que **DLTMQM** de funcionar corretamente.

Exclua o gerenciador de filas utilizando o comando **DLTMQM**. dados do gerenciador de filas são removidos do compartilhamento de rede. diários locais e remotos prefixado com AMQ e associado à instância são excluídos. **DLTMQM** também exclui as informações de configuração sobre a instância do gerenciador de filas, local para o servidor.

No exemplo, existem somente duas instâncias do gerenciador de filas. O IBM MQ suporta uma configuração de várias instâncias em execução que tenha uma instância ativa do gerenciador de filas e uma instância em espera. Se você tiver criado instâncias do gerenciador de filas adicionais para uso em configurações em execução, remova-os, usando o comando **RMVMQMINF**, antes de excluir a instância restantes.

Procedimento

1. Execute o comando **CHGMQMJRN RMTJRNSTS (*INACTIVE)** em cada servidor para tornar o diário remoto entre as instâncias do gerenciador de filas inativo.

a) Em ALPHA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('BETA') RMTJRNSTS(*INACTIVE)
```

b) Em BETA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('ALPHA') RMTJRNSTS(*INACTIVE)
```

2. Execute o comando **ENDMQM** em ALPHA, a instância ativa do gerenciador de filas, para parar ambas as instâncias de QM1.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) INSTANCE(*ALL) ENDCCTJOB(*YES)
```

3. Execute o comando **RMVMQMINF** em ALPHA para remover os recursos do gerenciador de filas para a instância de ALPHA e BETA.

```
RMVMQMINF MQMNAME(QM1)
```

RMVMQMINF remove as informações de configuração do gerenciador de filas para QM1 do ALPHA. Se o nome do diário é prefixado por AMQ, ele exclui o diário local associado ao QM1 de ALPHA. Se o nome do diário é prefixado por AMQ e um diário remoto tiver sido criado, ele também remove o diário remoto a partir de BETA.

4. Execute o comando **DLTMQM** em BETA para excluir QM1.

```
DLTMQM MQMNAME(QM1)
```

DLTMQM exclui os dados do gerenciador de filas do compartilhamento de rede em GAMMA. Ele remove as informações de configuração do gerenciador de filas para QM1 do BETA. Se o nome do diário é prefixado por AMQ, ele exclui o diário local associado a QM1 de BETA. Se o nome do diário é prefixado por AMQ e um diário remoto tiver sido criado, ele também remove o diário remoto a partir de ALPHA.

Resultados

DLTMQM e **RMVMQMINF** excluem os diários local e remoto criados por **CRTMQM** e **ADDMQJRN**. Os comandos também excluem os receptores de diário. Os diários e os receptores de diário devem seguir a convenção de nomenclatura de ter nomes que começam com AMQ. **DLTMQM** e **RMVMQMINF** remover os objetos do gerenciador de filas, os dados do gerenciador de filas e as informações de configuração do gerenciador de filas a partir de `mqs.ini`.

Como proceder a seguir

Uma abordagem alternativa é emitir os comandos a seguir após desativar o registro no diário na etapa “1” na [página 450](#) e antes do término das instâncias do gerenciador de filas. Ou, se você não tiver seguido a convenção de nomenclatura, deve-se excluir os diários e os receptores de diário por nome.

1. Em ALPHA:

```
RMVVMQMJRN MQMNAME('QM1') RMTJRNRDB('BETA')
```

2. Em BETA:

```
RMVVMQMJRN MQMNAME('QM1') RMTJRNRDB('ALPHA')
```

Depois de excluir os diários, continue com o restante das etapas.

Fazendo backup de um gerenciador de filas de várias instâncias no IBM i

O procedimento mostra como fazer backup de objetos do gerenciador de filas no servidor local e os dados do gerenciador de filas no servidor de arquivos da rede. Adapte o exemplo para fazer backup de dados para outros gerenciadores de filas.

Antes de começar

Neste exemplo, os dados do gerenciador de filas associado ao gerenciador de filas QM1 é armazenado no servidor chamado IBM i GAMMA, usando NetServer. Consulte [“Criando um gerenciador de filas de várias instâncias usando o espelhamento de diário e o NetServer no IBM i”](#) na [página 434](#). IBM MQ está instalado nos servidores, ALPHA e BETA. O gerenciador de filas, QM1, é configurado em ALPHA e BETA.

Sobre esta tarefa

IBM i não suporta o salvamento de dados a partir de um diretório remoto. Salve os dados do gerenciador de filas em um sistema de arquivos remoto utilizando os procedimentos de backup local para o servidor do sistema de arquivos. Nesta tarefa, o sistema de arquivos de rede está em um servidor IBM i, GAMMA. O backup dos dados do gerenciador de filas é feito em um arquivo de salvamento em GAMMA.

Se o sistema de arquivos de rede estava no Windows ou Linux, é possível armazenar os dados do gerenciador de filas em um arquivo compactado e, em seguida, salve-o. Se você tiver um sistema de backup, como o Tivoli Storage Manager, utilize-o para fazer backup dos dados do gerenciador de filas.

Procedimento

1. Crie um arquivo de salvamento no ALPHA para a biblioteca do gerenciador de filas associado ao QM1. Utilize o nome da biblioteca do gerenciador de filas para o nome do arquivo de salvamento.

```
CRTSAVF FILE(QGPL/QMQM1)
```

2. Salve a biblioteca do gerenciador de filas no arquivo de salvamento em ALPHA.

```
SAVLIB LIB(QMQM1) DEV(*SAVF) SAVF(QGPL/QMQM1)
```

3. Crie um arquivo de gravação para o diretório de dados do gerenciador de filas em GAMMA. Utilize o nome do gerenciador de filas para o nome do arquivo de salvamento.

```
CRTSAVF FILE(QGPL/QMDQM1)
```

4. Salve a cópia dos dados do gerenciador de filas do diretório local no GAMMA.

```
SAV DEV('/QSYS.LIB/QGPL.LIB/QMDQM1.FILE') OBJ('/QIBM/Userdata/mqm/qmgrs/QM1')
```

IBM i

Comandos para configurar gerenciadores de filas de várias instâncias

IBM MQ tem comandos para simplificar a configuração de replicação de diário, incluindo novas instâncias do gerenciador de filas e a configuração dos gerenciadores de filas para utilizar o ASP independente.

Os comandos de diário para criar e gerenciar diários locais e remotos são,

ADDMQMJRN

Com esse comando é possível criar diários locais e remotos denominados para uma instância do gerenciador de filas e configurar se a replicação é síncrona ou assíncrona, o que o tempo limite síncrono é e se o diário remoto deve ser ativado imediatamente.

CHGMQMJRN

O comando modifica os parâmetros de tempo limite, status e entrega que afetam os diários de réplica.

RMVMQMJRN

Remove diários *remotos* denominados a partir de uma instância do gerenciador de filas.

WRKMQMJRN

Lista o status de diários locais e remotos para uma instância do gerenciador de filas locais.

Inclua e gerencie instâncias adicionais do gerenciador de filas usando os seguintes comandos, que modificam o arquivo `mqs.ini`.

ADDMQMINF

O comando usa informações extraídas do arquivo `mqs.ini` com o comando `DSPMQMINF` para incluir uma nova instância do gerenciador de filas em um servidor IBM i diferente.

RMVMQMINF

Remova uma instância do gerenciador de filas. Utilize este comando para remover uma instância de um gerenciador de filas existente ou para remover as informações de configuração para um gerenciador de filas que foi excluído de um servidor diferente.

O comando **CRTMQM** tem três parâmetros para ajudar na configuração de um gerenciador de filas de várias instâncias,

MQMDIRP (*DFT | *directory-prefix*)

Utilize este parâmetro para selecionar um ponto de montagem que é mapeado para os dados do gerenciador de filas no armazenamento em rede.

ASP (*SYSTEM | *ASPDEV | *auxiliary-storage-pool-number*)

Especifique `*SYSTEM` ou um *auxiliary-storage-pool-number* para colocar o diário do gerenciador de filas no sistema ou um ASP de usuário básico. Selecione a opção `*ASPDEV` e também configure um nome de dispositivo utilizando o parâmetro **ASPDEV** para colocar o diário do gerenciador de filas em um ASP independente.

ASPDEV (*ASP | *device-name*)

Especifique um *device-name* de um dispositivo ASP independente primário ou secundário. Selecionar `*ASP` tem o mesmo resultado que a especificação de **ASP** (`*SYSTEM`).

IBM i

Considerações de desempenho e de failover de disco no IBM i

Utilize diferentes conjuntos de armazenamentos auxiliares para melhorar o desempenho e a confiabilidade.

Se você utilizar um grande número de mensagens persistentes ou mensagens grandes em seus aplicativos, o tempo gasto para gravar estas mensagens em disco se tornará um fator significativo no desempenho do sistema.

Assegure que você tenha ativação em disco suficiente para lidar com essa possibilidade ou considere um Auxiliary Storage Pool (ASP) separado no qual reter receptores de diário do gerenciador de filas.

É possível especificar qual ASP a sua biblioteca do gerenciador de filas e os diários são armazenados quando você cria seu gerenciador de filas usando o parâmetro ASP do **CRTMQM**. Por padrão, a biblioteca do gerenciador de filas e os diários e os dados do IFS são armazenados no ASP do sistema.

As ASPs permitem isolamento de objetos em uma ou mais unidades de disco específicas. Isso também pode reduzir a perda de dados devido a uma falha de mídia de disco. Na maioria dos casos, somente os dados que estão armazenados em unidades de disco na ASP afetada são perdidos.

É recomendado armazenar a biblioteca do gerenciador de filas e dados de diário em ASPs de usuário separadas daquela do sistema de arquivos IFS raiz para fornecer failover e reduzir a contenção de disco.

Para obter mais informações, consulte [Backup e recuperação](#) na documentação do IBM i.

IBM i Usando SAVLIB para salvar as bibliotecas do IBM MQ no IBM i

Não é possível usar o SAVLIB LIB(*ALLUSR) para salvar as bibliotecas do IBM MQ porque essas bibliotecas têm nomes que começam com Q.

É possível usar SAVLIB LIB(QM*) para salvar todas as bibliotecas do gerenciador de filas, mas somente se estiver usando um dispositivo de salvamento diferente de *SAVF. Para DEV(*SAVF), deve-se usar um comando SAVLIB para cada uma delas e todas as bibliotecas do gerenciador de filas em seu sistema.

IBM i Efetuando quiesce do IBM MQ for IBM i

Esta seção explica como colocar em modo quiesce (encerramento normal) o IBM MQ for IBM i.

Para colocar em modo quiesce o IBM MQ for IBM i:

1. Conecte-se a uma nova sessão interativa do IBM MQ for IBM i, assegurando que não esteja acessando nenhum objeto.
2. Certifique-se de:
 - Autoridade *ALLOBJ ou autoridade de gerenciamento de objetos para a biblioteca QMQM
 - Autoridade suficiente para usar o comando ENDSBS
3. Avisar todos os usuários que você pretende parar o IBM MQ for IBM i.
4. Como você, então, continua depende se você deseja encerrar (colocar em modo quiesce) um único gerenciador de filas (onde outros podem existir) (consulte [“Encerrando um único gerenciador de filas para IBM MQ for IBM i”](#) na página 454) ou todos os gerenciadores de filas (consulte [“Encerrando todos os gerenciadores de filas para IBM MQ for IBM i”](#) na página 455).
5. Encerre o servidor mqweb inserindo o comando a seguir em qshell:

```
/QIBM/ProdData/mqm/bin/endmqweb
```

Parâmetro ENDCCTJOB(*YES) de ENDMQM

O parâmetro ENDMQM ENDCCTJOB(*YES) funciona de forma diferente no IBM MQ for IBM i V6.0 e mais recente em comparação com versões anteriores.

Nas versões anteriores, quando você especifica o ENDCCTJOB(*YES), o MQ termina coercivamente os seus aplicativos.

No IBM MQ for IBM i V6.0 ou mais recente, quando você especifica o ENDCCTJOB(*YES), seus aplicativos não são terminados, mas são então desconectados do gerenciador de filas.

Se você especificar ENDCCTJOB(*YES) e tiver aplicativos que não são gravados para detectar que um gerenciador de filas está sendo encerrado, na próxima vez em que uma nova chamada MQI for emitida, a chamada será retornada com um erro MQRC_CONNECTION_BROKEN (2009).

Como alternativa ao uso de ENDCCTJOB(*YES), use o parâmetro ENDCCTJOB(*NO) e use WRKMQM opção 22 (Trabalhar com Tarefas) para terminar manualmente qualquer tarefa de aplicativo que evite um reinício do gerenciador de filas.

Use estas informações para entender os três tipos de encerramento.

Nos procedimentos que se seguem, utilizamos um nome do gerenciador de filas de amostra do QMgr1 e um nome de subsistema de amostra do SUBX. Substitua esses nomes por seus próprios valores, se necessário.

Encerramento planejado

Encerramento planejado de um gerenciador de filas no IBM i

1. Antes de encerramento, execute:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

2. Para encerrar o gerenciador de filas, execute:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Se QMgr1 não terminar, o canal ou aplicativos provavelmente estão ocupados.

3. Se for necessário encerrar QMgr1 imediatamente, execute o seguinte:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

Encerramento não planejado

1. Para encerrar o gerenciador de filas, execute:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Se QMgr1 não terminar, o canal ou aplicativos provavelmente estão ocupados.

2. Se você precisar encerrar QMgr1 imediatamente, execute o seguinte:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

Encerramento sob condições anormais

1. Para encerrar o gerenciador de filas, execute:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Se QMgr1 não terminar, continue com a etapa 3 desde que:

- QMgr1 esteja em seu próprio subsistema, ou
- É possível terminar todos os gerenciadores de filas que compartilham o mesmo subsistema como QMgr1. Use o procedimento de encerramento não planejado para todos os gerenciadores de filas desse tipo.

2. Depois de ter executado todas as etapas no procedimento para todos os gerenciadores de filas que compartilham o subsistema (SUBX em nossos exemplos), execute:

```
ENDSBS SUBX *IMMED
```

Se este comando falhar ao concluir, encerre todos os gerenciadores de filas, utilizando o procedimento de encerramento não planejado e executar um IPL em sua máquina.

Aviso: Não use ENDJOBABN para tarefas IBM MQ que falham ao terminar como resultado de ENDJOB ou ENDSBS, a menos que esteja preparado para executar um IPL em sua máquina imediatamente depois.

3. Inicie os subsistemas executando:

```
STRSBS SUBX
```

4. Encerre o gerenciador de filas imediatamente, executando:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(10)
```

5. Reinicie o gerenciador de filas executando:

```
STRMQM MQMNAME(QMgr1)
```

Se isto falhar e você:

- Reiniciou sua máquina desempenhando um IPL ou
- Tem somente um único gerenciador de filas

Limpe a memória compartilhada do IBM MQ executando:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

antes de repetir a etapa 5.

Se a reinicialização do gerenciador de filas levar mais do que alguns segundos, o IBM MQ inclui mensagens de status de forma intermitente no log da tarefa detalhando o progresso da inicialização.

Se você ainda tiver problemas para reiniciar seu gerenciador de filas, entre em contato com o suporte do IBM. Qualquer ação adicional que você pudesse tomar poderia danificar o gerenciador de filas, deixando o IBM MQ incapaz de recuperação.



Encerrando todos os gerenciadores de filas para IBM MQ for IBM i

Use estas informações para entender os três tipos de encerramento.

Os procedimentos são quase os mesmos que para um único gerenciador de filas, mas usando *ALL em vez do nome do gerenciador de filas quando possível e caso contrário, utilizando um comando repetidamente usando cada nome do gerenciador de filas no lugar. Em todos os procedimentos, utilizamos um nome do gerenciador de filas de amostra do QMgr1 e um nome de subsistema de amostra do SUBX. Substitua esses com seu próprio.

Encerramento planejado

1. Uma hora antes do encerramento, execute:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

Repita isso para cada gerenciador de filas que você deseja encerrar.

2. Para encerrar o gerenciador de filas, execute:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Repita isso para cada gerenciador de filas que você deseja encerrar; os comandos separados podem ser executados em paralelo.

Se qualquer gerenciador de filas não terminar dentro de um tempo razoável (por exemplo, 10 minutos), continue para a etapa 3.

3. Para encerrar todos os gerenciadores de filas imediatamente, execute o seguinte:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

Encerramento não planejado

1. Para encerrar um gerenciador de filas, execute:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Repita isso para cada gerenciador de filas que você deseja encerrar; os comandos separados podem ser executados em paralelo.

Se os gerenciadores de filas não terminarem, o canal ou aplicativos provavelmente estarão ocupados.

2. Se você precisar encerrar os gerenciadores de filas imediatamente, execute o seguinte:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

Encerramento sob condições anormais

1. Para encerrar os gerenciadores de filas, execute:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Repita isso para cada gerenciador de filas que você deseja encerrar; os comandos separados podem ser executados em paralelo.

2. Encerre os subsistemas (SUBX em nossos exemplos), executando:

```
ENDSBS SUBX *IMMED
```

Repita isso para cada subsistema que você deseja encerrar; os comandos separados podem ser executados em paralelo.

Se este comando falhar ao concluir, execute um IPL no seu sistema.

Aviso: Não use ENDJOBABN para tarefas que falham ao fim como resultado de ENDJOB ou ENDSBS, a menos que você está preparado para executar um IPL no sistema imediatamente após.

3. Inicie a subsistemas, executando:

```
STRSBS SUBX
```

Repita isso para cada subsistema que você deseja iniciar.

4. Encerre os gerenciadores de filas para imediatamente, executando:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. Reinicie os gerenciadores de filas executando:


```
STRMQM MQMNAME(QMgr1)
```

Repita isso para cada gerenciador de filas que você deseja iniciar.

Se a reinicialização do gerenciador de filas levar mais do que alguns segundos, o IBM MQ irá mostrar mensagens de status intermitentes detalhando o progresso de inicialização.

Se você ainda tiver problemas para reiniciar qualquer gerenciador de filas, entre em contato com o suporte do IBM. Qualquer ação adicional que é possível executar poderia danificar os gerenciadores de filas, deixando o MQSeries ou IBM MQ incapaz de se recuperar.

Administering IBM MQ for z/OS

IBM MQ for z/OS can be controlled and managed by MQSC and PCF commands, by a set of utilities and programs provided with the product, and by authorized applications.

For details of how to administer IBM MQ for z/OS and the different administrative tasks you might have to undertake, see the following links.

You can also administer IBM MQ for z/OS using the IBM MQ Explorer running in a Linux shell. For more information, see [“Administração usando o IBM MQ Explorer” on page 121](#).

Related concepts

[IBM MQ for z/OS concepts](#)

Related tasks

[“Administrando IBM MQ” on page 7](#)

Para administrar seus gerenciadores de filas e recursos associados do IBM MQ, escolha seu método preferencial dentre um conjunto de tarefas que é possível usar para ativar e gerenciar esses recursos.

[Planning your IBM MQ environment on z/OS](#)

[Configuring queue managers on z/OS](#)

Issuing queue manager commands on z/OS

You can control most of the operational environment of IBM MQ by using control commands. You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or authorized applications.

About this task

You use MQSC commands, in batch or interactive mode, to administer queue managers directly. You use PCF commands to help you create applications that administer queue managers. MQSC commands are in human-readable text form, whereas PCF commands let applications create requests and read the replies without having to parse text strings. Like MQSC commands, applications issue PCF commands by sending them as messages to the command input queue.

The following topics describe how you issue queue manager commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications.

Not all commands can be issued from all sources. See [“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS” on page 458](#).

Related tasks

[Preparing sample applications for the TSO environment on z/OS](#)

Related information

[Administering IBM MQ using MQSC commands](#)

Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS

You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications. Not all commands can be issued from all these sources.

Which MQSC and PCF commands can control each IBM MQ object

Table 1 in "Command summary for IBM MQ for z/OS" maps which MQSC and PCF commands can be used on IBM MQ for z/OS to alter, define, delete and display each IBM MQ object. See also [MQSC commands reference](#) and ["Usando os formatos de comando programável do IBM MQ"](#) on page 27.

List of sources from which commands can be issued

If you are a suitably authorized user, you can issue IBM MQ commands from the following sources:

- The z/OS console or equivalent (such as SDSF/TSO).

See also ["Using the operations and control panels on z/OS"](#) on page 471.

Note: When using the z/OS console, you need to add /cpf to the start of a command, where cpf is the command prefix for the queue manager subsystem.

- The initialization input data sets CSQINP1, CSQINP2, CSQINPT and CSQINPX.

See ["Initialization commands for IBM MQ for z/OS"](#) on page 468.

- The z/OS master get command routine, MGCRE (SVC 34).
- The IBM MQ batch utility programs such as CSQUTIL, which processes a list of commands in a sequential data set.

See ["Using the IBM MQ for z/OS utilities"](#) on page 480.

- Suitably authorized applications, sending commands as messages to the SYSTEM.COMMAND.INPUT queue.

The application can be any of the following:

- A batch region program
- A CICS application
- An IMS application
- A TSO application
- An application program or utility on another IBM MQ system

See ["Writing programs to administer IBM MQ for z/OS"](#) on page 488 and [Preparing sample applications for the TSO environment on z/OS](#).

Not all commands can be issued from all sources

Commands are classified according to where they can be issued from:

1

CSQINP1

2

CSQINP2

C

The z/OS console

R

The command server and command queue, by means of CSQUTIL, CSQINPT, CSQINPX, or authorized applications.

Within the command descriptions in [MQSC commands reference](#), these sources are identified by the use of the characters 1, 2, C, and R in each command description. Table 2 in "[Command summary for IBM MQ for z/OS](#)" summarizes the MQSC commands and the sources from which they can be issued.

Related tasks

[Preparing sample applications for the TSO environment on z/OS](#)

Related information

[Administering IBM MQ using MQSC commands](#)

Command summary for IBM MQ for z/OS

A summary of the main MQSC and PCF commands, and of the sources from which you can run MQSC commands on IBM MQ for z/OS.

Table 25 on page 459 maps which MQSC and PCF commands can be used on IBM MQ for z/OS to alter, define, delete and display each IBM MQ object.

<i>Table 25. Summary of the main MQSC and PCF commands by object type</i>				
MQSC command	ALTER	DEFINE	DISPLAY	DELETE
PCF command	Change	Create/Copy	Inquire	Delete
AUTHINFO	X	X	X	X
CFSTATUS			X	
CFSTRUCT	X	X	X	X
CHANNEL	X	X	X	X
CHSTATUS			X	
NAMELIST	X	X	X	X
PROCESS	X	X	X	X
QALIAS	M	M	M	M
QCLUSTER			M	
QLOCAL	M	M	M	M
QMGR	X		X	
QMODEL	M	M	M	M
QREMOTE	M	M	M	M
QUEUE	P	P	X	P
QSTATUS			X	
STGCLASS	X	X	X	X

Key to table symbols:

- M = MQSC only
- P = PCF only
- X = both

There are many other MQSC and PCF commands which allow you to manage other IBM MQ resources, and carry out other actions in addition to those summarized in [Table 25 on page 459](#).

[Table 26 on page 460](#) shows every MQSC command, and where each command can be issued from.

- CSQINP1 initialization input data set
- CSQINP2 initialization input data set
- z/OS console (or equivalent)
- SYSTEM.COMMAND.INPUT queue and command server (from applications, CSQUTIL, or the CSQINPX initialization input data set)

Table 26. Sources from which to run MQSC commands

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
ALTER AUTHINFO		X	X	X
ALTER BUFFPOOL		X	X	X
ALTER CFSTRUCT		X	X	X
ALTER CHANNEL		X	X	X
ALTER NAMELIST		X	X	X
ALTER PROCESS		X	X	X
ALTER PSID			X	X
ALTER QALIAS		X	X	X
ALTER QLOCAL		X	X	X
ALTER QMGR		X	X	X
ALTER QMODEL		X	X	X
ALTER QREMOTE		X	X	X
ALTER SECURITY	X	X	X	X
ALTER SMDS		X	X	X
ALTER STGCLASS		X	X	X
ALTER SUB			X	X
ALTER TOPIC		X	X	X
ALTER TRACE	X	X	X	X
ARCHIVE LOG	X	X	X	X
BACKUP CFSTRUCT			X	X
CLEAR QLOCAL		X	X	X
CLEAR TOPICSTR			X	X
DEFINE AUTHINFO		X	X	X
DEFINE BUFFPOOL	X			
DEFINE CFSTRUCT		X	X	X
DEFINE CHANNEL		X	X	X
DEFINE LOG			X	X
DEFINE MAXSMGS		X	X	X
DEFINE NAMELIST		X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
DEFINE PROCESS		X	X	X
DEFINE PSID	X		X	X
DEFINE QALIAS		X	X	X
DEFINE QLOCAL		X	X	X
DEFINE QMODEL		X	X	X
DEFINE QREMOTE		X	X	X
DEFINE STGCLASS		X	X	X
DEFINE SUB			X	X
DEFINE TOPIC		X	X	X
DELETE AUTHINFO		X	X	X
DELETE BUFFPOOL		X	X	X
DELETE CFSTRUCT		X	X	X
DELETE CHANNEL			X	X
DELETE NAMELIST		X	X	X
DELETE PROCESS		X	X	X
DELETE PSID			X	X
DELETE QALIAS		X	X	X
DELETE QLOCAL		X	X	X
DELETE QMODEL		X	X	X
DELETE QREMOTE		X	X	X
DELETE STGCLASS		X	X	X
DELETE SUB			X	X
DELETE TOPIC		X	X	X
DISPLAY ARCHIVE	X	X	X	X
DISPLAY AUTHINFO		X	X	X
DISPLAY CFSTATUS			X	X
DISPLAY CFSTRUCT		X	X	X
DISPLAY CHANNEL		X	X	X
DISPLAY CHINIT			X	X
DISPLAY CHLAUTH		X	X	X
DISPLAY CHSTATUS			X	X
DISPLAY CLUSQMGR			X	X
DISPLAY CMDSERV	X	X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
DISPLAY CONN		X	X	X
DISPLAY GROUP		X	X	X
DISPLAY LOG	X	X	X	X
DISPLAY MAXSMSGS		X	X	X
DISPLAY NAMELIST		X	X	X
DISPLAY PROCESS		X	X	X
DISPLAY PUBSUB		X	X	X
DISPLAY QALIAS		X	X	X
DISPLAY QCLUSTER		X	X	X
DISPLAY QLOCAL		X	X	X
DISPLAY QMGR		X	X	X
DISPLAY QMODEL		X	X	X
DISPLAY QREMOTE		X	X	X
DISPLAY QSTATUS		X	X	X
DISPLAY QUEUE		X	X	X
DISPLAY SBSTATUS			X	X
DISPLAY SECURITY			X	X
DISPLAY SMDS		X	X	X
DISPLAY SMDSCONN		X	X	X
DISPLAY STGCLASS		X	X	X
DISPLAY SUB			X	X
DISPLAY SYSTEM	X	X	X	X
DISPLAY TCLUSTER		X	X	X
DISPLAY THREAD		X	X	X
DISPLAY TOPIC		X	X	X
DISPLAY TPSTATUS		X	X	X
DISPLAY TRACE	X	X	X	X
DISPLAY USAGE		X	X	X
MOVE QLOCAL		X	X	X
PING CHANNEL			X	X
RECOVER BSDS			X	X
RECOVER CFSTRUCT			X	X
REFRESH CLUSTER			X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
REFRESH QMGR		X	X	X
REFRESH SECURITY			X	X
RESET CFSTRUCT			X	X
RESET CHANNEL			X	X
RESET CLUSTER			X	X
RESET QMGR		X	X	X
RESET QSTATS		X	X	X
RESET SMDS			X	X
RESET TPIPE			X	X
RESOLVE CHANNEL			X	X
RESOLVE INDOUBT		X	X	X
RESUME QMGR			X	X
RVERIFY SECURITY		X	X	X
SET ARCHIVE	X	X	X	X
SET CHLAUTH		X	X	X
SET LOG	X	X	X	X
SET SYSTEM	X	X	X	X
START CHANNEL			X	X
START CHINIT		X	X	X
START CMDSERV	X	X	X	
START LISTENER			X	X
START QMGR			X	
START SMDSCONN		X	X	X
START TRACE	X	X	X	X
STOP CHANNEL			X	X
STOP CHINIT			X	X
STOP CMDSERV	X	X	X	
STOP LISTENER			X	X
STOP QMGR			X	X
STOP SMDSCONN		X	X	X
STOP TRACE	X	X	X	X
SUSPEND QMGR			X	X

In [MQSC commands](#), each command description identifies the sources from which that command can be run.

Using MQSC to start and stop a queue manager on z/OS

An introduction to using control commands on IBM MQ for z/OS: After you have installed IBM MQ, use MQSC commands to start and stop a queue manager.

Before you begin

After you have installed IBM MQ, it is defined as a formal z/OS subsystem. This message appears during any initial program load (IPL) of z/OS:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

where *ssnm* is the IBM MQ subsystem name.

From now on, you can start the queue manager for that subsystem *from any z/OS console that has been authorized to issue system control commands*; that is, a z/OS SYS command group. You must issue the START command from the authorized console, you cannot issue it through JES or TSO.

If you are using queue sharing groups, you must start RRS first, and then Db2®, before you start the queue manager.

About this task

When a queue manager stops under normal conditions, its last action is to take a termination checkpoint. This checkpoint, and the logs, give the queue manager the information it needs to restart.

The following steps contain information about the START and STOP commands, and contain a brief overview of start-up after an abnormal termination has occurred.

Procedure

1. Start a queue manager

You start a queue manager by issuing a START QMGR command. However, you cannot successfully use the START command unless you have appropriate authority. See the [Setting up security on z/OS](#) for information about IBM MQ security. The following code shows examples of the START command. Note that you must prefix an MQSC command with a command prefix string (CPF).

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

See [START QMGR](#) for information about the syntax of the START QMGR command.

You cannot run the queue manager as a batch job or start it using a z/OS command START. These methods are likely to start an address space for IBM MQ that then ends abnormally. Nor can you start a queue manager from the CSQUTIL utility program or a similar user application.

You can, however, start a queue manager from an APF-authorized program by passing a START QMGR command to the z/OS MGCRC (SVC 34) service.

If you are using queue sharing groups, the associated Db2 systems and RRS must be active when you start the queue manager.

Start options

When you start a queue manager, a system parameter module is loaded. You can specify the name of the system parameter module in one of two ways:

- With the PARM parameter of the /cpf START QMGR command, for example


```
/cpf START QMGR PARM(CSQ1ZPRM)
```

- With a parameter in the startup procedure, for example, code the JCL EXEC statement as

```
//MQM EXEC PGM=CSQYASCP,PARM='ZPARM(CSQ1ZPRM)'
```

A system parameter module provides information specified when the queue manager was customized.

You can use the **QMGRPROD** option to specify the product against which the queue manager usage is to be recorded, and the **AMSPROD** option to specify the equivalent for AMS if that is used. See the MQSC [START QMGR](#) command for details of the permitted values.

An example JCL EXEC statement follows:

```
//MQM EXEC PGM=CSQYASCP,PARM='QMGRPROD(MQ)'
```

See [z/OS MVS Product Management](#) for more information on measured usage and product registration.

You can also use the ENVPARM option to substitute one or more parameters in the JCL procedure for the queue manager.

For example, you can update your queue manager startup procedure, so that the **DDname** CSQINP2 is a variable. This means that you can change the CSQINP2 **DDname** without changing the startup procedure. This is useful for implementing changes, providing back-outs for operators, and queue manager operations.

Suppose your start-up procedure for queue manager CSQ1 looked like this:

```
//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=th1qual.SCSQANLE
// DD DISP=SHR,DSN=th1qual.SCSQAUTH
// DD DISP=SHR,DSN=db2qual.SDSNLOAD
//BSDS1 DD DISP=SHR,DSN=myqual.BSDS01
//BSDS2 DD DISP=SHR,DSN=myqual.BSDS02
//CSQP0000 DD DISP=SHR,DSN=myqual.PSID00
//CSQP0001 DD DISP=SHR,DSN=myqual.PSID01
//CSQP0002 DD DISP=SHR,DSN=myqual.PSID02
//CSQP0003 DD DISP=SHR,DSN=myqual.PSID03
//CSQINP1 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1INP1)
//CSQINP2 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1&INP2.)
//CSQOUT1 DD SYSOUT=*
//CSQOUT2 DD SYSOUT=*
```

If you then start your queue manager with the following command:

```
+CSQ1 START QMGR
```

then the CSQINP2 used is a member called CSQ1NORM.

However, suppose you are putting a new suite of programs into production so that the next time you start queue manager CSQ1, the CSQINP2 definitions are to be taken from member CSQ1NEW. To do this, you would start the queue manager with this command:

```
+CSQ1 START QMGR ENVPARM('INP2=NEW')
```

and CSQ1NEW would be used instead of CSQ1NORM. Note: z/OS limits the KEYWORD=value specifications for symbolic parameters (as in INP2=NEW) to 255 characters.

Starting after an abnormal termination

IBM MQ automatically detects whether restart follows a normal shutdown or an abnormal termination.

Starting a queue manager after it ends abnormally is different from starting it after the STOP QMGR command has been issued. After STOP QMGR, the system finishes its work in an orderly way and takes a termination checkpoint before stopping. When you restart the queue manager, it uses information from the system checkpoint and recovery log to determine the system status at shutdown.

However, if the queue manager ends abnormally, it terminates without being able to finish its work or take a termination checkpoint. When you restart a queue manager after an abend, it refreshes its knowledge of its status at termination using information in the log, and notifies you of the status of various tasks. Normally, the restart process resolves all inconsistent states. But, in some cases, you must take specific steps to resolve inconsistencies.

User messages on start-up

When you start a queue manager successfully, the queue manager produces a set of startup messages.

2. Stop a queue manager.

Before stopping a queue manager, all IBM MQ-related write-to-operator-with-reply (WTOR) messages must receive replies, for example, getting log requests. Each of the following commands terminates a running queue manager.

```
+CSQ1  STOP QMGR
+CSQ1  STOP QMGR MODE(QUIESCE)
+CSQ1  STOP QMGR MODE(FORCE)
+CSQ1  STOP QMGR MODE(RESTART)
```

The command STOP QMGR defaults to STOP QMGR MODE(QUIESCE).

In QUIESCE mode, IBM MQ does not allow any new connection threads to be created, but allows existing threads to continue; it terminates only when all threads have ended. Applications can request to be notified in the event of the queue manager quiescing. Therefore, use the QUIESCE mode where possible so that applications that have requested notification have the opportunity to disconnect. See [What happens during termination](#) for details.

If the queue manager does not terminate in a reasonable time in response to a STOP QMGR MODE(QUIESCE) command, use the DISPLAY CONN command to determine whether any connection threads exist, and take the necessary steps to terminate the associated applications. If there are no threads, issue a STOP QMGR MODE(FORCE) command.

The STOP QMGR MODE(QUIESCE) and STOP QMGR MODE(FORCE) commands deregister IBM MQ from the MVS Automatic Restart Manager (ARM), preventing ARM from restarting the queue manager automatically. The STOP QMGR MODE(RESTART) command works in the same way as the STOP QMGR MODE(FORCE) command, except that it does not deregister IBM MQ from ARM. This means that the queue manager is eligible for immediate automatic restart.

If the IBM MQ subsystem is not registered with ARM, the STOP QMGR MODE(RESTART) command is rejected and the following message is sent to the z/OS console:

```
CSQY205I ARM element arm-element is not registered
```

If this message is not issued, the queue manager is restarted automatically. For more information about ARM, see [“Using the z/OS Automatic Restart Manager \(ARM\)”](#) on page 546.

Only cancel the queue manager address space if STOP QMGR MODE(FORCE) does not terminate the queue manager.

If a queue manager is stopped by either canceling the address space or by using the command STOP QMGR MODE(FORCE), consistency is maintained with connected CICS or IMS systems. Resynchronization of resources is started when a queue manager restarts and is completed when the connection to the CICS or IMS system is established.

Note: When you stop your queue manager, you might find message IEF352I is issued. z/OS issues this message if it detects that failing to mark the address space as unusable would lead to an integrity exposure. You can ignore this message.

Stop messages

After issuing a STOP QMGR command, you get the messages CSQY009I and CSQY002I, for example:

```
CSQY009I +CSQ1 ' STOP QMGR' COMMAND ACCEPTED FROM
USER(userid), STOP MODE(FORCE)
CSQY002I +CSQ1 QUEUE MANAGER STOPPING
```

Where `userid` is the user ID that issued the STOP QMGR command, and the MODE parameter depends on that specified in the command.

When the STOP command has completed successfully, the following messages are displayed on the z/OS console:

```
CSQ9022I +CSQ1 CSQYASCP ' STOP QMGR' NORMAL COMPLETION
CSQ3104I +CSQ1 CSQ3EC0X - TERMINATION COMPLETE
```

If you are using ARM, and you did not specify MODE(RESTART), the following message is also displayed:

```
CSQY204I +CSQ1 ARM DEREGISTER for element arm-element type
arm-element-type successful
```

You cannot restart the queue manager until the following message has been displayed:

```
CSQ3100I +CSQ1 CSQ3EC0X - SUBSYSTEM ssnm READY FOR START COMMAND
```

z/OS Issuing commands from a z/OS console or equivalent

You can issue IBM MQ MQSC and PCF commands from a z/OS console or its equivalent. You can also issue IBM MQ commands from anywhere where you can issue z/OS commands, such as SDSF or by a program using the MGCRC macro. When using the z/OS console, you add /cpf to the start of a command.

Before you begin

Not all commands can be issued by the z/OS console. Within the command description topics (the children of [MQSC commands reference](#)), each command that can be issued by the console is identified by the character 'C'. Table 2 in "[Command summary for IBM MQ for z/OS](#)" summarizes the MQSC commands and the sources from which they can be issued.

You cannot issue IBM MQ commands using the IMS/SSR command format from an IMS terminal. This function is not supported by the IMS adapter.

The input field provided by SDSF might not be long enough for some commands, particularly those commands for channels.

The maximum amount of data that can be displayed as a result of a command typed in at the console is 32 KB.

About this task

If you are a suitably authorized user, you can issue IBM MQ commands from the z/OS console or equivalent (such as SDSF/TSO).

When using the z/OS console, you need to add /cpf to the start of a command, where cpf is the command prefix for the queue manager subsystem.

The following steps refer to commands and attributes using their MQSC command names rather than their PCF names.

Procedure

- Use command prefix strings

Each IBM MQ command must be prefixed with a command prefix string (CPF).

Because more than one IBM MQ subsystem can run under z/OS, the CPF is used to indicate which IBM MQ subsystem processes the command.

For example, to start the queue manager for a subsystem called CSQ1, where CPF is ' +CSQ1 ', you issue the following command from the operator console:

```
+CSQ1 START QMGR
```

This CPF must be defined in the subsystem name table (for the subsystem CSQ1), as described in [Defining command prefix strings \(CPFs\)](#). In the examples, the string +CSQ1 is used as the command prefix.

- Use the z/OS console to issue commands

You can type simple commands from the z/OS console, for example:

```
+CSQ1 DISPLAY QUEUE(TRANSMIT.QUEUE.PROD) TYPE(QLOCAL)
```

However, for complex commands or for sets of commands that you issue frequently, the other methods of issuing commands are better.

- Receive command responses

Direct responses to commands are sent to the console that issued the command. IBM MQ supports the *Extended Console Support* (EMCS) function available in z/OS, and therefore consoles with 4 byte IDs can be used. Additionally, all commands except START QMGR and STOP QMGR support the use of Command and Response Tokens (CARTs) when the command is issued by a program using the MGCRCRE macro.

Related tasks

[“Using the operations and control panels on z/OS” on page 471](#)

You use these panels for defining, displaying, altering, or deleting IBM MQ objects. Use the panels for day-to-day administration and for making small changes to objects.

[Preparing sample applications for the TSO environment on z/OS](#)

Initialization commands for IBM MQ for z/OS

Initialization commands can be used to control the queue manager startup.

Commands in the initialization input data sets are processed when IBM MQ is initialized on queue manager startup. Three types of command can be issued from the initialization input data sets:

- Commands to define IBM MQ entities that cannot be defined elsewhere, for example DEFINE BUFFPOOL.

These commands must reside in the data set identified by the DD name CSQINP1. They are processed before the restart phase of initialization. They cannot be issued through the console, operations and control panels, or an application program. The responses to these commands are written to the sequential data set that you refer to in the CSQOUT1 statement of the started task procedure.

- Commands to define IBM MQ objects that are recoverable after restart. These definitions must be specified in the data set identified by the DD name CSQINP2. They are stored in page set zero. CSQINP2 is processed after the restart phase of initialization. The responses to these commands are written to the sequential data set that you refer to in the CSQOUT2 statement of the started task procedure.
- Commands to manipulate IBM MQ objects. These commands must also be specified in the data set identified by the DD name CSQINP2. For example, the IBM MQ-supplied sample contains an ALTER QMGR command to specify a dead-letter queue for the subsystem. The response to these commands is written to the CSQOUT2 output data set.

Note: If IBM MQ objects are defined in CSQINP2, IBM MQ attempts to redefine them each time the queue manager is started. If the objects already exist, the attempt to define them fails. If you need to define your objects in CSQINP2, you can avoid this problem by using the REPLACE parameter of the DEFINE commands, however, this overrides any changes that were made during the previous run of the queue manager.

Sample initialization data set members are supplied with IBM MQ for z/OS. They are described in [Sample definitions supplied with IBM MQ](#).

Initialization commands for distributed queuing

You can also use the CSQINP2 initialization data set for the START CHINIT command. If you need a series of other commands to define your distributed queuing environment (for example, starting listeners), IBM MQ provides a third initialization input data set, called CSQINPX, that is processed as part of the channel initiator started task procedure.

The MQSC commands contained in the data set are executed at the end of channel initiator initialization, and output is written to the data set specified by the CSQOUTX DD statement. You might use the CSQINPX initialization data set to start listeners for example.

A sample channel initiator initialization data set member is supplied with IBM MQ for z/OS. It is described in [Sample definitions supplied with IBM MQ](#).

Initialization commands for publish/Subscribe

If you need a series of commands to define your publish/subscribe environment (for example, when defining subscriptions), IBM MQ provides a fourth initialization input data set, called CSQINPT.

The MQSC commands contained in the data set are executed at the end of publish/subscribe initialization, and output is written to the data set specified by the CSQOUTT DD statement. You might use the CSQINPT initialization data set to define subscriptions for example.

A sample publish/subscribe initialization data set member is supplied with IBM MQ for z/OS. It is described in [Sample definitions supplied with IBM MQ](#).

Private and global definitions on IBM MQ for z/OS

When you define an object on IBM MQ for z/OS, you can choose whether you want to share that definition with other queue managers (a *global* definition), or whether the object definition is to be used by one queue manager only (a *private* definition). This is called the object *disposition*.

Global definition

If your queue manager belongs to a queue sharing group, you can choose to share any object definitions you make with the other members of the group. This means that you have to define an object once only, reducing the total number of definitions required for the whole system.

Global object definitions are held in a *shared repository* (a Db2 shared database), and are available to all the queue managers in the queue sharing group. These objects have a disposition of GROUP.

Private definition

If you want to create an object definition that is required by one queue manager only, or if your queue manager is not a member of a queue sharing group, you can create object definitions that are not shared with other members of a queue sharing group.

Private object definitions are held on page set zero of the defining queue manager. These objects have a disposition of QMGR.

You can create private definitions for all types of IBM MQ objects except CF structures (that is, channels, namelists, process definitions, queues, queue managers, storage class definitions, and authentication information objects), and global definitions for all types of objects except queue managers.

IBM MQ automatically copies the definition of a group object to page set zero of each queue manager that uses it. You can alter the copy of the definition temporarily if you want, and IBM MQ allows you to refresh the page set copies from the repository copy if required.

IBM MQ always tries to refresh the page set copies from the repository copy at startup (for channel commands, this is done when the channel initiator restarts), or if the group object is changed.

Note: The copy of the definition is refreshed from the definition of the group, only if the definition of the group has changed after you created the copy of the definition.

This ensures that the page set copies reflect the version on the repository, including any changes that were made when the queue manager was inactive. The copies are refreshed by generating DEFINE REPLACE commands, therefore there are circumstances under which the refresh is not performed, for example:

- If a copy of the queue is open, a refresh that changes the usage of the queue fails.
- If a copy of a queue has messages on it, a refresh that deletes that queue fails.
- If a copy of a queue would require ALTER with FORCE to change it.

In these circumstances, the refresh is not performed on that copy, but is performed on the copies on all other queue managers.

If the queue manager is shut down and then restarted stand-alone, any local copies of objects are deleted, unless for example, the queue has associated messages.

There is a third object disposition that applies to local queues only. This allows you to create shared queues. The definition for a shared queue is held on the shared repository and is available to all the queue managers in the queue sharing group. In addition, the messages on a shared queue are also available to all the queue managers in the queue sharing group. This is described in [Shared queues and queue sharing groups](#). Shared queues have an object disposition of SHARED.

The following table summarizes the effect of the object disposition options for queue managers started stand-alone, and as a member of a queue sharing group.

Disposition	Stand-alone queue manager	Member of a queue sharing group
QMGR	Object definition held on page set zero.	Object definition held on page set zero.
GROUP	Not allowed.	Object definition held in the shared repository. Local copy held on page set zero of each queue manager in the group.
SHARED	Not allowed.	Queue definition held in the shared repository. Messages available to any queue manager in the group.

Manipulating global definitions

If you want to change the definition of an object that is held in the shared repository, you need to specify whether you want to change the version on the repository, or the local copy on page set zero. Use the object disposition as part of the command to do this.

▶ z/OS Directing commands to different queue managers on z/OS

You can use the *command scope* to control on which queue manager the command runs.

You can choose to execute a command on the queue manager where it is entered, or on a different queue manager in the queue sharing group. You can also choose to issue a particular command in parallel on all the queue managers in a queue sharing group. This is possible for both MQSC commands and PCF commands.

This is determined by the *command scope*. The command scope is used with the object disposition to determine which version of an object you want to work with.

For example, you might want to alter some of the attributes of an object, the definition of which is held in the shared repository.

- You might want to change the version on one queue manager only, and not make any changes to the version on the repository or those in use by other queue managers.
- You might want to change the version in the shared repository for future users, but leave existing copies unchanged.
- You might want to change the version in the shared repository, but also want your changes to be reflected immediately on all the queue managers in the queue sharing group that hold a copy of the object on their page set zero.

Use the command scope to specify whether the command is executed on this queue manager, another queue manager, or all queue managers. Use the object disposition to specify whether the object you are manipulating is in the shared repository (a group object), or is a local copy on page set zero (a queue manager object).

You do not have to specify the command scope and object disposition to work with a shared queue because every queue manager in the queue sharing group handles the shared queue as a single queue.

▶ z/OS Using the operations and control panels on z/OS

You use these panels for defining, displaying, altering, or deleting IBM MQ objects. Use the panels for day-to-day administration and for making small changes to objects.

Before you begin

The IBM MQ for z/OS operations and controls panels (CSQOREXX) might not support all new function and parameters added from version 7 onwards. For example, there are no panels for the direct manipulation of topic objects or subscriptions. Use one of the following supported mechanisms to administer publish/subscribe definitions and other system controls that are not directly available from other panels:

1. IBM MQ Explorer
2. z/OS console
3. Programmable Command Format (PCF) messages
4. COMMAND function of CSQUTIL
5. IBM MQ Console

Note that the generic **Command** action in the CSQOREXX panels allows you to issue any valid MQSC command, including SMDS related commands. You can use all the commands that the COMMAND function of CSQUTIL issues.

You cannot issue the IBM MQ commands directly from the command line in the panels.

To use the operations and control panels, you must have the correct security authorization; this is described in the [User IDs for command security and command resource security](#).

You cannot provide a user ID and password using CSQUTIL, or the CSQOREXX panels. Instead, if you user ID has UPDATE authority to the BATCH profile in MQCONN, you can bypass the **CHKLOCL**(REQUIRED) setting. See [Using **CHKLOCL** on locally bound applications](#) for more information.

If you are setting up or changing many objects, use the COMMAND function of the CSQUTIL utility program. See [“Using the CSQUTIL utility for IBM MQ for z/OS”](#) on page 481.

About this task

The operations and control panels support the controls for the channel initiator (for example, to start a channel or a TCP/IP listener), for clustering, and for security. They also enable you to display information about threads and page set usage.

The panels work by sending MQSC type IBM MQ commands to a queue manager, through the system command input queue.

Example

This is the panel that displays when you start a panel session:

```
IBM MQ for z/OS - Main Menu
Complete fields. Then press Enter.
Action . . . . . 1      0. List with filter  4. Manage
                        1. List or Display  5. Perform
                        2. Define like    6. Start
                        3. Alter          7. Stop
                        8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A  Q=Qmgr, C=Copy, P=Private, G=Group,
                        S=Shared, A=All

Connect name . . . . . MQ1C - local queue manager or group
Target queue manager . . . . MQ1C
                        - connected or remote queue manager for command input
Action queue manager . . . . MQ1C - command scope in group
Response wait time . . . . 30  5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2024. All rights reserved.

Command ==>
F1=Help      F2=Split    F3=Exit     F4=Prompt   F9=SwapNext F10=Messages
F12=Cancel
```

From this panel you can perform actions such as these:

- Choose the local queue manager you want and whether you want the commands issued on that queue manager, on a remote queue manager, or on another queue manager in the same queue sharing group as the local queue manager. Over type the queue manager name if you need to change it.
- Select the action you want to perform by typing in the appropriate number in the **Action** field.
- Specify the object type that you want to work with. Press function key F1 for help about the object types if you are not sure what they are.
- Specify the disposition of the object type that you want to work with.
- Display a list of objects of the type specified. Type in an asterisk (*) in the **Name** field and press **Enter** to display a list of objects (of the type specified) that have already been defined on the action queue manager. You can then select one or more objects to work with in sequence. All the actions are available from the list.

Note: You are recommended to make choices that result in a list of objects being displayed, and then work from that list. Use the **Display** action, because that is allowed for all object types.

Invocation and rules for the operations and control panels

You can control IBM MQ and issue control commands through the ISPF panels.

How to access the IBM MQ operations and control panels

If the ISPF/PDF primary options menu has been updated for IBM MQ, you can access the IBM MQ operations and control panels from that menu. For details about updating the menu, see the [Task 20: Set up the operations and control panels](#).

You can access the IBM MQ operations and control panels from the TSO command processor panel (typically option 6 on the ISPF/PDF primary options menu). The name of the exec that you run to do this is CSQOREXX. It has two parameters; `th1qual` is the high-level qualifier for the IBM MQ libraries to be used, and `langletter` is the letter identifying the national language libraries to be used (for example, E for U.S. English). The parameters can be omitted if the IBM MQ libraries are permanently installed in your ISPF setup. Alternatively, you can issue CSQOREXX from the TSO command line.

These panels are designed to be used by operators and administrators with a minimum of formal training. Read these instructions with the panels running and try out the different tasks suggested.

Note: While using the panels, temporary dynamic queues with names of the form SYSTEM.CSQOREXX.* are created.

Rules for the operations and control panels

See [Rules for naming IBM MQ objects](#) about the general rules for IBM MQ character strings and names. However, there are some rules that apply only to the operations and control panels:

- Do not enclose strings, for example descriptions, in single or double quotation marks.
- If you include an apostrophe or quotation mark in a text field, you do not have to repeat it or add an escape character. The characters are saved exactly as you type them; for example:

```
This is Maria's queue
```

The panel processor doubles them for you to pass them to IBM MQ. However, if it has to truncate your data to do this, it does so.

- You can use uppercase or lowercase characters in most fields, and they are folded to uppercase characters when you press Enter. The exceptions are:
 - Storage class names and coupling facility structure names, which must start with uppercase A through Z and be followed by uppercase A through Z or numeric characters.
 - Certain fields that are not translated. These include:
 - Application ID
 - Description
 - Environment data
 - Object names (but if you use a lowercase object name, you might not be able to enter it at a z/OS console)
 - Remote system name
 - Trigger data
 - User data
- In names, leading blanks and leading underscores are ignored. Therefore, you cannot have object names beginning with blanks or underscores.
- Underscores are used to show the extent of blank fields. When you press Enter, trailing underscores are replaced by blanks.
- Many description and text fields are presented in multiple parts, each part being handled by IBM MQ independently. This means that trailing blanks are retained and the text is not contiguous.

Blank fields

When you specify the **Define** action for an IBM MQ object, each field on the define panel contains a value. See the general help (extended help) for the display panels for information about where IBM

MQ gets the values. If you type over a field with blanks, and blanks are not allowed, IBM MQ puts the installation default value in the field or prompts you to enter the required value.

When you specify the **Alter** action for an IBM MQ object, each field on the alter panel contains the current value for that field. If you type over a field with blanks, and blanks are not allowed, the value of that field is unchanged.

Objects and actions on z/OS

The operations and control panels offer you many different types of object and a number of actions that you can perform on them.

The actions are listed on the initial panel and enable you to manipulate the objects and display information about them. These objects include all the IBM MQ objects, together with some extra ones. The objects fall into the following categories.

- [Queues, processes, authentication information objects, namelists, storage classes and CF structures](#)
- [Channels](#)
- [Cluster objects](#)
- [Queue manager and security](#)
- [Connections](#)
- [System](#)

Refer to [Actions](#) for a cross-reference table of the actions which can be taken with the IBM MQ objects.

Queues, processes, authentication information objects, namelists, storage classes and CF structures

These are the basic IBM MQ objects. There can be many of each type. They can be listed, listed with filter, defined, and deleted, and have attributes that can be displayed and altered, using the LIST or DISPLAY, LIST with FILTER, DEFINE LIKE, MANAGE, and ALTER actions. (Objects are deleted using the MANAGE action.)

This category consists of the following objects:

QLOCAL	Local queue
QREMOTE	Remote queue
QALIAS	Alias queue for indirect reference to a queue
QMODEL	Model queue for defining queues dynamically
QUEUE	Any type of queue
QSTATUS	Status of a local queue
PROCESS	Information about an application to be started when a trigger event occurs
AUTHINFO	Authentication information: definitions required to perform Certificate Revocation List (CRL) checking using LDAP servers
NAMELIST	List of names, such as queues or clusters
STGCLASS	Storage class
CFSTRUCT	coupling facility (CF) structure
CFSTATUS	Status of a CF structure

Channels

Channels are used for distributed queuing. There can be many of each type, and they can be listed, listed with filter, defined, deleted, displayed, and altered. They also have other functions available

using the START, STOP and PERFORM actions. PERFORM provides reset, ping, and resolve channel functions.

This category consists of the following objects:

CHANNEL	Any type of channel
SENDER	Sender channel
SERVER	Server channel
RECEIVER	Receiver channel
REQUESTER	Requester channel
CLUSRCVR	Cluster-receiver channel
CLUSDR	Cluster-sender channel
SVRCONN	Server-connection channel
CLNTCONN	Client-connection channel
CHSTATUS	Status of a channel connection

Cluster objects

Cluster objects are created automatically for queues and channels that belong to a cluster. The base queue and channel definitions can be on another queue manager. There can be many of each type, and names can be duplicated. They can be listed, listed with filter, and displayed. PERFORM, START, and STOP are also available through the LIST actions.

This category consists of the following objects:

CLUSQ	Cluster queue, created for a queue that belongs to a cluster
CLUSCHL	Cluster channel, created for a channel that belongs to a cluster
CLUSQMGR	Cluster queue manager, the same as a cluster channel but identified by its queue manager name

Cluster channels and cluster queue managers do have the PERFORM, START and STOP actions, but only indirectly through the DISPLAY action.

Queue manager and security

Queue manager and security objects have a single instance. They can be listed, and have attributes that can be displayed and altered (using the LIST or DISPLAY, and ALTER actions), and have other functions available using the PERFORM action.

This category consists of the following objects:

MANAGER	Queue manager: the PERFORM action provides suspend and resume cluster functions
SECURITY	Security functions: the PERFORM action provides refresh and reverify functions

Connection

Connections can be listed, listed with filter and displayed.

This category consists only of the connection object, CONNECT.

System

A collection of other functions. This category consists of the following objects:

SYSTEM	System functions
CONTROL	Synonym for SYSTEM

The functions available are:

- LIST or DISPLAY Display queue sharing group, distributed queuing, page set, or data set usage information.
- PERFORM Refresh or reset clustering
- START Start the channel initiator or listeners
- STOP Stop the channel initiator or listeners

Actions

The actions that you can perform for each type of object are shown in the following table:

<i>Table 27. Valid operations and control panel actions for IBM MQ objects</i>								
Object	Alter	Define like	Manage (1)	List or Display	List with Filter	Perform	Start	Stop
AUTHINFO	X	X	X	X	X			
CFSTATUS				X				
CFSTRUCT	X	X	X	X	X			
CHANNEL	X	X	X	X	X	X	X	X
CHSTATUS				X	X			
CLNTCONN	X	X	X	X	X			
CLUSCHL				X	X	X(2)	X(2)	X(2)
CLUSQ				X	X			
CLUSQMGR				X	X	X(2)	X(2)	X(2)
CLUSRCVR	X	X	X	X	X	X	X	X
CLUSDR	X	X	X	X	X	X	X	X
CONNECT				X	X			
CONTROL				X		X	X	X
MANAGER	X			X		X		
NAMELIST	X	X	X	X	X			
PROCESS	X	X	X	X	X			
QALIAS	X	X	X	X	X			
QLOCAL	X	X	X	X	X			
QMODEL	X	X	X	X	X			
QREMOTE	X	X	X	X	X			
QSTATUS				X	X			
QUEUE	X	X	X	X	X			
RECEIVER	X	X	X	X	X	X	X	X
REQUESTER	X	X	X	X	X	X	X	X
SECURITY	X			X		X		

Table 27. Valid operations and control panel actions for IBM MQ objects (continued)

Object	Alter	Define like	Manage (1)	List or Display	List with Filter	Perform	Start	Stop
SENDER	X	X	X	X	X	X	X	X
SERVER	X	X	X	X	X	X	X	X
SVRCONN	X	X	X	X	X		X	X
STGCLASS	X	X	X	X	X			
SYSTEM				X		X	X	X

Note:

1. Provides Delete and other functions
2. Using the List or Display action

z/OS Object dispositions on z/OS

You can specify the *disposition* of the object with which you need to work. The disposition signifies where the object **definition** is kept, and how the object behaves.

The disposition is significant only if you are working with any of the following object types:

- queues
- channels
- processes
- namelists
- storage classes
- authentication information objects

If you are working with other object types, the disposition is disregarded.

Permitted values are:

Q

QMGR. The object definitions are on the page set of the queue manager and are accessible only by the queue manager.

C

COPY. The object definitions are on the page set of the queue manager and are accessible only by the queue manager. They are local copies of objects defined as having a disposition of GROUP.

P

PRIVATE. The object definitions are on the page set of the queue manager and are accessible only by the queue manager. The objects have been defined as having a disposition of QMGR or COPY.

G

GROUP. The object definitions are in the shared repository, and are accessible by all queue managers in the queue sharing group.

S

SHARED. This disposition applies only to local queues. The queue definitions are in the shared repository, and are accessible by all queue managers in the queue sharing group.

A

ALL. If the action queue manager is either the target queue manager, or *, objects of **all** dispositions are included; otherwise, objects of QMGR and COPY dispositions only are included. This is the default.

Selecting a queue manager, defaults, and levels using the ISPF control panel on z/OS

You can use the CSQOREXX exec in ISPF to control your queue managers.

While you are viewing the initial panel, you are not connected to any queue manager. However, as soon as you press Enter, you are connected to the queue manager, or a queue manager in the queue sharing group named in the **Connect name** field. You can leave this field blank; this means that you are using the default queue manager for batch applications. This is defined in CSQBDEFV (see [Task 19: Set up Batch, TSO, and RRS adapters](#) for information about this).

Use the **Target queue manager** field to specify the queue manager where the actions you request are to be performed. If you leave this field blank, it defaults to the queue manager specified in the **Connect name** field. You can specify a target queue manager that is not the one you connect to. In this case, you would normally specify the name of a remote queue manager object that provides a queue manager alias definition (the name is used as the *ObjectQMgrName* when opening the command input queue). To do this, you must have suitable queues and channels set up to access the remote queue manager.

The **Action queue manager** field allows you to specify a queue manager that is in the same queue sharing group as the queue manager specified in the **Target queue manager** field to be the queue manager where the actions you request are to be performed. If you specify * in this field, the actions you request are performed on all queue managers in the queue sharing group. If you leave this field blank, it defaults to the value specified in the **Target queue manager** field. The **Action queue manager** field corresponds to using the CMDSCOPE command modifier described in [The MQSC commands](#).

Queue manager defaults

If you leave any queue manager fields blank, or choose to connect to a queue sharing group, a secondary window opens when you press **Enter**. This window confirms the names of the queue managers you will be using. Press **Enter** to continue. When you return to the initial panel after having made some requests, you find fields completed with the actual names.

Queue manager levels

If the action queue manager is not at IBM MQ 8.0.0 or later, some fields are not displayed, and some values cannot be entered. A few objects and actions are disallowed. In such cases, a secondary window opens asking for you to confirm that you want to proceed.

Using the function keys and command line with the ISPF control panels on z/OS

To use the panels, you must use the function keys or enter the equivalent commands in the ISPF control panel command area.

- [Function keys](#)
 - [Processing your actions](#)
 - [“Displaying IBM MQ user messages” on page 479](#)
 - [Canceling your actions](#)
 - [Getting help](#)
- [Using the command line](#)

Function keys

The function keys have special settings for IBM MQ. (This means that you cannot use the ISPF default values for the function keys; if you have previously used the KEYLIST OFF ISPF command anywhere, you must type KEYLIST ON in the command area of any operations and control panel and then press Enter to enable the IBM MQ settings.)

These function key settings can be displayed on the panels, as shown in [“Using the operations and control panels on z/OS”](#) on page 471. If the settings are not shown, type PFSHOW in the command area of any operations and control panel and then press **Enter**. To remove the display of the settings, use the command PFSHOW OFF.

The function key settings in the operations and control panels conform to CUA standards. Although you can change the key setting through normal ISPF procedures (such as the **KEYLIST** utility), you are not recommended to do so.

Note: Using the **PFSHOW** and **KEYLIST** commands affects any other logical ISPF screens that you have, and their settings remain when you leave the operations and control panels.

Processing your actions

Press **Enter** to carry out the action requested on a panel. The information from the panel is sent to the queue manager for processing.

Each time you press **Enter** in the panels, IBM MQ generates one or more operator messages. If the operation was successful, you get confirmation message CSQ9022I, otherwise you get some error messages.

Displaying IBM MQ user messages

Press function key F10 in any panel to see the IBM MQ user messages.

Canceling your actions

On the initial panel, both F3 and F12 exit the operations and control panels and return you to ISPF. No information is sent to the queue manager.

On any other panel, press function keys F3 or F12 to leave the current panel **ignoring any data you have typed since last pressing Enter**. Again, no information is sent to the queue manager.

- F3 takes you straight back to the initial panel.
- F12 takes you back to the previous panel.

Getting help

Each panel has help panels associated with it. The help panels use the ISPF protocols:

- Press function key F1 on any panel to see general help (extended help) about the task.
- Press function key F1 with the cursor on any field to see specific help about that field.
- Press function key F5 from any field help panel to get the general help.
- Press function key F3 to return to the base panel, that is, the panel from which you pressed function key F1.
- Press function key F6 from any help panel to get help about the function keys.

If the help information carries on into a second or subsequent pages, a **More** indicator is displayed in the upper-right of the panel. Use these function keys to navigate through the help pages:

- F11 to get to the next help page (if there is one).
- F10 to get back to the previous help page (if there is one).

Using the command line

You never need to use the command line to issue the commands used by the operations and control panels because they are available from function keys. The command line is provided to allow you to enter normal ISPF commands (like **PFSHOW**).

The ISPF command PANELID ON displays the name of the current CSQOREXX panel.

The command line is initially displayed in the default position at the bottom of the panels, regardless of what ISPF settings you have. You can use the SETTINGS ISPF command from any of the operations and

control panels to change the position of the command line. The settings are remembered for subsequent sessions with the operations and control panels.

Using the IBM MQ for z/OS utilities

IBM MQ for z/OS provides a set of utility programs that you can use to help with system administration.

IBM MQ for z/OS supplies a set of utility programs to help you perform various administrative tasks, including the following:

- Manage message security policies.
- Perform backup, restoration, and reorganization tasks.
- Issue commands and process object definitions.
- Generate data-conversion exits.
- Modify the bootstrap data set.
- List information about the logs.
- Print the logs.
- Set up Db2 tables and other Db2 utilities.
- Process messages on the dead-letter queue.

The message security policy utility

The message security policy utility (CSQOUTIL) runs as a stand-alone utility to manage message security policies. See [The message security policy utility \(CSQOUTIL\)](#) for more information.

The CSQUTIL utility

This is a utility program provided to help you with backup, restore and reorganize tasks. See [“Using the CSQUTIL utility for IBM MQ for z/OS”](#) on page 481.

The data conversion exit utility

The IBM MQ for z/OS data conversion exit utility (**CSQUCVX**) runs as a stand-alone utility to create data conversion exit routines.

The change log inventory utility

The IBM MQ for z/OS change log inventory utility program (**CSQJU003**) runs as a stand-alone utility to change the bootstrap data set (BSDS). You can use the utility to perform the following functions:

- Add or delete active or archive log data sets.
- Supply passwords for archive logs.

The print log map utility

The IBM MQ for z/OS print log map utility program (**CSQJU004**) runs as a stand-alone utility to list the following information:

- Log data set name and log RBA association for both copies of all active and archive log data sets. If dual logging is not active, there is only one copy of the data sets.
- Active log data sets available for new log data.
- Contents of the queue of checkpoint records in the bootstrap data set (BSDS).
- Contents of the archive log command history record.

- System and utility time stamps.

The log print utility

The log print utility program (**CSQ1LOGP**) is run as a stand-alone utility. You can run the utility specifying:

- A bootstrap data set (BSDS)
- Active logs (with no BSDS)
- Archive logs (with no BSDS)

The queue sharing group utility

The queue sharing group utility program (**CSQ5PQSG**) runs as a stand-alone utility to set up Db2 tables and perform other Db2 tasks required for queue sharing groups.

The active log preformat utility

The active log preformat utility (**CSQJUFMT**) formats active log data sets before they are used by a queue manager. If the active log data sets are preformatted by the utility, log write performance is improved on the queue manager's first pass through the active logs.

The dead-letter queue handler utility

The dead-letter queue handler utility program (**CSQUDLQH**) runs as a stand-alone utility. It checks messages that are on the dead-letter queue and processes them according to a set of rules that you supply to the utility.

The queue load and unload utility

The queue load and unload utility copies or moves the contents of a queue, or its messages, to a file. The utility was originally shipped as the **QLOAD** utility in IBM MQ Supportpac MO03. From IBM MQ 8.0 it is integrated into the product as executable module **CSQUDMSG** in the SCSQLOAD library, with an alias of **QLOAD** for compatibility. Sample JCL is provided as member CSQ4QLOD in SCSQPROC.

The equivalent utility for Multiplatforms is called **dmpmqmsg**. For details of the available options, including the differences for z/OS, see [dmpmqmsg \(queue load and unload\)](#).

You can also reload messages as described in [Restoring messages from a data set to a queue \(LOAD\) on z/OS](#) and [Restoring messages from a data set to a queue \(SLOAD\) on z/OS](#).

Using the CSQUTIL utility for IBM MQ for z/OS

The CSQUTIL utility program is provided with IBM MQ for z/OS to help you perform backup, restoration, and reorganization tasks, and to issue commands and process object definitions.

About this task

Use this utility program to invoke the following functions. For example, you can issue commands from a sequential data set using the **COMMAND** function of the CSQUTIL utility. This function transfers the commands, as messages, to the *system-command input queue* and waits for the response, which is printed together with the original commands in SYSPRINT.

For more information about the CSQUTIL utility program, see [IBM MQ utility program \(CSQUTIL\)](#).

Procedure

- [COMMAND](#)

Use this function to issue MQSC commands, to record object definitions, and to make client-channel definition files.

- [COPY](#)

Use this function to read the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set, and put them into a sequential file and retain the original queue.

- [COPYPAGE](#)

Use this function to copy whole page sets to larger page sets.

- [EMPTY](#)

Use this function to delete the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set, retaining the definitions of the queues.

- [FORMAT](#)

Use this function to format IBM MQ for z/OS page sets.

- [Restoring messages from a data set to a queue \(LOAD\) on z/OS](#)

Use this function to restore the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set from a sequential file created by the COPY function.

- [PAGEINFO](#)

Use this function to extract page set information from one or more page sets.

- [RESETPAGE](#)

Use this function to copy whole page sets to other page set data sets and reset the log information in the copy.

- [SCOPY](#)

Use this function to copy the contents of a queue to a data set while the queue manager is offline.

- [SDEFS](#)

Use this function to produce a set of define commands for objects while the queue manager is offline.

- [SLOAD](#)

Use this function to restore messages from the destination data set of an earlier COPY or SCOPY operation. SLOAD processes a single queue.

- [SWITCH](#)

Use this function to switch or query the transmission queue associated with cluster-sender channels.

Using the Command Facility on z/OS

Use the editor to enter or amend MQSC commands to be passed to the queue manager.

From the primary panel, CSQOPRIA, select option **8 Command**, to start the Command Facility.

You are presented with an edit session of a sequential file, *prefix*.CSQUTIL.COMMANDS, used as input to the CSQUTIL COMMAND function; see [Issuing commands to IBM MQ](#).

You do not need to prefix commands with the command prefix string (CPF).

You can continue MQSC commands on subsequent lines by terminating the current line with the continuation characters + or -. Alternatively, use line edit mode to provide long MQSC commands or the values of long attribute values within the command.

line edit

To use line edit, move the cursor to the appropriate line in the edit panel and use **F4** to display a single line in a scrollable panel. A single line can be up to 32 760 bytes of data.

To leave line edit:

- **F3 exit** saves changes made to the line and exits
- **F12 cancel** returns to the edit panel discarding changes made to the line.

To discard changes made in the edit session, use **F12 cancel** to terminate the edit session leaving the contents of the file unchanged. Commands are not executed.

Executing commands

When you have finished entering MQSC commands, terminate the edit session with **F3 exit** to save the contents of the file and invoke CSQUTIL to pass the commands to the queue manager. The output from command processing is held in file *prefix.CSQUTIL.OUTPUT*. An edit session opens automatically on this file so that you can view the responses. Press **F3 exit** to exit this session and return to the main menu.

Working with IBM MQ objects on z/OS

Many of the tasks described in this documentation involve manipulating IBM MQ objects. The object types are queue managers, queues, process definitions, namelists, channels, client connection channels, listeners, services, and authentication information objects.

- [Defining simple queue objects](#)
- [Defining other types of objects](#)
- [Working with object definitions](#)
- [Working with namelists](#)

Defining simple queue objects

To define a new object, use an existing definition as the basis for it. You can do this in one of three ways:

- By selecting an object that is a member of a list displayed as a result of options selected on the initial panel. You then enter action type 2 (**Define like**) in the action field next to the selected object. Your new object has the attributes of the selected object, except the disposition. You can then change any attributes in your new object as you require.
- On the initial panel, select the **Define like** action type, enter the type of object that you are defining in the **Object type** field, and enter the name of a specific existing object in the **Name** field. Your new object has the same attributes as the object you named in the **Name** field, except the disposition. You can then change any attributes in your new object definition as you require.
- By selecting the **Define like** action type, specifying an object type and then leaving the **Name** field blank. You can then define your new object and it has the default attributes defined for your installation. You can then change any attributes in your new object definition as you require.

Note: You do not enter the name of the object you are defining on the initial panel, but on the **Define** panel you are presented with.

The following example demonstrates how to define a local queue using an existing queue as a template.

Defining a local queue

To define a local queue object from the operations and control panels, use an existing queue definition as the basis for your new definition. There are several panels to complete. When you have completed all the panels and you are satisfied that the attributes are correct, press Enter to send your definition to the queue manager, which then creates the actual queue.

Use the **Define like** action either on the initial panel or against an object entry in a list displayed as a result of options selected on the initial panel.

For example, starting from the initial panel, complete these fields:

Action	2 (Define like)
Object type	QLOCAL

Name QUEUE.YOU.LIKE. This is the name of the queue that provides the attributes for your new queue.

Press Enter to display the **Define a Local Queue** panel. The queue name field is blank so that you can supply the name for the new queue. The description is that of the queue upon which you are basing this new definition. Over type this field with your own description for the new queue.

The values in the other fields are those of the queue upon which you are basing this new queue, except the disposition. You can over type these fields as you require. For example, type Y in the **Put enabled** field (if it is not already Y) if suitably authorized applications can put messages on this queue.

You get field help by moving the cursor into a field and pressing function key F1. Field help provides information about the values that can be used for each attribute.

When you have completed the first panel, press function key F8 to display the second panel.

Hints:

1. Do not press Enter at this stage, otherwise the queue will be created before you have a chance to complete the remaining fields. (If you do press Enter prematurely, do not worry; you can always alter your definition later on.)
2. Do not press function keys F3 or F12, or the data you typed will be lost.

Press function key F8 repeatedly to see and complete the remaining panels, including the trigger definition, event control, and backout reporting panels.

When your local queue definition is complete

When your definition is complete, press Enter to send the information to the queue manager for processing. The queue manager creates the queue according to the definition you have supplied. If you do not want the queue to be created, press function key F3 to exit and cancel the definition.

Defining other types of objects

To define other types of object, use an existing definition as the base for your new definition as explained in [Defining a local queue](#).

Use the **Define like** action either on the initial panel or against an object entry in a list displayed as a result of options selected on the initial panel.

For example, starting from the initial panel, complete these fields:

Action	2 (Define like)
Object type	QALIAS, NAMELIST, PROCESS, CHANNEL, and other resource objects.
Name	Leave blank or enter the name of an existing object of the same type.

Press Enter to display the corresponding DEFINE panels. Complete the fields as required and then press Enter again to send the information to the queue manager.

Like defining a local queue, defining another type of object generally requires several panels to be completed. Defining a namelist requires some additional work, as described in [“Working with namelists”](#) on page 485.

Working with object definitions

When an object has been defined, you can specify an action in the **Action** field, to alter, display, or manage it.

In each case, you can either:

- Select the object you want to work with from a list displayed as a result of options selected on the initial panel. For example, having entered 1 in the **Action** field to display objects, Queue in the **Object type**

field, and * in the **Name** field, you are presented with a list of all queues defined in the system. You can then select from this list the queue with which you need to work.

- Start from the initial panel, where you specify the object you are working with by completing the **Object type** and **Name** fields.

Altering an object definition

To alter an object definition, specify action 3 and press Enter to see the ALTER panels. These panels are very similar to the DEFINE panels. You can alter the values you want. When your changes are complete, press Enter to send the information to the queue manager.

Displaying an object definition

If you want to see the details of an object without being able to change them, specify action 1 and press Enter to see the DISPLAY panels. Again, these panels are similar to the DEFINE panels except that you cannot change any of the fields. Change the object name to display details of another object.

Deleting an object

To delete an object, specify action 4 (Manage) and the **Delete** action is one of the actions presented on the resulting menu. Select the **Delete** action.

You are asked to confirm your request. If you press function key F3 or F12, the request is canceled. If you press Enter, the request is confirmed and passed to the queue manager. The object you specified is then deleted.

Note: You cannot delete most types of channel object unless the channel initiator is started.

Working with namelists

When working with namelists, proceed as you would for other objects.

For the actions DEFINE LIKE or ALTER, press function key F11 to add names to the list or to change the names in the list. This involves working with the ISPF editor and all the normal ISPF edit commands are available. Enter each name in the namelist on a separate line.

When you use the ISPF editor in this way, the function key settings are the normal ISPF settings, and **not** those used by the other operations and control panels.

If you need to specify lowercase names in the list, specify CAPS(OFF) on the editor panel command line. When you do this, all the namelists that you edit in the future are in lowercase until you specify CAPS(ON).

When you have finished editing the namelist, press function key F3 to end the ISPF edit session. Then press Enter to send the changes to the queue manager.

Attention: If you do not press Enter at this stage but press function key F3 instead, you lose any updates that you have typed in.

Implementing the system using multiple cluster transmission queues

It makes no difference if the channel is used in a single cluster, or an overlapping cluster. When the channel is selected and started, the channel selects the transmission queue depending on the definitions.

Procedure

- If you are using the DEFCLXQ option, see [“Using the automatic definition of queues and switching” on page 486.](#)
- If you are using a staged approach, see [“Changing your cluster-sender channels using a phased approach” on page 486.](#)

Using the automatic definition of queues and switching

Use this option if you are planning on using the DEFCLXQ option. There will be a queue created for every channel, and every new channel.

Procedure

1. Review the definition of the SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE and change the attributes if required.

This queue is defined in member SCSQPROC (csq4insx).

2. Create the SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE model queue.
3. Apply security policies for this model queue, and the SYSTEM.CLUSTER.TRANSMIT.** queues.

For z/OS the channel initiator started task user ID needs:

- Control access to CLASS(MQADMIN) for

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelName
```

- Update access to CLASS(MQQUEUE) for

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelName
```

Changing your cluster-sender channels using a phased approach

Use this option if you are planning on using a staged approach. This process allows you to move to the new cluster-sender channels at various times to suit the needs of your enterprise.

Before you begin

- Identify your business applications, and which channels are used.
- For the queues you use, display the clusters they are in.
- Display the channels to show the connection names, the names of the remote queue managers, and which clusters the channel supports.

About this task

- Create a transmission queue. On z/OS you might want to consider which page set you use for the queue.
- Set up security policy for the queue.
- Change any queue monitoring to include this queue name.
- Decide which channels are to use this transmission queue. The channels should have a similar name, so generic characters ' * ' in the CLCHNAME identify the channel.
- When you are ready to use the new function, alter the transmission queue to specify the name of the channels to use this transmission queue. For example CLUSTER1.TOPARIS, or CLUSTER1.* or *.TOPARIS
- Start the channels

Procedure

1. Use the DIS CLUSQMGR(xxxx) XMITQ command to display the cluster sender channels defined in the cluster, where xxxx is the name of the remote queue manager.
2. Set up the security profile for the transmission queue and give the queue access to the channel initiator.
3. Define the transmission queue to be used, and specify USAGE(XMITQ) INDXTYPE(CORRELID) SHARE and CLCHNAME(value)

The channel initiator started task user ID needs the following access:

```
alter class(MQADMIN) ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channel  
update class(MQQUEUE ssid.SYSTEM.CLUSTER.TRANSMIT.channel
```

and the user ID using the SWITCH command needs the following access:

```
alter cl(MQADMIN) ssid.QUEUE.queueName
```

4. Stop and restart the channels.

The channel change occurs when the channel starts using an MQSC command, or you use CSQUTIL. You can identify which channels need to be restarted using the SWITCH CHANNEL(*) STATUS of CSQUTIL

If you have problems when the channel is started, stop the channel, resolve the problems, and restart the channel.

Note that you can change the CLCHNAME attribute as often as you need to.

The value of CLCHNAME used is the one when the channel is started, so you can change the CLCHNAME definition while the channel continues to use the definitions from the time that it started. The channel uses the new definition when it is restarted.

Undoing a change to a transmission queue on z/OS

You need to have a process to backout a change if the results are not as you expect.

What can go wrong?

If the new transmission queue is not what you expect:

1. Check the CLCHNAME is as you expect
2. Review the job log to check if the switch process has finished. If not, wait and check the new transmission queue of the channel later.

If you are using multiple cluster transmission queues, it is important that you design the transmission queues definitions explicitly and avoid complicated overlapping configuration. In this way, you can make sure that if there are problems, you can go back to the original queues and configuration.

If you encounter problems during the move to using a different transmission queue, you must resolve any problems before you can proceed with the change.

An existing change request must complete before a new change request can be made. For example, you:

1. Define a new transmission queue with a maximum depth of one and there are 10 messages waiting to be sent.
2. Change the transmission queue to specify the channel name in the CLCHNAME parameter.
3. Stop and restart the channel. The attempt to move the messages fails and reports the problems.
4. Change the CLCHNAME parameter on the transmission queue to be blank.
5. Stop and restart the channel. The channel continues to try and complete the original request, so the channel continues to use the new transmission queue.
6. Need to resolve the problems and restart the channel so the moving of messages completes successfully.

Next time the channel is restarted it picks up any changes, so if you had set CLCHNAME to blanks, the channel will not use the specified transmission queue.

In this example, changing the CLCHNAME on the transmission queue to blanks does not necessarily mean that the channel uses the SYSTEM.CLUSTER.TRANSMIT queue, as there might be other transmission queues whose CLCHNAME parameter match the channel name. For example, a generic name, or the

queue manager attribute DEFCLXQ might be set to channel, so the channel uses a dynamic queue instead of the SYSTEM.CLUSTER.TRANSMIT queue.

Writing programs to administer IBM MQ for z/OS

You can write your own application programs to administer a queue manager. Use this topic to understand the requirements for writing your own administration programs.

Start of General-use programming interface information

This set of topics contains hints and guidance to enable you to issue IBM MQ commands from an IBM MQ application program.

Note: In this topic, the MQI calls are described using C-language notation. For typical invocations of the calls in the COBOL, PL/I, and assembler languages, see [Function calls](#).

Understanding how it all works

In outline, the procedure for issuing commands from an application program is as follows:

1. Build an IBM MQ command into a type of IBM MQ message called a *request message*. The command can be in MQSC or PCF format.
2. Send (use MQPUT) this message to a special queue called the system-command input queue. The IBM MQ command processor runs the command.
3. Retrieve (use MQGET) the results of the command as *reply messages* on the reply-to queue. These messages contain the user messages that you need to determine whether your command was successful and, if it was, what the results were.

Then it is up to your application program to process the results.

This set of topics contains:

Preparing queues for administration programs

Administration programs require a number of predefined queues for system command input and receiving responses.

This section applies to commands in the MQSC format. For the equivalent in PCF, see [“Usando os formatos de comando programável do IBM MQ” on page 27](#).

Before you can issue any MQPUT or MQGET calls, you must first define, and then open, the queues you are going to use.

Defining the system-command input queue

The system-command input queue is a local queue called SYSTEM.COMMAND.INPUT. The supplied CSQINP2 initialization data set, thlqual.SCSQPROC(CSQ4INSG), contains a default definition for the system-command input queue. For compatibility with IBM MQ on other platforms, an alias of this queue, called SYSTEM.ADMIN.COMMAND.QUEUE is also supplied. See [Sample definitions supplied with IBM MQ](#) for more information.

Defining a reply-to queue

You must define a reply-to queue to receive reply messages from the IBM MQ command processor. It can be any queue with attributes that allow reply messages to be put on it. However, for normal operation, specify these attributes:

- USAGE(NORMAL)
- NOTRIGGER (unless your application uses triggering)

Avoid using persistent messages for commands, but if you choose to do so, the reply-to queue must not be a temporary dynamic queue.

The supplied CSQINP2 initialization data set, thlqual.SCSQPROC(CSQ4INSG), contains a definition for a model queue called SYSTEM.COMMAND.REPLY.MODEL. You can use this model to create a dynamic reply-to queue.

Note: Replies generated by the command processor can be up to 15 000 bytes in length.

If you use a permanent dynamic queue as a reply-to queue, your application should allow time for all PUT and GET operations to complete before attempting to delete the queue, otherwise MQRC2055 (MQRC_Q_NOT_EMPTY) can be returned. If this occurs, try the queue deletion again after a few seconds.

Opening the system-command input queue

Before you can open the system-command input queue, your application program must be connected to your queue manager. Use the MQI call MQCONN or MQCONNX to do this.

Then use the MQI call MQOPEN to open the system-command input queue. To use this call:

1. Set the **Options** parameter to MQOO_OUTPUT
2. Set the MQOD object descriptor fields as follows:

ObjectType

MQOT_Q (the object is a queue)

ObjectName

SYSTEM.COMMAND.INPUT

ObjectQMgrName

If you want to send your request messages to your local queue manager, leave this field blank. This means that your commands are processed locally.

If you want your IBM MQ commands to be processed on a remote queue manager, put its name here. You must also have the correct queues and links set up, as described in [Distributed queuing and clusters](#).

Opening a reply-to queue

To retrieve the replies from an IBM MQ command, you must open a reply-to queue. One way of doing this is to specify the model queue, SYSTEM.COMMAND.REPLY.MODEL in an MQOPEN call, to create a permanent dynamic queue as the reply-to queue. To use this call:

1. Set the **Options** parameter to MQOO_INPUT_SHARED
2. Set the MQOD object descriptor fields as follows:

ObjectType

MQOT_Q (the object is a queue)

ObjectName

The name of the reply-to queue. If the queue name you specify is the name of a model queue object, the queue manager creates a dynamic queue.

ObjectQMgrName

To receive replies on your local queue manager, leave this field blank.

DynamicQName

Specify the name of the dynamic queue to be created.

Using the command server

The command server is an IBM MQ component that works with the command processor component. You can send formatted messages to the command server which interprets the messages, runs the administration requests, and sends responses back to your administration application.

The command server reads request messages from the system-command input queue, verifies them, and passes the valid ones as commands to the command processor. The command processor processes the commands and puts any replies as reply messages on to the reply-to queue that you specify. The first reply message contains the user message CSQN205I. See [“Interpreting the reply messages from the command server”](#) on page 493 for more information. The command server also processes channel initiator and queue sharing group commands, wherever they are issued from.

Identifying the queue manager that processes your commands

The queue manager that processes the commands you issue from an administration program is the queue manager that owns the system-command input queue that the message is put onto.

Starting the command server

Normally, the command server is started automatically when the queue manager is started. It becomes available as soon as the message CSQ9022I 'START QMGR' NORMAL COMPLETION is returned from the START QMGR command. The command server is stopped when all the connected tasks have been disconnected during the system termination phase.

You can control the command server yourself using the START CMDSERV and STOP CMDSERV commands. To prevent the command server starting automatically when IBM MQ is restarted, you can add a STOP CMDSERV command to your CSQINP1 or CSQINP2 initialization data sets. However, this is not recommended as it prevents any channel initiator or queue sharing group commands being processed.

The STOP CMDSERV command stops the command server as soon as it has finished processing the current message, or immediately if no messages are being processed.

If the command server has been stopped by a STOP CMDSERV command in the program, no other commands from the program can be processed. To restart the command server, you must issue a START CMDSERV command from the z/OS console.

If you stop and restart the command server while the queue manager is running, all the messages that are on the system-command input queue when the command server stops are processed when the command server is restarted. However, if you stop and restart the queue manager after the command server is stopped, only the persistent messages on the system-command input queue are processed when the command server is restarted. All nonpersistent messages on the system-command input queue are lost.

Sending commands to the command server

For each command, you build a message containing the command, then put it onto the system-command input queue.

Building a message that includes IBM MQ commands

You can incorporate IBM MQ commands in an application program by building request messages that include the required commands. For each such command you:

1. Create a buffer containing a character string representing the command.
2. Issue an MQPUT call specifying the buffer name in the **buffer** parameter of the call.

The simplest way to do this in C is to define a buffer using 'char'. For example:

```
char message_buffer[ ] = "ALTER QLOCAL(SALES) PUT(ENABLED)";
```

When you build a command, use a null-terminated character string. Do not specify a command prefix string (CPF) at the start of a command defined in this way. This means that you do not have to alter your command scripts if you want to run them on another queue manager. However, you must take into account that a CPF is included in any response messages that are put onto the reply-to queue.

The command server folds all lowercase characters to uppercase unless they are inside quotation marks.

Commands can be any length up to a maximum 32 762 characters.

Putting messages on the system-command input queue

Use the MQPUT call to put request messages containing commands on the system-command input queue. In this call you specify the name of the reply-to queue that you have already opened.

To use the MQPUT call:

1. Set these MQPUT parameters:

Hconn

The connection handle returned by the MQCONN or MQCONNX call.

Hobj

The object handle returned by the MQOPEN call for the system-command input queue.

BufferLength

The length of the formatted command.

Buffer

The name of the buffer containing the command.

2. Set these MQMD fields:

MsgType

MQMT_REQUEST

Format

MQFMT_STRING or MQFMT_NONE

If you are not using the same code page as the queue manager, set *CodedCharSetId* as appropriate and set MQFMT_STRING, so that the command server can convert the message. Do not set MQFMT_ADMIN, as that causes your command to be interpreted as PCF.

ReplyToQ

Name of your reply-to queue.

ReplyToQMgr

If you want replies sent to your local queue manager, leave this field blank. If you want your IBM MQ commands to be sent to a remote queue manager, put its name here. You must also have the correct queues and links set up, as described in [Distributed queuing and clusters](#).

3. Set any other MQMD fields, as required. You should normally use nonpersistent messages for commands.

4. Set any *PutMsgOpts* options, as required.

If you specify MQPMO_SYNCPOINT (the default), you must follow the MQPUT call with a syncpoint call.

Using MQPUT1 and the system-command input queue

If you want to put just one message on the system-command input queue, you can use the **MQPUT1** call. This call combines the functions of an **MQOPEN**, followed by an **MQPUT** of one message, followed by an **MQCLOSE**, all in one call. If you use this call, modify the parameters accordingly. See [Putting one message on a queue using the MQPUT1 call](#) for details.

Retrieving replies to your commands

The command server sends a response to a reply queue for each request message it receives. Any administration application must receive, and handle the reply messages.

When the command processor processes your commands, any reply messages are put onto the reply-to queue specified in the MQPUT call. The command server sends the reply messages with the same persistence as the command message it received.

Waiting for a reply

Use the MQGET call to retrieve a reply from your request message. One request message can produce several reply messages. For details, see [“Interpreting the reply messages from the command server” on page 493](#).

You can specify a time interval that an MQGET call waits for a reply message to be generated. If you do not get a reply, use the checklist beginning in topic [“If you do not receive a reply” on page 494](#).

To use the MQGET call:

1. Set these parameters:

Hconn

The connection handle returned by the MQCONN or MQCONNX call.

Hobj

The object handle returned by the MQOPEN call for the reply-to queue.

Buffer

The name of the area to receive the reply.

BufferLength

The length of the buffer to receive the reply. This must be a minimum of 80 bytes.

2. To ensure that you only get the responses from the command that you issued, you must specify the appropriate *MsgId* and *CorrelId* fields. These depend on the report options, MQMD_REPORT, you specified in the MQPUT call:

MQRO_NONE

Binary zero, '00...00' (24 nulls).

MQRO_NEW_MSG_ID

Binary zero, '00...00' (24 nulls).

This is the default if none of these options has been specified.

MQRO_PASS_MSG_ID

The *MsgId* from the MQPUT.

MQRO_NONE

The *MsgId* from the MQPUT call.

MQRO_COPY_MSG_ID_TO_CORREL_ID

The *MsgId* from the MQPUT call.

This is the default if none of these options has been specified.

MQRO_PASS_CORREL_ID

The *CorrelId* from the MQPUT call.

For more details on report options, see [Report options and message flags](#).

3. Set the following *GetMsgOpts* fields:

Options

MQGMO_WAIT

If you are not using the same code page as the queue manager, set MQGMO_CONVERT, and set *CodedCharSetId* as appropriate in the MQMD.

WaitInterval

For replies from the local queue manager, try 5 seconds. Coded in milliseconds, this becomes 5 000. For replies from a remote queue manager, and channel control and status commands, try 30 seconds. Coded in milliseconds, this becomes 30 000.

Discarded messages

If the command server finds that a request message is not valid, it discards this message and writes the message CSQN205I to the named reply-to queue. If there is no reply-to queue, the CSQN205I message is put onto the dead-letter queue. The return code in this message shows why the original request message was not valid:

- 00D5020F** It is not of type MQMT_REQUEST.
- 00D50210** It has zero length.
- 00D50212** It is longer than 32 762 bytes.
- 00D50211** It contains all blanks.
- 00D5483E** It needed converting, but *Format* was not MQFMT_STRING.
- Other** See [Command server codes](#)

The command server reply message descriptor

For any reply message, the following MQMD message descriptor fields are set:

<i>MsgType</i>	MQMT_REPLY
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>Priority</i>	As for the MQMD in the message you issued.
<i>Persistence</i>	As for the MQMD in the message you issued.
<i>CorrelId</i>	Depends on the MQPUT report options.
<i>ReplyToQ</i>	None.

The command server sets the *Options* field of the MQPMO structure to MQPMO_NO_SYNCPOINT. This means that you can retrieve the replies as they are created, rather than as a group at the next syncpoint.

Interpreting the reply messages from the command server

Each request message correctly processed by IBM MQ produces at least two reply messages. Each reply message contains a single IBM MQ user message.

The length of a reply depends on the command that was issued. The longest reply you can get is from a **DISPLAY NAMELIST** command, and that can be up to 15 000 bytes in length.

The first user message, CSQN205I, always contains:

- A count of the replies (in decimal), which you can use as a counter in a loop to get the rest of the replies. The count includes this first message.
- The return code from the command preprocessor.
- A reason code, which is the reason code from the command processor.

This message does not contain a CPF.

For example:

```
CSQN205I    COUNT=      4, RETURN=0000000C, REASON=00000008
```

The COUNT field is 8 bytes long and is right-justified. It always starts at position 18, that is, immediately after COUNT=. The RETURN field is 8 bytes long in character hexadecimal and is immediately after RETURN= at position 35. The REASON field is 8 bytes long in character hexadecimal and is immediately after REASON= at position 52.

If the RETURN= value is 00000000 and the REASON= value is 00000004, the set of reply messages is incomplete. After retrieving the replies indicated by the CSQN205I message, issue a further MQGET call to wait for a further set of replies. The first message in the next set of replies is again CSQN205I, indicating how many replies there are, and whether there are still more to come.

See the [mensagens, conclusão e códigos de razão do IBM MQ for z/OS](#) documentation for more details about the individual messages.

If you are using a non-English language feature, the text and layout of the replies are different from those shown here. However, the size and position of the count and return codes in message CSQN205I are the same.

If you do not receive a reply

There are a series of steps you can take if you do not receive a response to request to the command server.

If you do not receive a reply to your request message, work through this checklist:

- Is the command server running?
- Is the *WaitInterval* long enough?
- Are the system-command input and reply-to queues correctly defined?
- Were the MQOPEN calls to these queues successful?
- Are both the system-command input and reply-to queues enabled for MQPUT and MQGET calls?
- Have you considered increasing the MAXDEPTH and MAXMSGL attributes of your queues?
- Are you are using the *CorrelId* and *MsgId* fields correctly?
- Is the queue manager still running?
- Was the command built correctly?
- Are all your remote links defined and operating correctly?
- Were the MQPUT calls correctly defined?
- Has the reply-to queue been defined as a temporary dynamic queue instead of a permanent dynamic queue? (If the request message is persistent, you must use a permanent dynamic queue for the reply.)

When the command server generates replies but cannot write them to the reply-to queue that you specify, it writes them to the dead-letter queue.

Passing commands using MGCRE

With appropriate authorization, an application program can make requests to multiple queue managers using a z/OS service routine.

If you have the correct authorization, you can pass IBM MQ commands from your program to multiple queue managers by the MGCRE (SVC 34) z/OS service. See the [z/OS MVS Programming: Authorized Assembler Services Guide](#) for more information.

The value of the CPF identifies the particular queue manager to which the command is directed. For information about CPFs, see [User IDs for command security and command resource security](#) and [“Issuing queue manager commands on z/OS” on page 457](#).

If you use MGCRE, you can use a Command and Response Token (CART) to get the direct responses to the command.

Examples of commands and their replies

Use this topic as a series of examples of commands to the command server and the responses from the command server.

Here are some examples of commands that could be built into IBM MQ messages, and the user messages that are the replies. Unless otherwise stated, each line of the reply is a separate message.

- [Messages from a DEFINE command](#)
- [Messages from a DELETE command](#)
- [Messages from DISPLAY commands](#)
- [Messages from commands with CMDSCOPE](#)
- [Messages from commands that generate commands with CMDSCOPE](#)

Messages from a DEFINE command

The following command:

```
DEFINE QLOCAL(Q1)
```

produces these messages:

```
CSQN205I    COUNT=    2, RETURN=00000000, REASON=00000000  
CSQ9022I +CSQ1 CSQMMSGP ' DEFINE QLOCAL' NORMAL COMPLETION
```

These reply messages are produced on normal completion.

Messages from a DELETE command

The following command:

```
DELETE QLOCAL(Q2)
```

produces these messages:

```
CSQN205I    COUNT=    4, RETURN=00000000C, REASON=000000008  
CSQM125I +CSQ1 CSQMUQLC QLOCAL (Q2) QSGDISP(QMGR) WAS NOT FOUND  
CSQM090E +CSQ1 CSQMUQLC FAILURE REASON CODE X'00D44002'  
CSQ9023E +CSQ1 CSQMUQLC ' DELETE QLOCAL' ABNORMAL COMPLETION
```

These messages indicate that a local queue called Q2 does not exist.

Messages from DISPLAY commands

The following examples show the replies from some DISPLAY commands.

Finding out the name of the dead-letter queue

If you want to find out the name of the dead-letter queue for a queue manager, issue this command from an application program:

```
DISPLAY QMGR DEADQ
```

The following three user messages are returned, from which you can extract the required name:

```

CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM409I +CSQ1 QMNAME(CSQ1) DEADQ(SYSTEM.DEAD.QUEUE      )
CSQ9022I +CSQ1 CSQMDRTS ' DISPLAY QMGR' NORMAL COMPLETION

```

Messages from the DISPLAY QUEUE command

The following examples show how the results from a command depend on the attributes specified in that command.

Example 1

You define a local queue using the command:

```

DEFINE QLOCAL(Q1) DESCR('A sample queue') GET(ENABLED) SHARE

```

If you issue the following command from an application program:

```

DISPLAY QUEUE(Q1) SHARE GET DESCR

```

these three user messages are returned:

```

CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(Q1                                ) TYPE(
QLOCAL ) QSGDISP(QMGR  )
DESCR(A sample queue
) SHARE GET(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION

```

Note: The second message, CSQM401I, is shown here occupying four lines.

Example 2

Two queues have names beginning with the letter A:

- A1 is a local queue with its PUT attribute set to DISABLED.
- A2 is a remote queue with its PUT attribute set to ENABLED.

If you issue the following command from an application program:

```

DISPLAY QUEUE(A*) PUT

```

these four user messages are returned:

```

CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(A1                                ) TYPE(
QLOCAL ) QSGDISP(QMGR  )
PUT(DISABLED )
CSQM406I +CSQ1 QUEUE(A2                                ) TYPE(
QREMOTE ) PUT(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION

```


Note: The second and third messages, CSQM401I and CSQM406I, are shown here occupying three and two lines.

Messages from the DISPLAY NAMELIST command

You define a namelist using the command:

```
DEFINE NAMELIST(N1) NAMES(Q1,SAMPLE_QUEUE)
```

If you issue the following command from an application program:

```
DISPLAY NAMELIST(N1) NAMES NAMCOUNT
```

the following three user messages are returned:

```
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000000
CSQM407I +CSQ1 NAMELIST(N1
GDISP(QMGR ) NAMCOUNT(    2) NAMES(Q1
,SAMPLE_QUEUE
CSQ9022I +CSQ1 CSQMMSGP ' DISPLAY NAMELIST' NORMAL COMPLETION
```

Note: The second message, CSQM407I, is shown here occupying three lines.

Messages from commands with CMDSCOPE

The following examples show the replies from commands that have been entered with the CMDSCOPE attribute.

Messages from the ALTER PROCESS command

The following command:

```
ALT PRO(V4) CMDSCOPE(*)
```

produces the following messages:

```
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'ALT PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I  COUNT=    5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ26
CSQM125I !MQ26 CSQMMSGP PROCESS(V4) QSGDISP(QMGR) WAS NOT FOUND
CSQM090E !MQ26 CSQMMSGP FAILURE REASON CODE X'00D44002'
CSQ9023E !MQ26 CSQMMSGP ' ALT PRO' ABNORMAL COMPLETION
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' ALT PRO' NORMAL COMPLETION
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000008
CSQN123E !MQ25 'ALT PRO' command for CMDSCOPE(*) abnormal completion
```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). The command was successful on MQ25 but the process definition did not exist on MQ26, so the command failed on that queue manager.

Messages from the DISPLAY PROCESS command

The following command:

```
DIS PRO(V*) CMDSCOPE(*)
```

produces the following messages:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ26
CSQM408I !MQ26 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ26 PROCESS(V3) QSGDISP(QMGR)
CSQ9022I !MQ26 CSQMDRTS 'DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 7, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ25
CSQM408I !MQ25 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ25 PROCESS(V2) QSGDISP(GROUP)
CSQM408I !MQ25 PROCESS(V3) QSGDISP(QMGR)
CSQM408I !MQ25 PROCESS(V4) QSGDISP(QMGR)
CSQ9022I !MQ25 CSQMDRTS 'DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS PRO' command for CMDSCOPE(*) normal completion
```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Information is displayed about all the processes on each queue manager with names starting with the letter V.

Messages from the DISPLAY CHSTATUS command

The following command:

```
DIS CHS(VT) CMDSCOPE(*)
```

produces the following messages:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS CHS' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ25
CSQM422I !MQ25 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ25 CSQXDRTS 'DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ26
CSQM422I !MQ26 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ26 CSQXDRTS 'DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS CHS' command for CMDSCOPE(*) normal completion
```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Information is displayed about channel status on each queue manager.

Messages from the STOP CHANNEL command

The following command:

```
STOP CHL(VT) CMDSCOPE(*)
```

produces these messages:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'STOP CHL' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQM134I !MQ25 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
SQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQM134I !MQ26 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQ9022I !MQ26 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQ9022I !MQ25 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'STOP CHL' command for CMDSCOPE(*) normal completion

```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Channel VT was stopped on each queue manager.

Messages from commands that generate commands with CMDSCOPE

The following command:

```
DEF PRO(V2) QSGDISP(GROUP)
```

produces these messages:

```

CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQM122I !MQ25 CSQMMSGP ' DEF PRO' COMPLETED FOR QSGDISP(GROUP)
CSQN138I !MQ25 'DEFINE PRO' command generated for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ26
CSQ9022I !MQ26 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DEFINE PRO' command for CMDSCOPE(*) normal completion

```

These messages tell you that the command was entered on queue manager MQ25. When the object was created on the shared repository, another command was generated and sent to all the active queue managers in the queue sharing group (MQ25 and MQ26).

z/OS

Managing IBM MQ resources on z/OS

Use the links in this topic to find out how to manage the resources used by IBM MQ for z/OS, for example, managing log files, data sets, page sets, buffer pools, and coupling facility structures.

Use the following links for details of the different administrative tasks you might have to complete while using IBM MQ for z/OS:

- [“Managing the logs” on page 500](#)
- [“Managing the bootstrap data set \(BSDS\)” on page 509](#)
- [“Managing page sets” on page 516](#)
- [“How to back up and recover page sets” on page 523](#)
- [“How to back up and restore queues using CSQUTIL” on page 526](#)
- [“Managing buffer pools” on page 526](#)

- [“Managing queue sharing groups and shared queues on z/OS” on page 528](#)

Related concepts

[IBM MQ for z/OS concepts](#)

[“Administering IBM MQ for z/OS” on page 457](#)

IBM MQ for z/OS can be controlled and managed by MQSC and PCF commands, by a set of utilities and programs provided with the product, and by authorized applications.

[“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS” on page 458](#)

You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications. Not all commands can be issued from all these sources.

[“Recovery and restart on z/OS” on page 537](#)

Use this topic to understand the recovery and restart mechanisms used by IBM MQ.

Related tasks

[Planning your IBM MQ environment on z/OS](#)

[Configuring queue managers on z/OS](#)

[IBM MQ utilities on z/OS reference](#)

Related reference

[“Using the IBM MQ for z/OS utilities” on page 480](#)

IBM MQ for z/OS provides a set of utility programs that you can use to help with system administration.

[Programmable command formats reference](#)

Managing the logs

Use this topic to understand how to manage your IBM MQ log files, including the log archiving process, using log record compression, log record recovery, and printing log records.

This topic describes the tasks involved in managing the IBM MQ logs. It contains these sections:

Archiving logs with the ARCHIVE LOG command

An authorized operator can archive the current IBM MQ active log data sets whenever required using the **ARCHIVE LOG** command.

When you issue the ARCHIVE LOG command, IBM MQ truncates the current active log data sets, then runs an asynchronous offload process, and updates the BSDS with a record of the offload process.

The **ARCHIVE LOG** command has a **MODE(QUIESCE)** option. With this option, IBM MQ jobs and users are quiesced after a commit point, and the resulting point of consistency is captured in the current active log before it is offloaded.

Consider using the **MODE(QUIESCE)** option when planning a backup strategy for off site recovery. It creates a system-wide point of consistency, which minimizes the number of data inconsistencies when the archive log is used with the most current backup page set copy during recovery. For example:

```
ARCHIVE LOG MODE(QUIESCE)
```

If you issue the **ARCHIVE LOG** command without specifying a **TIME** parameter, the quiesce time period defaults to the value of the **QUIESCE** parameter of the CSQ6ARVP macro. If the time required for the ARCHIVE LOG MODE(QUIESCE) to complete is less than the time specified, the command completes successfully; otherwise, the command fails when the time period expires. You can specify the time period explicitly by using the **TIME** option, for example:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

This command specifies a quiesce period of up to 60 seconds before **ARCHIVE LOG** processing occurs.

Attention: Using the **TIME** option when time is critical can significantly disrupt IBM MQ availability for all jobs and users that use IBM MQ resources.

By default, the command is processed asynchronously from the time you submit the command. (To process the command synchronously with other IBM MQ commands use the **WAIT (YES)** option with **QUIESCE**, but be aware that the z/OS console is locked from IBM MQ command input for the entire **QUIESCE** period.)

During the quiesce period:

- Jobs and users on the queue manager are allowed to go through commit processing, but are suspended if they try to update any IBM MQ resource after the commit.
- Jobs and users that only read data can be affected, since they might be waiting for locks held by jobs or users that were suspended.
- New tasks can start, but they cannot update data.

The output from the **DISPLAY LOG** command uses the message CSQV400I to indicate that a quiesce is in effect.

For example:

```
CSQJ322I +CSQ1 DISPLAY LOG report ...
Parameter  Initial value      SET value
-----
INBUFF     60
OUTBUFF    400
MAXRTU     2
MAXARCH    2
TWOACTV    YES
TWOARCH    YES
TWOBSDS    YES
OFFLOAD    YES
MAXCNOFF   0
WRTHRS    20
DEALLCT    0
COMPLOG    NONE
ZHYWRITE   NO
End of LOG report
CSQJ370I +CSQ1 LOG status report ...
Copy %Full zHyperWrite Encrypted DSName
  1   68 NO          NO          VICY.CSQ1.LOGCOPY1.DS01
  2   68 NO          NO          VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2019-08-15 09:49:30 using RBA=000000000891B000
Latest RBA=000000000891CCF8
Offload task is AVAILABLE
Full logs to offload - 0 of 4
CSQV400I +CSQ1 ARCHIVE LOG QUIESCE CURRENTLY ACTIVE
CSQ9022I +CSQ1 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION
```

When all updates are quiesced, the quiesce history record in the BSDS is updated with the date and time that the active log data sets were truncated, and with the last-written RBA in the current active log data sets. IBM MQ truncates the current active log data sets, switches to the next available active log data sets, and issues message CSQJ311I stating that the offload process started.

If updates cannot be quiesced before the quiesce period expires, IBM MQ issues message CSQJ317I, and **ARCHIVE LOG** processing terminates. The current active log data sets are not truncated, nor switched to the next available log data sets, and the offload process is not started.

Whether the quiesce was successful or not, all suspended users and jobs are then resumed, and IBM MQ issues message CSQJ312I, stating that the quiesce is ended and update activity is resumed.

If **ARCHIVE LOG** is issued when the current active log is the last available active log data set, the command is not processed, and IBM MQ issues the following message:

```
CSQJ319I - csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST
AVAILABLE ACTIVE LOG DATA SET. ARCHIVE LOG PROCESSING
WILL BE TERMINATED
```

If **ARCHIVE LOG** is issued when another **ARCHIVE LOG** command is already in progress, the new command is not processed, and IBM MQ issues the following message:

CSQJ318I - ARCHIVE LOG COMMAND ALREADY IN PROGRESS

For information about the messages issued during archiving, see [Messages for IBM MQ for z/OS](#).

Restarting the log archive process after a failure

If there is a problem during the log archive process (for example, a problem with allocation or tape mounts), the archiving of the active log might be suspended. You can cancel the archive process and restart it by using the following command:

```
ARCHIVE LOG CANCEL OFFLOAD
```

This command cancels any offload processing currently in progress, and restarts the archive process. It starts with the oldest log data set that has not been archived, and proceeds through all active log data sets that need offloading. Any log archive operations that have been suspended are restarted.

Use this command only if you are sure that the current log archive task is no longer functioning, or if you want to restart a previous attempt that failed. This is because the command might cause an abnormal termination of the offload task, which might result in a dump.

Controlling archiving and logging

You can control compression, printing, archiving, recovery and logging with using the CSQ6LOGP, CSQ6ARVP, and CSQ6SYSP macros. Note, that changes to private objects only are logged in IBM MQlogs. Changes to GROUP objects (like shared inbound channels) are also logged, because the definitions are propagated around the group and held locally.

Many aspects of archiving and logging are controlled by parameters set using the CSQ6LOGP, CSQ6ARVP and CSQ6SYSP macros of the system parameter module when the queue manager is customized. See [Tailor your system parameter module](#) for details of these macros.

Some of these parameters can be changed while a queue manager is running using the IBM MQ MQSC SET LOG, SET SYSTEM and SET ARCHIVE commands. They are shown in [Table 28 on page 502](#):

SET command	Parameters
LOG	WRTHRSH, MAXARCH, DEALLCT, MAXRTU, COMPLOG
ARCHIVE	All
SYSTEM	LOGLOAD

You can display the settings of all the parameters using the MQSC [DISPLAY LOG](#), [DISPLAY ARCHIVE](#) and [DISPLAY SYSTEM](#) commands. These commands also show status information about archiving and logging.

Controlling log compression

You can enable and disable the compression of log records using either

- The SET and DISPLAY LOG commands in MQSC; see [The MQSC commands](#)
- Invoking PCF interface. See [“Introdução aos formatos de comando programável do IBM MQ” on page 26](#)
- Using the CSQ6LOGP macro in the system parameter module; see [Using CSQ6LOGP](#)

Printing log records

You can extract and print log records using the CSQ1LOGP utility. For instructions, see [The log print utility](#).

Recovering logs

Normally, you do not need to back up and restore the IBM MQ logs, especially if you are using dual logging. However, in rare circumstances, such as an I/O error on a log, you might need to recover the logs. Use Access Method Services to delete and redefine the data set, and then copy the corresponding dual log into it.

Discarding archive log data sets

You can discard your archive log data sets and choose to discard the logs automatically or manually.

You must keep enough log data to be able to perform unit of work recovery, page set media recovery if a page set is lost, or CF structure media recovery if a CF structure is lost. Do not discard archive log data sets that might be required for recovery; if you discard these archive log data sets you might not be able to perform required recovery operations.

If you have confirmed that your archive log data sets can be discarded, you can do this in either of the following ways:

- [Automatic archive log data set deletion](#)
- [Manually deleting archive log data sets](#)

Automatic archive log data set deletion

You can use a DASD or tape management system to delete archive log data sets automatically. The retention period for IBM MQ archive log data sets is specified by the retention period field ARCRETN in the CSQ6ARVP installation macro (see the [Using CSQ6ARVP](#) for more information).

The default for the retention period specifies that archive logs are to be kept for 9999 days (the maximum).

Important: You can change the retention period but you must ensure that you can accommodate the number of backup cycles that you have planned for.

.

IBM MQ uses the retention period value as the value for the JCL parameter RETPD when archive log data sets are created.

The retention period set by the MVS™/DFP storage management subsystem (SMS) can be overridden by this IBM MQ parameter. Typically, the retention period is set to the smaller value specified by either IBM MQ or SMS. The storage administrator and IBM MQ administrator must agree on a retention period value that is appropriate for IBM MQ.

Note: IBM MQ does not have an automated method to delete information about archive log data sets from the BSDS, because some tape management systems provide external manual overrides of retention periods. Therefore, information about an archive log data set can still be in the BSDS long after the data set retention period has expired and the data set has been scratched by the tape management system. Conversely, the maximum number of archive log data sets might have been exceeded and the data from the BSDS might have been dropped before the data set has reached its expiration date.

If archive log data sets are deleted automatically, remember that the operation does not update the list of archive logs in the BSDS. You can update the BSDS with the change log inventory utility, as described in [“Changing the BSDS” on page 510](#). The update is not essential. Recording old archive logs wastes space in the BSDS, but does no other harm.

Manually deleting archive log data sets

You must keep all the log records as far back as the lowest RBA identified in messages CSQI024I and CSQI025I. This RBA is obtained using the DISPLAY USAGE command that you issued when creating a point of recovery using [Method 1: Full backup](#).

Read [Creating a point of recovery for non-shared resources before discarding any logs](#).

Locate and discard archive log data sets

Having established the minimum log RBA required for recovery, you can find archive log data sets that contain only earlier log records by performing the following procedure:

1. Use the print log map utility to print the contents of the BSDS. For an example of the output, see [The print log map utility](#).
2. Find the sections of the output titled ARCHIVE LOG COPY n DATA SETS. If you use dual logging, there are two sections. The columns labeled STARTRBA and ENDRBA show the range of RBAs contained in each volume. Find the volumes with ranges that include the minimum RBA you found with messages CSQI024I and CSQI025I. These are the earliest volumes you need to keep. If you are using dual-logging, there are two such volumes.

If no volumes have an appropriate range, one of the following cases applies:

- The minimum RBA has not yet been archived, and you can discard all archive log volumes.
- The list of archive log volumes in the BSDS wrapped around when the number of volumes exceeded the number allowed by the MAXARCH parameter of the CSQ6LOGP macro. If the BSDS does not register an archive log volume, that volume cannot be used for recovery. Therefore, consider adding information about existing volumes to the BSDS. For instructions, see [“Changes for archive logs” on page 512](#).

Also consider increasing the value of MAXARCH. For information, see the [Using CSQ6LOGP](#).

3. Delete any archive log data set or volume with an ENDRBA value that is less than the STARTRBA value of the earliest volume you want to keep. If you are using dual logging, delete both such copies.

Because BSDS entries wrap around, the first few entries in the BSDS archive log section might be more recent than the entries at the end. Look at the combination of date and time and compare their ages. Do not assume that you can discard all entries before the entry for the archive log containing the minimum LOGRBA.

Delete the data sets. If the archives are on tape, erase the tapes. If they are on DASD, run a z/OS utility to delete each data set. Then, if you want the BSDS to list only existing archive volumes, use the change log inventory utility (CSQJU003) to delete entries for the discarded volumes. See [“Changes for archive logs” on page 512](#) for an example.

The effect of log shunting

Long running transactions can cause unit of work log records which span log data sets. IBM MQ handles this scenario by using log shunting, a technique which moves the log records to optimize the quantity of log data retained, and queue manager restart time.

When a unit of work is considered to be long, a representation of each log record is written further down the log. This is known as *log shunting*. It is described more fully in [Log files](#).

The queue manager uses these shunted log records instead of the originals after a failure, to ensure unit of work integrity. There are two benefits to this:

- the quantity of log data which must be retained for unit of work coordination is reduced
- less log data must be traversed at queue manager restart time, so the queue manager is restarted more quickly

Shunted log records do not contain sufficient information for media recovery operations.

Data held in the log is used for two distinct purposes; media recovery and unit of work coordination. If a media failure occurs which affects either a CF structure or page set, the queue manager can recover the media to the point of failure by restoring a prior copy and updating this using data contained in the log.

Persistent activity performed in a unit of work is recorded on the log so that in the event of a failure, it can either be backed out or locks can be recovered on changed resources. The quantity of log data you need to retain to enable queue manager recovery is affected by these two elements.

For media recovery, you must retain sufficient log data to be able to perform media recovery from at least the most recent media copy and to be able to back out. (Your site may stipulate the ability to recover from older backups.) For unit of work integrity, you must retain the log data for your oldest in flight or indoubt units of work.

To assist you with managing the system, the queue manager detects old units of work at each log archive and reports them in messages CSQJ160 and CSQJ161. An internal task reads unit of work log information for these old units of work and rewrites it in a more succinct form to the current position in the log. Message CSQR026 indicates when this has happened. The MQSC command DISPLAY USAGE TYPE(DATASET) can also help you to manage the retention of log data. The command reports the following three pieces of recovery information:

1. How much of the log must be retained for unit of work recovery.
2. How much of the log must be retained for media recovery of page sets.
3. For a queue manager in a queue sharing group, how much of the log must be retained for media recovery of CF structures.

For each of these pieces of information, an attempt is made to map the oldest log data required into a data set. As new units of work start and stop, (1) would be expected to move to a more recent position in the log. If it is not moving, the long running UOW messages warn you that there is an issue. (2) relates to page set media recovery if the queue manager were to be shut down now and restarted. It does not know about when you last backed up your page sets, or which backup you might have to use if there was a page set failure. It normally moves to a more recent position in the log during checkpoint processing as changes held in the buffer pools are written to the page sets. In (3), the queue manager does know about CF structure backups taken either on this queue manager or on other queue managers in the queue sharing group. However, CF structure recovery requires a merge of log data from all queue managers in the queue sharing group which have interacted with the CF structure since the last backup. This means that the log data is identified by a log record sequence number, (or LRSN), which is timestamp based and so applicable across the entire queue sharing group rather than an RBA which would be different on different queue managers in the queue sharing group. It normally moves to a more recent position in the log as BACKUP CFSTRUCT commands are performed on either this or other queue managers in the queue sharing group.

Resetting the queue manager's log

Use this topic to understand how to reset the queue manager's log.

You must not allow the queue manager log RBA to wrap around from the end of the log RBA range to 0, as this leads to a queue manager outage and all persistent data will become unrecoverable. The end of the log RBA is either a value of FFFFFFFFFF (if 6-byte RBAs as in use), or FFFFFFFFFFFFFFFF (if 8-byte RBAs are in use).

The queue manager issues messages [CSQI045I](#), [CSQI046E](#), [CSQI047E](#), [CSQJ031D](#), and [CSQJ032E](#) to indicate that the used log range is significant and that you should plan to take action to avoid an unplanned outage.

The queue manager terminates with reason code [00D10257](#) when the RBA value reaches FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFFC0000000000 (if 8-byte log RBAs are in use).

If 6-byte log RBAs are in use, consider converting the queue manager to use 8-byte log RBAs rather than resetting the queue manager's log, following the process described in [“Implementing the larger log Relative Byte Address”](#) on page 508. Converting a queue manager to use 8-byte log RBAs requires a shorter outage than resetting the log, and increases the period of time before you have to reset the log.

Message [CSQJ034I](#), issued during queue manager initialization, indicates the end of the log RBA range for the queue manager as configured, and can be used to determine whether 6-byte or 8-byte log RBAs are in use.

The procedure to follow to reset the queue manager's log is as follows:

1. Resolve any unresolved units of work. The number of unresolved units of work is displayed at queue manager startup in message CSQR005I as the INDOUBT count. At each checkpoint, and at queue manager shutdown, the queue manager automatically issues the command **DISPLAY CONN(*) TYPE(CONN) ALL WHERE(UOWSTATE EQ UNRESOLVED)** to provide information about unresolved units of work.

See [How in-doubt units of recovery are resolved](#) for information on resolving units of recovery. The ultimate recourse is to use the **RESOLVE INDOUBT MQSC** command to manually resolve indoubt units of recovery.
2. Shut down the queue manager cleanly.

You can use either **STOP QMGR** or **STOP QMGR MODE(FORCE)** as both these commands flush any changed pages from bufferpools to the page sets.
3. If a queue manager is part of a queue sharing group, take CFSTRUCT backups on other queue managers for all structures in the queue sharing group. This ensures that the most recent backups are not in this queue manager's log, and that this queue manager's log is not required for CFSTRUCT recovery.
4. Define new logs and BSDS using CSQJU003 (see [The change log inventory utility](#) for more information on using the change log inventory utility).
5. Run **CSQUTIL RESETPAGE** against all the page sets for this queue manager (see [Copying a page and resetting the log](#) for more information on using this function). Note that page set RBAs can be reset independently, so multiple concurrent jobs (for example, one per page set) can be submitted to reduce the elapsed time for this step.
6. Restart the queue manager

Warning messages

When IBM MQ detects that the end of the log is approaching, it issues console messages in the following order, which indicate that a log reset should be planned. In this section the messages show 6-byte log RBA values. The same console messages are issued when IBM MQ is running in 8-byte log RBA mode but with different values; see [“Warning thresholds” on page 507](#) for the 8-byte log RBA thresholds.

1. When IBM MQ detects that the end of the log is approaching in the near future, (approximately 94% full) IBM MQ issues console message CSQI045I, as in the following example:

```
CSQI045I -CSQ7 CSQILCUR Log RBA has reached 0000F00000000000.  
Plan a log reset
```

2. IBM MQ issues the following CSQI046E error console message when the end of the log is near (approximately 97% full). This informs the IBM MQ administrator to take action soon.

```
CSQI046E -CSQ7 CSQILCUR Log RBA has reached 0000F80000000000.  
Perform a log reset
```

3. After the CSQI046E message is issued, at the next log switch, IBM MQ issues the following CSQJ032E console message with the word WARNING:

```
CSQJ032E -CSQ7 CSQJW307 WARNING - APPROACHING END OF  
THE LOG RBA RANGE OF 0000FFFFFFFF. CURRENT LOG RBA IS 0000F80000022000.
```

4. After the CSQI046E and CSQJ032E console messages are issued, IBM MQ issues one more error message, which does not require immediate IBM MQ administrator intervention. IBM MQ issues console message CSQI047E (when the log is approximately 99% full):

```
CSQI047E -CSQ7 CSQILCUR Log RBA has reached 0000FF0000000000.  
Stop queue manager and reset logs
```

5. When the log RBA reaches FF8000000000, IBM MQ increases the urgency of the situation and issues console message CSQJ032E with the word CRITICAL:

```
CSQJ032E -CSQ7 CSQJW009 CRITICAL - APPROACHING END OF THE LOG RBA RANGE OF 0000FFFFFFFFFFFF.
CURRENT LOG RBA IS 0000FFF7FFFFDFFF.
```

6. If the queue manager is started when the log RBA is almost at the maximum, the following CSQJ031D console message is issued. This stage requires the input of the IBM MQ administrator:.

```
CSQJ031D -CSQ7 CSQYSTRT THE LOG RBA RANGE MUST BE RESET.
REPLY 'Y' TO CONTINUE STARTUP OR 'N' TO SHUTDOWN
```

7. IBM MQ startup remains suspended until a reply is given to message CSQJ031D.

The purpose of these messages is to give the IBM MQ administrator time to plan for a system outage to reset the logs. In an ideal configuration, there are at least two queue managers, possibly in a queue sharing group (QSG), sharing the workload. When one is down for maintenance the other can continue to receive work.

The severity of console messages that IBM MQ issues becomes greater as the RBA gets closer to the end. Ideally your IBM MQ administrator should plan to reset the log RBA when the first console message is seen.

If the warning and error console messages are ignored, IBM MQ terminates with reason code 5C6-00D10257 when the log RBA reaches FFF800000000, at which point IBM MQ determines that the available range is too small for the queue manager to continue. When this point is reached, the only option is to take an outage and either reset the log or extend the size of the log RBA.

Note: When the end of the log is reached it is not possible to resolve any in-flight units of work (UOW); these are lost during the log reset process. Enough of the RBA range should be left to start the queue manager and resolve any UOW. Because IBM MQ issues console messages several times to inform that the end of the log is approaching, a log reset should be planned.

The preferred option to avoid losing any in-flight UOW is to extend the log RBA to use 8 bytes. This means that a log RBA reset will not be necessary for a long period.

Warning thresholds

The following table lists the thresholds, based on the length of the log RBA.

Console message	6-byte log RBA	8-byte log RBA
CSQI045I	0000F00000000000	FFFF800000000000
CSQI046E	0000F80000000000	FFFFC00000000000
CSQI047E	0000FF8000000000	FFFFFC0000000000
CSQJ032E	0000FF8000000000 0000FF8000000000	FFFFFC0000000000 FFFFFC0000000000
CSQJ031D	0000FF8000000000	FFFFFC0000000000

Notes:

1. For message CSQJ032E, the first number applies to the WARNING text and the second number applies to the CRITICAL text in the console message.
2. Message CSQJ031D is issued at IBM MQ initialization only.

Related concepts

[“Implementing the larger log Relative Byte Address” on page 508](#)

Before IBM MQ for z/OS 8.0, IBM MQ for z/OS used a 6 byte log RBA to identify the location of data within the log. From IBM MQ for z/OS 8.0, the log RBA can be 8 bytes long, increasing the period of time before you have to reset the log.

Implementing the larger log Relative Byte Address

Before IBM MQ for z/OS 8.0, IBM MQ for z/OS used a 6 byte log RBA to identify the location of data within the log. From IBM MQ for z/OS 8.0, the log RBA can be 8 bytes long, increasing the period of time before you have to reset the log.



Attention: You only have to carry out the following procedure to enable this feature if your queue managers were created before IBM MQ 9.3.0, as queue managers created at IBM MQ 9.3.0 and later already have this feature enabled.

See [Planning to increase the maximum addressable log range](#) for considerations when planning to enable 8 byte log RBA.

Perform these instructions, in the order shown, to enable 8 byte log RBA on a single IBM MQ for z/OS queue manager. For queue managers in a queue sharing group, perform the steps on each queue manager in turn.

1. Allocate new BSDS data sets with similar attributes to the current BSDS. You can tailor sample CSQ4BSDS and delete any irrelevant statement, or you can use your existing JCL, but change the BSDS name to something like ++HLQ++ .NEW .BSDS01.

Notes:

- a. Check the attributes of your new BSDS before submitting the job to allocate the new BSDS. The only attribute that might change is the size of the BSDS.
 - b. The new BSDS contains more data than the current BSDS, therefore, you must ensure that the new data sets are allocated with sufficient available space. The sample JCL in `thlqual.SCSQPROC(CSQ4BSDS)` contains the recommended values when defining a new BSDS.
2. Shut down the queue manager cleanly.
 3. Run the [BSDS conversion utility \(CSQJUCNV\)](#) to convert the existing BSDS to the new BSDS data sets. This usually takes a few seconds to run.

Your existing BSDS will not be changed during this process, and you can use that for the initialization of the queue manager in the case of an unsuccessful conversion.
 4. Rename the current BSDS to become the old BSDS, and the new BSDS to become the current BSDS, so that the new data sets are used when you next restart the queue manager. You can use the DFSMS Access Method Services ALTER command, for example:

```
ALTER '++HLQ++.BSDS01' NEWNAME('++HLQ++.OLD.BSDS01')
ALTER '++HLQ++.NEW.BSDS01' NEWNAME('++HLQ++.BSDS01')
```

Ensure that you also issue commands to rename both the data and index portions of the VSAM cluster.

5. Restart the queue manager. It should start in the same amount of time as it would have done when using 6 byte log RBA.

If the queue manager does not restart successfully due to a failure to access the converted BSDS, attempt to identify the cause of the failure, resolve the problem and retry the operation. If required, contact your IBM support center for assistance.

If necessary, the change can be backed out at this point by:

- a. Renaming the current BSDS to become the new BSDS.
 - b. Renaming the old BSDS to become the current BSDS.
 - c. Restarting the queue manager.
6. Once the queue manager has been successfully restarted with the converted BSDS, do not attempt to start the queue manager using the old BSDS.
 7. Message `CSQJ034I` is issued during queue manager initialization to indicate the end of the log RBA for the queue manager as configured. Confirm that the end of the log RBA range displayed is `FFFFFFFFFFFFFFFF`. This indicates that 8 byte log RBA is in use.

Related tasks

[Planning to increase the maximum addressable log range](#)

Related reference

[Larger log Relative Byte Address](#)

[The BSDS conversion utility \(CSQJUCNV\)](#)

Managing the bootstrap data set (BSDS)

The bootstrap data set (BSDS) is used to reference log data sets, and log records. Use this topic to understand how you can examine, change, and recover the BSDS.

For more information, see [The bootstrap data set](#).

This topic describes the tasks involved in managing the bootstrap data set. It contains these sections:

- [“Finding out what the BSDS contains” on page 509](#)
- [“Changing the BSDS” on page 510](#)
- [“Recovering the BSDS” on page 514](#)

Finding out what the BSDS contains

You can use the print log map utility (CSQJU004) to examine the contents of the BSDS.

The print log map utility (CSQJU004) is a batch utility that lists the information stored in the BSDS. For instructions on running it, see [The print log map utility](#).

The BSDS contains:

- [Time stamps](#)
- [Active log data set status](#)

Time stamps in the BSDS

The output of the print log map utility shows the time stamps, which are used to record the date and time of various system events, that are stored in the BSDS.

The following time stamps are included in the header section of the report:

SYSTEM TIMESTAMP

Reflects the date and time the BSDS was last updated. The BSDS time stamp can be updated when:

- The queue manager starts.
- The write threshold is reached during log write activities. Depending on the number of output buffers you have specified and the system activity rate, the BSDS might be updated several times a second, or might not be updated for several seconds, minutes, or even hours. For details of the write threshold, see the WRTHRS parameter of the CSQ6LOGP macro in [Using CSQ6LOGP](#).
- IBM MQ drops into a single BSDS mode from its normal dual BSDS mode due to an error. This can occur when a request to get, insert, point to, update, or delete a BSDS record is unsuccessful. When this error occurs, IBM MQ updates the time stamp in the remaining BSDS to force a time stamp mismatch with the disabled BSDS.

UTILITY TIMESTAMP

The date and time the contents of the BSDS were altered by the change log inventory utility (CSQJU003).

The following time stamps are included in the active and archive log data sets portion of the report:

Active log date

The date the active log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done.

Active log time

The time the active log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done.

Archive log date

The date the archive log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done or the archive itself was done.

Archive log time

The time the archive log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done or the archive itself was done.

Active log data set status

The BSDS records the status of an active log data set as one of the following:

NEW

The data set has been defined but never used by IBM MQ, or the log was truncated to a point before the data set was first used. In either case, the data set starting and ending RBA values are reset to zero.

REUSABLE

Either the data set has been defined but never used by IBM MQ, or the data set has been offloaded. In the print log map output, the start RBA value for the last REUSABLE data set is equal to the start RBA value of the last archive log data set.

NOT REUSABLE

The data set contains records that have not been offloaded.

STOPPED

The offload processor encountered an error while reading a record, and that record could not be obtained from the other copy of the active log.

TRUNCATED

Either:

- An I/O error occurred, and IBM MQ has stopped writing to this data set. The active log data set is offloaded, beginning with the starting RBA and continuing up to the last valid record segment in the truncated active log data set. The RBA of the last valid record segment is lower than the ending RBA of the active log data set. Logging is switched to the next available active log data set, and continues uninterrupted.

or

- An ARCHIVE LOG function has been called, which has truncated the active log.

The status appears in the output from the print log map utility.

 **Changing the BSDS**

You do not have to take special steps to keep the BSDS updated with records of logging events because IBM MQ does that automatically.

However, you might want to change the BSDS if you do any of the following:

- Add more active log data sets.
- Copy active log data sets to newly allocated data sets, for example, when providing larger active log allocations.
- Move log data sets to other devices.
- Recover a damaged BSDS.
- Discard outdated archive log data sets.

You can change the BSDS by running the change log inventory utility (CSQJU003). Only run this utility when the queue manager is inactive, or you might get inconsistent results. The action of the utility is controlled by statements in the SYSIN data set. This section shows several examples. For complete instructions, see [The change log inventory utility](#).

You can copy an active log data set only when the queue manager is inactive because IBM MQ allocates the active log data sets as exclusive (DISP=OLD) at queue manager startup.

Changes for active logs

Use this topic to understand how you can change the active logs using the BSDS.

You can add to, delete from, and record entries in the BSDS for active logs using the change log utility. Examples only are shown here; replace the data set names shown with the ones you want to use. For more details of the utility, see [The change log inventory utility](#).

See these sections for more information:

- [Adding record entries to the BSDS](#)
- [Deleting information about the active log data set from the BSDS](#)
- [Recording information about the log data set in the BSDS](#)
- [Increasing the size of the active log](#)
- [The use of CSQJUFMT](#)

Adding record entries to the BSDS

If an active log has been flagged as "stopped", it is not reused for logging; however, it continues to be used for reading. Use the access method services to define new active log data sets, then use the change log inventory utility to register the new data sets in the BSDS. For example, use:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY1
NEWLOG DSNAMES=MQM111.LOGCOPY2.DS10,COPY2
```

If you are copying the contents of an old active log data set to the new one, you can also give the RBA range and the starting and ending time stamps on the NEWLOG function.

Deleting information about the active log data set from the BSDS

To delete information about an active log data set from the BSDS, you could use:

```
DELETE DSNAMES=MQM111.LOGCOPY1.DS99
DELETE DSNAMES=MQM111.LOGCOPY2.DS99
```

Recording information about the log data set in the BSDS

To record information about an existing active log data set in the BSDS, use:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY2,STARTIME=19930212205198,
ENDTIME=19930412205200,STARTRBA=6400,ENDRBA=94FF
```

You might need to insert a record containing this type of information in the BSDS because:

- The entry for the data set has been deleted, but is needed again.
- You are copying the contents of one active log data set to another data set.
- You are recovering the BSDS from a backup copy.

Increasing the size of the active log

There are two methods of achieving this process.

1. When the queue manager is active:
 - a. Define new larger log data sets using JCL.
 - b. Add the new log data sets to the active queue manager using the MQSC DEFINE LOG command.
 - c. Use the MQSC ARCHIVE LOG command to move the current active log, to be a new larger log.
 - d. Wait for the archive of the smaller active log data set to complete.
 - e. Shut down the queue manager, using the CSQJU003 utility to remove the old small active logs.
 - f. Restart the queue manager.
2. When the queue manager is inactive:
 - a. Stop the queue manager. This step is required because IBM MQ allocates all active log data sets for its exclusive use when it is active.
 - b. Use Access Method Services ALTER with the NEWNAME option to rename your active log data sets.
 - c. Use Access Method Services DEFINE to define larger active log data sets.

By reusing the old data set names, you do not have to run the change log inventory utility to establish new names in the BSDSs. The old data set names and the correct RBA ranges are already in the BSDSs.
 - d. Use Access Method Services REPRO to copy the old (renamed) data sets into their appropriate new data sets.

Note: This step can take a long time, so your enterprise could be out of action for this period.
 - e. Start the queue manager.

If all your log data sets are the same size, your system will be operationally more consistent and efficient. If the log data sets are not the same size, it is more difficult to track your system's logs, and so space can be wasted.

The use of CSQJUFMT

Do not run a CSQJUFMT format when increasing the size of an active log.

If you run CSQJUFMT (in order to provide a performance advantage the first time the queue manager writes to the new active log) you receive messages:

```
IEC070I 203-204,XS95GTLX,REPRO02,OUTPUT,B857,SPMG02, 358
IEC070I MG.W.MG4E.LOGCOPY1.DS02,MG.W.MG4E.LOGCOPY1.DS02.DATA,
IDC3302I ACTION ERROR ON MG.W.MG4E.LOGCOPY1.DS02
IDC3351I ** VSAM I/O RETURN CODE IS 28 - RPLFDBWD = X'2908001C'
IDC31467I MAXIMUM ERROR LIMIT REACHED.

IDC0005I NUMBER OF RECORDS PROCESSED WAS 0
```

In addition, if you use the Access Method Services REPRO, ensure that you define a new empty log.

If you use REPRO to copy the old (renamed) data set into its respective new data set, the default is NOREPLACE.

This means that REPRO does not replace a record that is already on the designated data set. When formatting is done on the data set, the RBA value is reset. The net result is a data set that is not empty after formatting.

Changes for archive logs

Use this topic to understand how to change the archive logs.

You can add to, delete from, and change the password of, entries in the BSDS for archive logs. Examples only are shown here; replace the data set names shown with the ones you want to use. For more details of the utility, see [The change log inventory utility](#).

- [Adding an archive log](#)
- [Deleting an archive log](#)
- [Changing the password of an archive log](#)

Adding an archive log

When the recovery of an object depends on reading an existing archive log data set, the BSDS must contain information about that data set so that IBM MQ can find it. To register information about an existing archive log data set in the BSDS, use:

```
NEWLOG DSNAME=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04,  
UNIT=TAPE,STARTRBA=3A190000,ENDRBA=3A1F0FFF,CATALOG=NO
```

Deleting an archive log

To delete an entire archive log data set on one or more volumes, use:

```
DELETE DSNAME=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04
```

Changing the password of an archive log

If you change the password of an existing archive log data set, you must also change the information in the BSDS.

1. List the BSDS, using the print log map utility.
2. Delete the entry for the archive log data set with the changed password, using the DELETE function of the CSQJU003 utility (see topic [The change log inventory utility](#)).
3. Name the data set as for a new archive log data set. Use the NEWLOG function of the CSQJU003 utility (see topic [The change log inventory utility](#)), and give the new password, the starting and ending RBAs, and the volume serial numbers (which can be found in the print log map utility output, see [The print log map utility](#)).

To change the password for new archive log data sets, use:

```
ARCHIVE PASSWORD= password
```

To stop placing passwords on new archive log data sets, use:

```
ARCHIVE NOPASSWD
```

Note: Only use the ARCHIVE utility function if you do not have an external security manager.

 **Changing the high-level qualifier (HLQ) for the logs and BSDS**

Use this topic to understand the procedure required to change the high-level qualifier (HLQ).

Before you begin

You must end the queue manager normally before copying any of the logs or data sets to the new data sets. This is to ensure that the data is consistent and no recovery is needed during restart.

About this task

This task provides information about how to change the HLQ for the logs and BSDS. To do this, follow these steps:

Procedure

1. Run the log print utility CSQJU004 to record the log data set information. This information is needed later.
2. You can either:
 - a) run DSS backup and restore with rename on the log and BSDS data sets to be renamed, or
 - b) use AMS DEFINE and REPRO to create the HLQ data sets and copy the data from the old data sets.
3. Modify the MSTR and CHIN procedures to point to the new data sets.
4. Delete the old log information in the new copy of the BSDS using CSQJU003.
5. Define the new log data sets to the new BSDS using the NEWLOG function of CSQJU003.
Keep all information about each log the same, apart from the HLQ.
6. The new BSDS should reflect the same information that was recorded for the old logs in the old BSDS.
The HLQ should be the only thing that has changed.

What to do next

Compare the CSQJU004 output for the old and new BSDS to ensure that they look EXACTLY the same (except for the HLQs) before starting the queue manager.

Note: Care must be taken when performing these operations. Incorrect actions might lead to unrecoverable situations. Check the PRINT LOG MAP UTILITY output and make sure that all the information needed for recovery or restart has been included.

Recovering the BSDS

If IBM MQ is operating in dual BSDS mode and one BSDS becomes damaged, forcing IBM MQ into single BSDS mode, IBM MQ continues to operate without a problem (until the next restart).

To return the environment to dual BSDS mode:

1. Use Access Method Services to rename or delete the damaged BSDS and to define a new BSDS with the same name as the damaged BSDS. Example control statements can be found in job CSQ4BREC in thlqual.SCSQPROC.
2. Issue the IBM MQ command RECOVER BSDS to make a copy of the valid BSDS in the newly allocated data set and to reinstate dual BSDS mode.

If IBM MQ is operating in single BSDS mode and the BSDS is damaged, or if IBM MQ is operating in dual BSDS mode and both BSDSs are damaged, the queue manager stops and does not restart until the BSDS data sets are repaired. In this case:

1. Locate the BSDS associated with the most recent archive log data set. The data set name of the most recent archive log appears on the job log in the last occurrence of message CSQJ003I, which indicates that offload processing has been completed successfully. In preparation for the rest of this procedure, it is a good practice to keep a log of all successful archives noted by that message:
 - If archive logs are on DASD, the BSDS is allocated on any available DASD. The BSDS name is like the corresponding archive log data set name; change only the first letter of the last qualifier, from A to B, as in this example:

Archive log nameCSQ.ARCHLOG1. **A** 0000001**BSDS copy name**CSQ.ARCHLOG1. **B** 0000001

- If archive logs are on tape, the BSDS is the first data set of the first archive log volume. The BSDS is not repeated on later volumes.
- 2. If the most recent archive log data set has no copy of the BSDS (for example, because an error occurred when offloading it), locate an earlier copy of the BSDS from earlier offload processing.
- 3. Rename *damaged* BSDSs using the Access Method Services ALTER command with the NEWNAME option. If you want to delete a damaged BSDS, use the Access Method Services DELETE command. For each damaged BSDS, use Access Method Services to define a new BSDS as a replacement data set. Job CSQ4BREC in thlqual.SCSQPROC contains Access Method Services control statements to define a new BSDS.
- 4. Use the Access Method Services REPRO command to copy the BSDS from the archive log to one of the replacement BSDSs you defined in step “3” on page 515. Do not copy any data to the second replacement BSDS, you do that in step “5” on page 516.

- a. Print the contents of the replacement BSDS.

Use the print log map utility (CSQJU004) to print the contents of the replacement BSDS. This enables you to review the contents of the replacement BSDS before continuing your recovery work.

- b. Update the archive log data set inventory in the replacement BSDS.

Examine the output from the print log map utility and check that the replacement BSDS does not contain a record of the archive log from which the BSDS was copied. If the replacement BSDS is an old copy, its inventory might not contain all archive log data sets that were created more recently. The BSDS inventory of the archive log data sets must be updated to reflect the current subsystem inventory.

Use the change log inventory utility (CSQJU003) NEWLOG statement to update the replacement BSDS, adding a record of the archive log from which the BSDS was copied. If the archive log data set is password-protected, use the PASSWORD option of the NEWLOG function. Also, if the archive log data set is cataloged, ensure that the CATALOG option of the NEWLOG function is properly set to CATALOG=YES. Use the NEWLOG statement to add any additional archive log data sets that were created later than the BSDS copy.

- c. Update passwords in the replacement BSDS.

The BSDS contains passwords for the archive log data sets and for the active log data sets. To ensure that the passwords in the replacement BSDS reflect the current passwords used by your installation, use the change log inventory ARCHIVE utility function with the PASSWORD option.

- d. Update the active log data set inventory in the replacement BSDS.

In unusual circumstances, your installation might have added, deleted, or renamed active log data sets since the BSDS was copied. In this case, the replacement BSDS does not reflect the actual number or names of the active log data sets your installation currently has in use.

If you need to delete an active log data set from the replacement BSDS log inventory, use the change log inventory utility DELETE function.

If you need to add an active log data set to the replacement BSDS log inventory, use the change log inventory utility NEWLOG function. Ensure that the RBA range is specified correctly on the NEWLOG function. If the active log data set is password-protected, use the PASSWORD option.

If you need to rename an active log data set in the replacement BSDS log inventory, use the change log inventory utility DELETE function, followed by the NEWLOG function. Ensure that the RBA range is specified correctly on the NEWLOG function. If the active log data set is password-protected, use the PASSWORD option.

- e. Update the active log RBA ranges in the replacement BSDS.

Later, when the queue manager restarts, it compares the RBAs of the active log data sets listed in the BSDS with the RBAs found in the actual active log data sets. If the RBAs do not agree, the queue manager does not restart. The problem is magnified when an old copy of the BSDS is used. To solve this problem, use the change log inventory utility (CSQJU003) to adjust the RBAs found in the BSDS using the RBAs in the actual active log data sets. You do this by:

- Using the print log records utility (CSQ1LOGP) to print a summary report of the active log data set. This shows the starting and ending RBAs.
- Comparing the actual RBA ranges with the RBA ranges you have just printed, when the RBAs of all active log data sets are known.

If the RBA ranges are equal for all active log data sets, you can proceed to the next recovery step without any additional work.

If the RBA ranges are not equal, adjust the values in the BSDS to reflect the actual values. For each active log data set that needs to have the RBA range adjusted, use the change log inventory utility DELETE function to delete the active log data set from the inventory in the replacement BSDS. Then use the NEWLOG function to redefine the active log data set to the BSDS. If the active log data sets are password-protected, use the PASSWORD option of the NEWLOG function.

- f. If only two active log data sets are specified for each copy of the active log, IBM MQ can have difficulty during queue manager restart. The problem can arise when one of the active log data sets is full and has not been offloaded, while the second active log data set is close to filling. In this case, add a new active log data set for each copy of the active log and define each new active log data set in the replacement BSDS log inventory.

Use the Access Method Services DEFINE command to define a new active log data set for each copy of the active log and use the change log inventory utility NEWLOG function to define the new active log data sets in the replacement BSDS. You do not need to specify the RBA ranges on the NEWLOG statement. However, if the active log data sets are password-protected, use the PASSWORD option of the NEWLOG function. Example control statements to accomplish this task can be found in job CSQ4LREC in thlqual.SCSQPROC.

5. Copy the updated BSDS to the second new BSDS data set. The BSDSs are now identical.

Use the print log map utility (CSQJU004) to print the contents of the second replacement BSDS at this point.

6. See [Active log problems](#) for information about what to do if you have lost your current active log data set.
7. Restart the queue manager using the newly constructed BSDS. IBM MQ determines the current RBA and what active logs need to be archived.

Managing page sets

Use this topic to understand how to manage the page sets associated with a queue manager.

This topic describes how to add, copy, and generally manage the page sets associated with a queue manager. It contains these sections:

- [“How to change the high-level qualifier \(HLQ\) for the page sets” on page 517](#)
- [“How to add a page set to a queue manager” on page 517](#)
- [“What to do when one of your page sets becomes full” on page 517](#)
- [“How to balance loads on page sets” on page 518](#)
- [How to increase the size of a page set](#)
- [“How to reduce a page set” on page 521](#)
- [“How to reintroduce a page set” on page 522](#)

- [“How to back up and recover page sets” on page 523](#)
- [“How to delete page sets” on page 526](#)
- [“How to back up and restore queues using CSQUTIL” on page 526](#)

See [Page sets](#) for a description of page sets, storage classes, buffers, and buffer pools, and some of the performance considerations that apply.

How to change the high-level qualifier (HLQ) for the page sets

This task gives information on how to change the HLQ for the page sets. To perform this task, do the following:

1. Define the new HLQ page sets.
2. If the size allocation is the same as the old page sets, copy the existing page set using REPRO to the empty new HLQ page sets.
3. If you are increasing the size of the page sets, use the FORMAT function of CSQUTIL to format the destination pages, and then the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set.

For more information, see [Formatting page sets \(FORMAT\)](#), and [Expanding a page set \(COPYPAGE\)](#).

4. Change the CSQP00xx DD statement in the queue manager procedure to point to the new HLQ page sets.

Restart the queue manager and verify the changes to the page sets.

How to add a page set to a queue manager

This description assumes that you have a queue manager that is already running. You might need to add a page set if, for example, your queue manager has to cope with new applications using new queues.

To add a new page set, use the following procedure:

1. Define and format the new page set. You can use the sample JCL in thlqual.SCSQPROC(CSQ4PAGE) as a basis. For more information, see [Formatting page sets \(FORMAT\)](#).

Take care not to format any page sets that are in use, unless this is what you intend. If so, use the FORCE option of the FORMAT utility function.

2. Use the DEFINE PSID command with the DSN option to associate the page set with a buffer pool.
3. Add the appropriate storage class definitions for your page set by issuing DEFINE STGCLASS commands.
4. Optionally, to document how your queue manager is configured:
 - a. Add the new page set to the started task procedure for your queue manager.
 - b. Add a definition for the new page set to your CSQINP1 initialization data set.
 - c. Add a definition for the new storage class to your CSQ4INZR initialization data set member.

For details of the DEFINE PSID and DEFINE STGCLASS commands, see [DEFINE PSID](#) and [DEFINE STGCLASS](#).

What to do when one of your page sets becomes full

You can find out about the utilization of page sets by using the IBM MQ command DISPLAY USAGE. For example, the command:

```
DISPLAY USAGE PSID(03)
```

displays the current state of the page set 03. This tells you how many free pages this page set has.

If you have defined secondary extents for your page sets, they are dynamically expanded each time they fill up. Eventually, all secondary extents are used, or no further disk space is available. If this happens, an application receives the return code MQRC_STORAGE_MEDIUM_FULL.

If an application receives a return code of MQRC_STORAGE_MEDIUM_FULL from an MQI call, this is a clear indication that there is not enough space remaining on the page set. If the problem persists or is likely to recur, you must do something to solve it.

You can approach this problem in a number of ways:

- Balance the load between page sets by moving queues from one page set to another.
- Expand the page set. See [“How to increase the size of a page set”](#) on page 520 for instructions.
- Redefine the page set so that it can expand beyond 4 GB to a maximum size of 64 GB. See [Defining a page set to be larger than 4 GB](#) for instructions.

How to balance loads on page sets

Load balancing on page sets means moving the messages associated with one or more queues from one page set to another, less used, page set. Use this technique if it is not practical to expand the page set.

To identify which queues are using a page set, use the appropriate IBM MQ commands. For example, to find out which queues are mapped to page set 02, first, find out which storage classes map to page set 02, by using the command:

```
DISPLAY STGCLASS(*) PSID(02)
```

Then use the following command to find out which queues use which storage class:

```
DISPLAY QUEUE(*) TYPE(QLOCAL) STGCLASS
```

Moving a non-shared queue

To move queues and their messages from one page set to another, use the MQSC MOVE QLOCAL command (described in MOVE QLOCAL). When you have identified the queue or queues that you want to move to a new page set, follow this procedure for each of these queues:

1. Ensure that the queue you want to move is not in use by any applications (that is, IPPROCS and OPPOCS values from the DISPLAY QSTATUS command are zero) and that it has no uncommitted messages (the UNCOM value from the DISPLAY QSTATUS command is NO).

Note: The only way to ensure that this state continues is to change the security authorization of the queue temporarily. See [Profiles for queue security](#) for more information.

If you cannot do this, later stages in this procedure might fail if applications start to use the queue despite precautionary steps such as setting PUT(DISABLED). However, messages can never be lost by this procedure.

2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Change the queue definition to PUT(DISABLED).
3. Define a temporary queue with the same attributes as the queue that is being moved, using the command:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
```

Note: If this temporary queue already exists from a previous run, delete it before doing the define.

4. Move the messages to the temporary queue using the following command:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Delete the queue you are moving, using the command:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Define a new storage class that maps to the required page set, for example:

```
DEFINE STGCLASS(NEW) PSID(nn)
```

Add the new storage class definition to the CSQINP2 data sets ready for the next queue manager restart.

7. Redefine the queue that you are moving, by changing the storage class attribute:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
```

When the queue is redefined, it is based on the temporary queue created in step “3” on page 518.

8. Move the messages back to the new queue, using the command:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. The queue created in step “3” on page 518 is no longer required. Use the following command to delete it:

```
DELETE QL(TEMP_QUEUE)
```

10. If the queue being moved was defined in the CSQINP2 data sets, change the STGCLASS attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Add the REPLACE keyword so that the existing queue definition is replaced.

[Figure 29 on page 520](#) shows an extract from a load balancing job.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
//      DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE STGCLASS(NEW) PSID(2)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*

```

Figure 29. Extract from a load balancing job for a page set

How to increase the size of a page set

You can initially allocate a page set larger than 4 GB, See [Defining a page set to be larger than 4 GB](#)

A page set can be defined to be automatically expanded as it becomes full by specifying EXPAND(SYSTEM) or EXPAND(USER). If your page set was defined with EXPAND(NONE), you can expand it in either of two ways:

- Alter its definition to allow automatic expansion. See [Altering a page set to allow automatic expansion](#)
- Create a new, larger page set and copy the messages from the old page set to the new one. See [Moving messages to a new, larger page set](#)

Defining a page set to be larger than 4 GB

IBM MQ can use a page set up to 64 GB in size, provided the data set is defined with 'extended addressability' to VSAM. Extended addressability is an attribute which is conferred by an SMS data class.

Note: Page sets and active log data sets are eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV) and, from z/OS V1.12, an archive log dataset can also reside in the EAS.

In the example shown in the following sample JCL, the management class 'EXTENDED' is defined to SMS with 'Extended addressability'. If your existing page set is not currently defined as having extended addressability, use the following method to migrate to an extended addressability format data set.

1. Stop the queue manager.
2. Use Access Method Services to rename the existing page set.
3. Define a destination page set, the same size as the existing page set, but with DATACLAS(EXTENDED).

Note: Extended-format data sets must be SMS managed. These are the mechanisms for requesting extended format for VSAM data sets:

- Using a data class that has a DSNTYPE value of EXT and the subparameter R or P to indicate required or preferred.
- Coding DSNTYPE=EXTREQ (extended format is required) or DSNTYPE=EXTPREF (extended format is preferred) on the DD statement.

- Coding the LIKE= parameter on the DD statement to refer to an existing extended format data set.

For more information, see [Restrictions on Defining Extended-Format Data Sets](#).

4. Use the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set. See [Expanding a page set \(COPYPAGE\)](#) for more details.
5. Restart the queue manager.
6. Alter the page set to use system expansion, to allow it to continue growing beyond its current allocation.

The following JCL shows example Access Method Services commands:

```
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER 'VICY.CSQ1.PAGE01' -
NEWNAME('VICY.CSQ1.PAGE01.OLD')
ALTER 'VICY.CSQ1.PAGE01.DATA' -
NEWNAME('VICY.CSQ1.PAGE01.DATA.OLD')
DEFINE CLUSTER (NAME('VICY.CSQ1.PAGE01') -
MODEL('VICY.CSQ1.PAGE01.OLD') -
DATACLAS(EXTENDED))
/*
```

Altering a page set to allow automatic expansion

Use the ALTER PSID command with the EXPAND(USER) or EXPAND(SYSTEM) options. See [ALTER PSID](#) and [Expanding a page set \(COPYPAGE\)](#) for general information on expanding page sets.

Moving messages to a new, larger page set

This technique involves stopping and restarting the queue manager. This deletes any nonpersistent messages that are not on shared queues at restart time. If you have nonpersistent messages that you do not want to be deleted, use load balancing instead. For more details, see [“How to balance loads on page sets” on page 518](#). In this description, the page set that you want to expand is referred to as the *source* page set; the new, larger page set is referred to as the *destination* page set.

Follow these steps:

1. Stop the queue manager.
2. Define the destination page set, ensuring that it is larger than the source page set, with a larger secondary extent value.
3. Use the FORMAT function of CSQUTIL to format the destination page set. See [Formatting page sets \(FORMAT\)](#) for more details.
4. Use the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set. See [Expanding a page set \(COPYPAGE\)](#) for more details.
5. Restart the queue manager using the destination page set by doing one of the following:
 - Change the queue manager started task procedure to reference the destination page set.
 - Use Access Method Services to delete the source page set and then rename the destination page set, giving it the same name as that of the source page set.

Attention:

Before you delete any IBM MQ page set, be sure that you have made the required backup copies.

How to reduce a page set

Prevent all users, other than the IBM MQ administrator, from using the queue manager. For example; by changing the access security settings.

If you have a large page set that is mostly empty (as shown by the DISPLAY USAGE command), you might want to reduce its size. The procedure to do this involves using the COPY, FORMAT, and LOAD functions of CSQUTIL (see [IBM MQ utility program](#)). This procedure does not work for page set zero (0), as it is not practical to reduce the size of this page set; the only way to do so is by reinitializing your queue manager (see [“Reinitializing a queue manager” on page 544](#)). The prerequisite of this procedure is to try and remove all users from the system so that all UOWs are complete and the page sets are consistent.

1. Use the STOP QMGR command with the QUIESCE or FORCE attribute to stop the queue manager.
2. Run the SCOPY function of CSQUTIL with the PSID option, to copy all message data from the large page set and save them in a sequential data set.
3. Define a new smaller page set data set to replace the large page set.
4. Run the FORMAT TYPE(NEW) function of CSQUTIL against the page set that you created in step [“3” on page 522](#).
5. Restart the queue manager using the page set created in step [“3” on page 522](#).
6. Run the LOAD function of CSQUTIL to load back all the messages saved during step [“2” on page 522](#).
7. Allow all users access to the queue manager.
8. Delete the old large page set.

How to reintroduce a page set

In certain scenarios it is useful to be able to bring an old page set online again to the queue manager. Unless specific action is taken, when the old page set is brought online the queue manager will recognize that the page set recovery RBA stored in the page set itself and in the checkpoint records is old, and will therefore automatically start media recovery of the page set to bring it up to date.

Such media recovery can only be performed at queue manager restart, and is likely to take a considerable length of time, especially if archive logs held on tape must be read. However, normally in this circumstance, the page set has been offline for the intervening period and so the log contains no information pertinent to the page set recovery.

The following three choices are available:

Allow full media recovery to be performed.

1. Stop the queue manager.
2. Ensure definitions are available for the page set in both the started task procedure for the queue manager and in the CSQINP1 initialization data set.
3. Restart the queue manager.

Allow any messages on the page set to be destroyed.

This choice is useful where a page set has been offline for a long time (some months, for example) and it has now been decided to reuse it for a different purpose.

1. Format the page set using the FORMAT function of CSQUTIL with the TYPE(NEW) option.
2. Add definitions for the page set to both the started task procedure for the queue manager and the CSQINP1 initialization data set.
3. Restart the queue manager.

Using the TYPE(NEW) option for formatting clears the current contents of the page set and tells the queue manager to ignore any historical information in the checkpoint about the page set.

Bring the page set online avoiding the media recovery process.

Use this technique only if you are sure that the page set has been offline since a clean shutdown of the queue manager. This choice is most appropriate where the page set has been offline for a short period, typically due to operational issues such as a backup running while the queue manager is being started.

1. Format the page set using the FORMAT function of CSQUTIL with the TYPE(REPLACE) option.

2. Either add the page set back into the queue manager dynamically using the DEFINE PSID command with the DSN option or allow it to be added at a queue manager restart.

Using the TYPE(REPLACE) option for formatting checks that the page set was cleanly closed by the queue manager, and marks it so that media recovery will not be performed. No other changes are made to the contents of the page set.

How to back up and recover page sets

There are different mechanisms available for back up and recovery. Use this topic to understand these mechanisms.

This section describes the following topics:

- [“Creating a point of recovery for non-shared resources” on page 523](#)
- [“Backing up page sets” on page 524](#)
- [“Recovering page sets” on page 525](#)
- [How to delete page sets](#)

For information about how to create a point of recovery for shared resources, see [“Recovering shared queues” on page 531](#).

Creating a point of recovery for non-shared resources

IBM MQ can recover objects and non-shared persistent messages to their current state if both:

1. Copies of page sets from an earlier point exist.
2. All the IBM MQ logs are available to perform recovery from that point.

These represent a point of recovery for non-shared resources.

Both objects and messages are held on page sets. Multiple objects and messages from different queues can exist on the same page set. For recovery purposes, objects and messages cannot be backed up in isolation, so a page set must be backed up as a whole to ensure the correct recovery of the data.

The IBM MQ recovery log contains a record of all persistent messages and changes made to objects. If IBM MQ fails (for example, due to an I/O error on a page set), you can recover the page set by restoring the backup copy and restarting the queue manager. IBM MQ applies the log changes to the page set from the point of the backup copy.

There are two ways of creating a point of recovery:

Full backup

Stop the queue manager, which forces all updates on to the page sets.

This allows you to restart from the point of recovery, using only the backed up page set data sets and the logs from that point on.

Fuzzy backup

Take *fuzzy* backup copies of the page sets without stopping the queue manager.

If you use this method, and your associated logs later become damaged or lost, you cannot use the fuzzy page set backup copies to recover. This is because the fuzzy page set backup copies contain an inconsistent view of the state of the queue manager and are dependent on the logs being available. If the logs are not available, you need to return to the last set of backup page set copies taken while the subsystem was inactive ([Method 1](#)) and accept the loss of data from that time.

Method 1: Full backup

This method involves shutting the queue manager down. This forces all updates on to the page sets so that the page sets are in a consistent state.

1. Stop all the IBM MQ applications that are using the queue manager (allowing them to complete first). This can be done by changing the access security or queue settings, for example.
2. When all activity has completed, display and resolve any in-doubt units of recovery. (Use the commands `DISPLAY CONN` and `RESOLVE INDOUBT`, as described in [DISPLAY CONN](#) and [RESOLVE INDOUBT](#).)
This brings the page sets to a consistent state; if you do not do this, your page sets might not be consistent, and you are effectively doing a fuzzy backup.
3. Issue the `ARCHIVE LOG` command to ensure that the latest log data is written out to the log data sets.
4. Issue the `STOP QMGR MODE(QUIESCE)` command. Record the lowest RBA value in the `CSQI024I` or `CSQI025I` messages (see [CSQI024I](#) and [CSQI025I](#) for more information). You should keep the log data sets starting from the one indicated by the RBA value up to the current log data set.
5. Take backup copies of all the queue manager page sets (see [“Backing up page sets”](#) on page 524).

Method 2: Fuzzy backup

This method does not involve shutting the queue manager down. Therefore, updates might be in virtual storage buffers during the backup process. This means that the page sets are not in a consistent state, and can only be used for recovery with the logs.

1. Issue the `DISPLAY USAGE TYPE(ALL)` command, and record the RBA value in the `CSQI024I` or `CSQI025I` messages (see [CSQI024I](#) and [CSQI025I](#) for more information).
2. Take backup copies of the page sets (see [“Backing up page sets”](#) on page 524).
3. Issue the `ARCHIVE LOG` command, to ensure that the latest log data is written out to the log data sets. To restart from the point of recovery, you must keep the log data sets starting from the log data set indicated by the RBA value up to the current log data set.

Backing up page sets

To recover a page set, IBM MQ needs to know how far back in the log to go. IBM MQ maintains a log RBA number in page zero of each page set, called the *recovery log sequence number* (LSN). This number is the starting RBA in the log from which IBM MQ can recover the page set. When you back up a page set, this number is also copied.

If the copy is later used to recover the page set, IBM MQ must have access to all the log records from this RBA value to the current RBA. That means you must keep enough of the log records to enable IBM MQ to recover from the oldest backup copy of a page set you intend to keep.

Use `ADRDSSU COPY` function to copy the page sets.

For more information, see the [COPY DATASET Command Syntax for Logical Data Set](#) documentation .

For example:

```
//STEP2 EXEC PGM=ADRDSSU,REGION=6M
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
COPY -
DATASET(INCLUDE(SCENDATA.MQPA.PAGESET.*)) -
RENAMEU(SCENDATA.MQPA.PAGESET.** ,SCENDATA.MQPA.BACKUP1.** ) -
SPHERE -
REPUNC -
FASTRPLICATION(PREF ) -
CANCELERROR -
TOL(ENQF)
/*
//
```

If you copy the page set while the queue manager is running you must use a copy utility that copies page zero of the page set first. If you do not do this you could corrupt the data in your page set.

If the process of dynamically expanding a page set is interrupted, for example by power to the system being lost, you can still use ADRDSSU to take a backup of a page set.

If you perform an Access Method Services IDCAMS LISTCAT ENT('page set data set name') ALLOC, you will see that the HI-ALLOC-RBA is higher than the HI-USED-RBA.

The next time this page set fills up it is extended again, if possible, and the pages between the high used RBA and the highest allocated RBA are used, along with another new extent.

Backing up your object definitions

You should also back up copies of your object definitions. To do this, use the MAKEDEF feature of the CSQUTIL COMMAND function (described in [Issuing commands to IBM MQ \(COMMAND\)](#)).

Back up your object definitions whenever you take a backup copy of your queue manager, and keep the most current version.

Recovering page sets

If the queue manager has terminated due to a failure, the queue manager can normally be restarted with all recovery being performed during restart. However, such recovery is not possible if any of your page sets or log data sets are not available. The extent to which you can now recover depends on the availability of backup copies of page sets and log data sets.

To restart from a point of recovery you must have:

- A backup copy of the page set that is to be recovered.
- If you used the "fuzzy" backup process described in ["Method 2: Fuzzy backup"](#) on page 524, the log data set that included the recorded RBA value, the log data set that was made by the ARCHIVE LOG command, and all the log data sets between these.
- If you used full backup, but you do not have the log data sets following that made by the ARCHIVE LOG command, you do **not** need to run the FORMAT TYPE(REPLACE) function of the CSQUTIL utility against all your page sets.

To recover a page set to its current state, you must also have all the log data sets and records since the ARCHIVE LOG command.

There are two methods for recovering a page set. To use either method, the queue manager must be stopped.

Simple recovery

This is the simpler method, and is appropriate for most recovery situations.

1. Delete the page set you want to restore from backup.
2. Use the ADRDSSU COPY function to recover your page set from the backup copy..

Alternatively, you can rename your backup copy to the original name, or change the CSQP00xx DD statement in your queue manager procedure to point to your backup page set. However, if you then lose or corrupt the page set, you will no longer have a backup copy to restore from.

3. Restart the queue manager.
4. When the queue manager has restarted successfully, you can restart your applications
5. Reinstate your normal backup procedures for the restored page.

Advanced recovery

This method provides performance advantages if you have a large page set to recover, or if there has been much activity on the page set since the last backup copy was taken. However, it requires more manual intervention than the simple method, which might increase the risk of error and the time taken to perform the recovery.

1. Delete and redefine the page set you want to restore from backup.
2. Use ADRDSSU to copy the backup copy of the page set into the new page set. Define your new page set with a secondary extent value so that it can be expanded dynamically.
Alternatively, you can rename your backup copy to the original name, or change the CSQP00xx DD statement in your queue manager procedure to point to your backup page set. However, if you then lose or corrupt the page set, you will no longer have a backup copy to restore from.
3. Change the CSQINP1 definitions for your queue manager to make the buffer pool associated with the page set being recovered as large as possible. By making the buffer pool large, you might be able to keep all the changed pages resident in the buffer pool and reduce the amount of I/O to the page set.
4. Restart the queue manager.
5. When the queue manager has restarted successfully, stop it (using quiesce) and then restart it using the normal buffer pool definition for that page set. After this second restart completes successfully, you can restart your applications
6. Reinstate your normal backup procedures for the restored page.

What happens when the queue manager is restarted

When the queue manager is restarted, it applies all changes made to the page set that are registered in the log, beginning at the restart point for the page set. IBM MQ can recover multiple page sets in this way. The page set is dynamically expanded, if required, during media recovery.

During restart, IBM MQ determines the log RBA to start from by taking the lowest value from the following:

- Recovery LSN from the checkpoint log record for each page set.
- Recovery LSN from page zero in each page set.
- The RBA of the oldest incomplete unit of recovery in the system at the time the backup was taken.

All object definitions are stored on page set zero. Messages can be stored on any available page set.

Note: The queue manager cannot restart if page set zero is not available.

How to delete page sets

You delete a page set by using the DELETE PSID command; see [DELETE PSID](#) for details of this command.

You cannot delete a page set that is still referenced by any storage class. Use DISPLAY STGCLASS to find out which storage classes reference a page set.

The data set is deallocated from IBM MQ but is not deleted. It remains available for future use, or can be deleted using z/OS facilities.

Remove the page set from the started task procedure for your queue manager.

Remove the definition of the page set from your CSQINP1 initialization data set.

How to back up and restore queues using CSQUTIL

Use this topic as a reference for further information about back up and restore using CSQUTIL.

You can use the CSQUTIL utility functions for backing up and restoring queues. To back up a queue, use the COPY or SCOPY function to copy the messages from a queue onto a data set. To restore the queue, use the complementary function LOAD or SLOAD. For more information, see [IBM MQ utility program](#).

Managing buffer pools

Use this topic if you want to change or delete your buffer pools.

This topic describes how to alter and delete buffer pools. It contains these sections:

- [“How to change the number of buffers in a buffer pool” on page 527](#)
- [“How to delete a buffer pool” on page 527](#)

Buffer pools are defined during queue manager initialization, using [DEFINE BUFFPOOL](#) commands issued from the initialization input data set CSQINP1. Their attributes can be altered in response to business requirements while the queue manager is running, using the processes detailed in this topic. The queue manager records the current buffer pool attributes in checkpoint log records. These are automatically restored on subsequent queue manager restart, unless the buffer pool definition in CSQINP1 includes the REPLACE attribute.

Use the [DISPLAY USAGE](#) command to display the current buffer attributes.

You can also define buffer pools dynamically using the [DEFINE PSID](#) command with the DSN option.

If you change buffer pools dynamically, you should also update their definitions in the initialization data set CSQINP1.

See [Planejando em z/OS](#) for a description of page sets, storage classes, buffers, and buffer pools, and some of the performance considerations that apply.

Note: Buffer pools use significant storage. When you increase the size of a buffer pool or define a new buffer pool ensure that sufficient storage is available. For more information, see [Address space storage](#).

How to change the number of buffers in a buffer pool

If a buffer pool is too small, the condition can result in message [CSQP020E](#) on the console, you can allocate more buffers to it using the ALTER BUFFPOOL command as follows:

1. Determine how much space is available for new buffers by looking at the [CSQY220I](#) messages in the log. The available space is reported in MB. As a buffer has a size of 4 KB, each MB of available space allows you to allocate 256 buffers. Do not allocate all the free space to buffers, as some is required for other tasks.

If the buffer pool uses fixed 4 KB pages, that is, its PAGECLAS attribute is FIXED4KB, ensure that there is sufficient real storage available on the LPAR.

2. If the reported free space is inadequate, release some buffers from another buffer pool using the command

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

where *buf-pool-id* is the buffer pool from which you want to reclaim space and *integer* is the new number of buffers to be allocated to this buffer pool, which must be smaller than the original number of buffers allocated to it.

3. Add buffers to the buffer pool you want to expand using the command

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

where *buf-pool-id* is the buffer pool to be expanded and *integer* is the new number of buffers to be allocated to this buffer pool, which must be larger than the original number of buffers allocated to it.

How to delete a buffer pool

When a buffer pool is no longer used by any page sets, delete it to release the virtual storage allocated to it.

You delete a buffer pool using the [DELETE BUFFPOOL](#) command. The command fails if any page sets are using this buffer pool.

See [“How to delete page sets” on page 526](#) for information about how to delete page sets.

Managing queue sharing groups and shared queues on z/OS

IBM MQ can use different types of shared resources, for example queue sharing groups, shared queues, and the coupling facility. Use this topic to review the procedures needed to manage these shared resources.

This section contains information about the following topics:

- [“Managing queue sharing groups” on page 528](#)
- [“Managing shared queues” on page 531](#)
- [“Managing group objects” on page 535](#)
- [“Managing the coupling facility” on page 536](#)

Managing queue sharing groups

You can add or remove a queue manager to a queue sharing group (QSG), and manage the associated Db2 tables.

This topic has sections about the following tasks:

- [“Setting up a queue sharing group” on page 528](#)
- [“Adding a queue manager to a queue sharing group” on page 529](#)
- [“Removing a queue manager from a queue sharing group” on page 530](#)
- [“Removing a queue sharing group from the Db2 tables” on page 530](#)
- [“Validating the consistency of Db2 definitions” on page 531](#)

Setting up a queue sharing group

Each queue sharing group has a name of up to four characters. The name must be unique in your network, and must be different from any queue manager names.

Follow these steps to set up a queue sharing group:

1. If this is the first queue sharing group to use the Db2 data-sharing group, [set up the Db2 environment](#).
2. [Set up the coupling facility](#).
3. Add the queue sharing group to the Db2 tables. Use the ADD QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45AQS).
4. Add a queue manager to the queue sharing group by following the steps in [“Adding a queue manager to a queue sharing group” on page 529](#)
5. Define application structures to IBM MQ by following the steps in [“Adding a coupling facility structure” on page 536](#).
6. If required, [migrate non-shared queues to shared queues](#).
7. For availability, create shared channels into and out of the queue sharing group.
 - For connections into the queue sharing group:
 - Set up a VIPA socket or hardware router to distribute workload between the available queue managers in the QSG.
 - Define a receiver channel with QSGDISP(GROUP), to ensure the channel definition is available on all queue managers in the QSG.
 - Start a listener with INDISP(GROUP), on each queue manager, for MCA channel connections into the QSG. Client connections into the QSG should still connect to a listener started with INDISP(QMGR).
 - Change applications to connect using the QSG name, rather than a specific queue manager name.

- Ensure that the channel authentication rules on all queue managers in the QSG are the same, to allow applications to connect to any queue manager in the QSG.
- For connections out of the queue sharing group:
 - Define a shared transmission queue.
 - Define the outbound channel with QSGDISP(GROUP) and DEFCDISP(SHARED).

If you convert an existing channel to a shared channel, you might need to issue the `RESET CHANNEL` command before starting the channel as the synchronization queue used by the channel will have changed.

Adding a queue manager to a queue sharing group

A queue manager can be added to an existing queue sharing group.

Note that:

- The queue sharing group must exist before you can add queue managers to it.
- A queue manager can be a member of only one queue sharing group.

Follow these steps to add a queue manager to a queue sharing group:

1. Perform the tasks in [implement ESM security controls for the queue sharing group](#) to grant the appropriate access to the queue manager and channel initiator user IDs.
2. If the queue sharing group has CF structures configured to offload data to SMDS, perform the tasks in [set up the SMDS environment](#).
3. Stop the queue manager.
4. Use the ADD QMGR function of the queue sharing group utility (CSQ5PQSG). This program is described in the [queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45AQM).
5. [Change your system parameter module](#) to add queue sharing group data:
 - a. Modify CSQ6SYSP to specify the QSGDATA parameter. See [using CSQ6SYSP](#) for more information.
 - b. Assemble and link the system parameter module. You might want to use a different name for the load module.
 - c. Change your startup process to use the new module.
6. Copy and tailor sample member thlqual.SCSQPROC(CSQ4INSS), which defines required CF structures and SYSTEM queues. Add the customized member to the CSQINP2 DD in the queue manager startup JCL.
7. Restart your queue manager using the queue sharing group system parameter module.
8. Optionally, migrate to security profiles prefixed by the queue sharing group name, instead of the queue manager name.
9. If shared channels are used for connections into the QSG, create channel authentication rules that mirror those on the other queue managers in the QSG, to allow applications to connect to any queue manager in the QSG.
10. 10. Optionally, do either of the following to allow applications connected to the queue manager in the QSG to put messages to queues hosted by other queue managers in the QSG:
 - Turn on [intra-group queuing](#) by issuing the command `ALTER QMGR IGQ(ENABLED)`.
 - Define transmission queues and channels to the other queue managers in the QSG. Defining transmission queues with the same name as the target queue managers avoids the need to define remote queues and queue manager aliases.

Note: To add a queue manager to an existing queue sharing group containing queue managers running earlier versions of IBM MQ, you must first apply the coexistence PTF for the highest version of IBM MQ in the group to every earlier version queue manager in the group.

Removing a queue manager from a queue sharing group

You can only remove a queue manager from a queue sharing group if the queue manager's logs are not needed by another process, and all SMDS owned by the queue manager are empty.

See [Deleting shared message data sets](#) and [DELETE CFSTRUCT](#) for more information.

The logs are needed if they contain:

- The latest backup of one of the coupling facility (CF) application structures used by the queue sharing group
- Data needed by a future restore process, that is, the queue manager has used a recoverable structure since the time described by the last backup exclusion interval value.

If either or both of these points apply, or an SMDS owned by the queue manager contains messages, the queue manager cannot be removed. To determine which queue managers' logs are needed for a future restore process, use the MQSC DISPLAY CFSTATUS command with the TYPE(BACKUP) option (for details of this command, see [DISPLAY CFSTATUS](#)).

Use the following steps to remove a queue manager from a queue sharing group:

1. Stop any applications connected to the queue manager that put messages to shared queues.
2. Resolve any indoubt units of work involving this queue manager.
3. Determine if there are any messages in any SMDS owned by the queue manager by issuing the command DISPLAY USAGE TYPE(SMDS).
4. If there are offloaded messages for any application structure, wait until those messages have been retrieved from the queue. The number of offloaded messages reported by DISPLAY USAGE TYPE(SMDS) should be zero before proceeding.
5. Shut the queue manager down cleanly using STOP QMGR MODE(QUIESCE).
6. Wait for an interval at least equivalent to the value of the EXCLINT parameter you will specify in the BACKUP CFSTRUCT command in the next step.
7. On another queue manager, run a CF structure backup for each recoverable CF structure by using the MQSC BACKUP CFSTRUCT command and specifying an EXCLINT value as required in the previous step.
8. Confirm that the queue manager's logs are not needed to restore any CF structures, by inspecting the output from the command DISPLAY CFSTATUS(*) TYPE(BACKUP).
9. Use the REMOVE QMGR function of the CSQ5PQSG utility to remove the queue manager from the queue sharing group. This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45RQM).
10. Before restarting the queue manager, reset the QSGDATA system parameter to its default value, and recreate the system parameter module. See [Using CSQ6SYSP](#) for information about how to tailor your system parameters.

Note, that when removing the last queue manager in a queue sharing group, you must use the FORCE option, rather than REMOVE. This removes the queue manager from the queue sharing group, while not performing the consistency checks of the queue manager logs being required for recovery. You should only perform this operation if you are deleting the queue sharing group.

Removing a queue sharing group from the Db2 tables

To remove a queue sharing group from the Db2 tables, use the REMOVE QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45RQS).

You can only remove a queue sharing group from the common Db2 data-sharing group tables after you have removed all the queue managers from the queue sharing group (as described in [“Removing a queue manager from a queue sharing group”](#) on page 530).

When the queue sharing group record is deleted from the queue sharing group administration table, all objects and administrative information relating to that queue sharing group are deleted from other IBM MQ Db2 tables. This includes shared queue and group object information.

Validating the consistency of Db2 definitions

Problems for shared queues within a queue sharing group can occur if the Db2 object definitions have, for any reason, become inconsistent.

To validate the consistency of the Db2 object definitions for queue managers, CF structures, and shared queues, use the VERIFY QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#).

Managing shared queues

Use this topic to understand how to recover, move, and migrate shared queues.

This section describes the following tasks:

- [“Recovering shared queues” on page 531](#)
- [“Moving shared queues” on page 532](#)
- [“Migrating non-shared queues to shared queues” on page 534](#)
- [Suspending a Db2 connection](#)

Recovering shared queues

IBM MQ can recover persistent messages on shared queues if all:

- Backups of the CF structures containing the messages have been performed.
- All the logs for all queue managers in the queue sharing group are available, to perform recovery from the point the backups are taken.
- Db2 is available and the structure backup table is more recent than the most recent CF structure backup.

The messages on a shared queue are stored in a coupling facility (CF) structure. Persistent messages can be put onto shared queues, and like persistent messages on non-shared queues, they are copied to the queue manager log. The MQSC [BACKUP CFSTRUCT](#) and [RECOVER CFSTRUCT](#) commands are provided to allow the recovery of a CF structure in the unlikely event of a coupling facility failure. In such circumstances, any nonpersistent messages stored in the affected structure are lost, but persistent messages can be recovered. Any further application activity using the structure is prevented until the structure has been recovered.

To enable recovery, you must back up your coupling facility list structures frequently using the MQSC [BACKUP CFSTRUCT](#) command. The messages in the CF structure are written onto the active log data set of the queue manager making the backup. It writes a record of the backup to Db2: the name of the CF structure being backed up, the name of the queue manager doing the backup, the RBA range for this backup on that queue manager log, and the backup time. Back up CF list structures even if you are not actively using shared queues, for example, if you have set up a queue sharing group intending to use it in the future.

You can recover a CF structure by issuing an MQSC [RECOVER CFSTRUCT](#) command to the queue manager that can perform the recovery; you can use any queue manager in the queue sharing group. You can specify a single CF structure to be recovered, or you can recover several CF structures simultaneously.

As noted previously, it is important that you back up your CF list structures frequently, otherwise recovering a CF structure can take a long time. Moreover, the recovery process cannot be canceled.

The definition of a shared queue is kept in a Db2 database and can therefore be recovered if necessary using standard Db2 database procedures. See [Shared queues and queue sharing groups](#) for more information.

Moving shared queues

This section describes how to perform load balancing by moving a shared queue from one coupling facility structure to another. It also describes how to move a non-shared queue to a shared queue, and how to move a shared queue to a non-shared queue.

When you move a queue, you need to define a temporary queue as part of the procedure. This is because every queue must have a unique name, so you cannot have two queues of the same name, even if the queues have different queue dispositions. IBM MQ tolerates having two queues with the same name (as in step “2” on page 532), but you cannot use the queues.

- Moving a queue from one coupling facility structure to another
- Moving a non-shared queue to a shared queue
- Moving a shared queue to a non-shared queue

Moving a queue from one coupling facility structure to another

To move queues and their messages from one CF structure to another, use the MQSC [MOVE QLOCAL](#) command. When you have identified the queue or queues that you want to move to a new CF structure, use the following procedure to move each queue:

1. Ensure that the queue you want to move is not in use by any applications, that is, the queue attributes IPPROCS and OPPROCS are zero on all queue managers in the queue sharing group.
2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Change the queue definition to PUT(DISABLED).
3. Define a temporary queue with the same attributes as the queue that is being moved using the following command:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
```

Note: If this temporary queue exists from a previous run, delete it before doing the define.

4. Move the messages to the temporary queue using the following command:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Delete the queue you are moving, using the command:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Redefine the queue that is being moved, changing the CFSTRUCT attribute, using the following command:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
```

When the queue is redefined, it is based on the temporary queue created in step “3” on page 532.

7. Move the messages back to the new queue using the command:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

8. The queue created in step “3” on page 532 is no longer required. Use the following command to delete it:

```
DELETE QL(TEMP_QUEUE)
```

9. If the queue being moved was defined in the CSQINP2 data sets, change the CFSTRUCT attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Add the REPLACE keyword so that the existing queue definition is replaced.

Figure 30 on page 533 shows a sample job for moving a queue from one CF structure to another.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//      DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 30. Sample job for moving a queue from one CF structure to another

Moving a non-shared queue to a shared queue

The procedure for moving a non-shared queue to a shared queue is like the procedure for moving a queue from one CF structure to another (see “Moving a queue from one coupling facility structure to another” on page 532). Figure 31 on page 533 gives a sample job to do this.

Note: Remember that messages on shared queues are subject to certain restrictions on the maximum message size, message persistence, and queue index type, so you might not be able to move some non-shared queues to a shared queue.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//      DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 31. Sample job for moving a non-shared queue to a shared queue

Moving a shared queue to a non-shared queue

The procedure for moving a shared queue to a non-shared queue is like the procedure for moving a queue from one CF structure to another (see [“Moving a queue from one coupling facility structure to another”](#) on page 532).

Figure 32 on page 534 gives a sample job to do this.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
// DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) STGCLASS(NEW) QSGDISP(QMGR)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 32. Sample job for moving a shared queue to a non-shared queue

Migrating non-shared queues to shared queues

There are two stages to migrating non-shared queues to shared queues:

- Migrating the first (or only) queue manager in the queue sharing group
- Migrating any other queue managers in the queue sharing group

Migrating the first (or only) queue manager in the queue sharing group

Figure 31 on page 533 shows an example job for moving a non-shared queue to a shared queue. Do this for each queue that needs migrating.

Note:

1. Messages on shared queues are subject to certain restrictions on the maximum message size, message persistence, and queue index type, so you might not be able to move some non-shared queues to a shared queue.
2. You must use the correct index type for shared queues. If you migrate a transmission queue to be a shared queue, the index type must be MSGID.

If the queue is empty, or you do not need to keep the messages that are on it, migrating the queue is simpler. [Figure 33 on page 535](#) shows an example job to use in these circumstances.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
//          DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
DELETE QL(TEMP_QUEUE)
/*

```

Figure 33. Sample job for moving a non-shared queue without messages to a shared queue

Migrating any other queue managers in the queue sharing group

1. For each queue that does not have the same name as an existing shared queue, move the queue as described in [Figure 31 on page 533](#) or [Figure 33 on page 535](#).
2. For queues that have the same name as an existing shared queue, move the messages to the shared queue using the commands shown in [Figure 34 on page 535](#).

```

MOVE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR) TOQLOCAL(Queue_TO_MOVE)
DELETE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR)

```

Figure 34. Moving messages from a non-shared queue to an existing shared queue

Suspending a connection to Db2

If you want to apply maintenance or service to the Db2 tables or package related to shared queues without stopping your queue manager, you must temporarily disconnect queue managers in the data sharing group (DSG) from Db2.

To do this:

1. Use the MQSC command [SUSPEND QMGR FACILITY](#)(Db2).
2. Do the binds.
3. Reconnect to Db2 using the MQSC command [RESUME QMGR FACILITY](#)(Db2)

Note that there are restrictions on the use of these commands.



Attention: While the Db2 connection is suspended, the following operations will not be available. Therefore, you need to do this work during a time when your enterprise is at its least busy.

- Access to Shared queue objects for administration (define, delete,alter)
- Starting shared channels
- Storing messages in Db2
- Backup or recover CFSTRUCT

Managing group objects

Use this topic to understand how to work with group objects.

IBM MQ automatically copies the definition of a group object to page set zero of each queue manager that uses it. You can alter the copy of the definition temporarily, and IBM MQ allows you to refresh the page set copies from the repository copy. IBM MQ always tries to refresh the page set copies from the repository

copy on start-up (for channel objects, this is done when the channel initiator restarts). This ensures that the page set copies reflect the version on the repository, including any changes that were made when the queue manager was inactive.

There are circumstances under which the refresh is not performed, for example:

- If a copy of the queue is open, a refresh that would change the usage of the queue fails.
- If a copy of a queue has messages on it, a refresh that would delete that queue fails.

In these circumstances, the refresh is not performed on that copy, but is performed on the copies on all other queue managers. Check for and correct any problems with copy objects after adding, changing, or deleting a group object, and at queue manager or channel initiator restart.

Managing the coupling facility

Use this topic to understand how to add or remove coupling facility (CF) structures.

This section describes the following tasks:

- [“Adding a coupling facility structure” on page 536](#)
- [“Removing a coupling facility structure” on page 536](#)

Adding a coupling facility structure

To add a coupling facility structure, use the following procedure:

1. Define the CF structure in the CFRM policy data set. The information about setting up the coupling facility in [Set up the coupling facility](#) describes the rules for naming coupling facility structures, and how to define structures in the CFRM policy data set.
2. If you want to configure the structure to offload message data to SMDS, allocate and preformat data sets. See [creating a shared message data set](#) for details.
3. Define the structure to IBM MQ using the [DEFINE CFSTRUCT](#) command.

Removing a coupling facility structure

To remove a coupling facility structure, use the following procedure:

1. Use the following command to get a list of all the queues using the coupling facility structure that you want to delete:

```
DISPLAY QUEUE(*) QSGDISP(SHARED) CFSTRUCT(structure-name)
```

2. Delete all the queues that use the structure.
3. Delete the CF structure from IBM MQ using the [DELETE CFSTRUCT](#) command.
4. If the structure was configured to offload message data to SMDS, delete the SMDS.
5. Remove the structure definition from your CFRM policy data set and run the IXCMIAPU utility. (This is the reverse of the customization task set up the coupling facility, described in [Set up the coupling facility](#).)

Tuning coupling facility list monitoring

Use this topic to understand coupling facility list monitoring

Coupling facility (CF) list monitoring is used to monitor the state of list structures containing IBM MQ shared queues. When a message is added to a shared queue, and the queue's depth transitions from zero to non-zero, the CF notifies all queue managers in the queue sharing group. When notified the

queue managers might perform a number of actions, including notifying trigger monitors that are using TRIGGER(FIRST), or applications which are performing a get-wait.

By default, the CF notifies all queue managers in the queue sharing group at the same time. In certain configurations this can cause problems, such as:

- Skewed workload distribution, where a large percentage of messages go to a particular queue manager in the queue sharing group, often the queue manager running on the fastest LPAR, or which is closest to the CF, or
- A large number of failed gets, resulting in wasted CPU time.

z/OS V2R3 introduces a new coupling facility resource manager (CFRM) attribute called **KEYRNOTIFYDELAY**, which can be used with list structures containing shared queues (that is, application structures, and not the admin structure), and which can, for certain workloads, minimize the effects of workload skewing and empty MQGET calls, or empty MQGET calls.

KEYRNOTIFYDELAY can only be set on structures in a CF, running at CFLEVEL 22 or higher.

Its value must be one to seven decimal digits, in a range from 0 to 1,000,000 microseconds. If set to a non-zero value and the depth of a queue transitions from zero to non-zero, the CF selects a single queue manager from the queue sharing group, and notifies that queue manager before all the other queue managers in the group.

The queue manager is selected in a round-robin manner. If the selected queue manager does not process the message inside the time interval described by **KEYRNOTIFYDELAY** all the other queue managers in the queue sharing group will also be notified.

More information on **KEYRNOTIFYDELAY** is available here: [Understanding Keyrange Monitoring Notification Delay](#).

Note that there are two similar CFRM attributes called **LISTNOTIFYDELAY** and **SUBNOTIFYDELAY**. Neither of these has any measurable effect on IBM MQ workload.

Recovery and restart on z/OS

Use this topic to understand the recovery and restart mechanisms used by IBM MQ.

Restarting IBM MQ

After a queue manager terminates there are different restart procedures needed depending on how the queue manager terminated. Use this topic to understand the different restart procedures that you can use.

This topic contains information about how to restart your queue manager in the following circumstances:

- [“Restarting after a normal shutdown” on page 537](#)
- [“Restarting after an abnormal termination” on page 538](#)
- [“Restarting if you have lost your page sets” on page 538](#)
- [“Restarting if you have lost your log data sets” on page 538](#)
- [Restarting if you have lost your CF structures](#)

Restarting after a normal shutdown

If the queue manager was stopped with the STOP QMGR command, the system finishes its work in an orderly way and takes a termination checkpoint before stopping. When you restart the queue manager, it uses information from the system checkpoint and recovery log to determine the system status at shutdown.

To restart the queue manager, issue the START QMGR command as described in [“Using MQSC to start and stop a queue manager on z/OS” on page 464](#).

Restarting after an abnormal termination

IBM MQ automatically detects whether restart follows a normal shutdown or an abnormal termination.

Starting the queue manager after it has terminated abnormally is different from starting it after the STOP QMGR command has been issued. If the queue manager terminates abnormally, it terminates without being able to finish its work or take a termination checkpoint.

To restart the queue manager, issue the START QMGR command as described in [“Using MQSC to start and stop a queue manager on z/OS” on page 464](#). When you restart a queue manager after an abnormal termination, it refreshes its knowledge of its status at termination using information in the log, and notifies you of the status of various tasks.

Normally, the restart process resolves all inconsistent states. But, in some cases, you must take specific steps to resolve inconsistencies. This is described in [“Recovering units of work manually” on page 550](#).

Restarting if you have lost your page sets

If you have lost your page sets, you need to restore them from your backup copies before you can restart the queue manager. This is described in [“How to back up and recover page sets” on page 523](#).

The queue manager might take a long time to restart under these circumstances because of the length of time needed for media recovery.

Restarting if you have lost your log data sets

If, after stopping a queue manager (using the STOP QMGR command), both copies of the log are lost or damaged, you can restart the queue manager providing you have a consistent set of page sets (produced using [Method 1: Full backup](#)).

Follow this procedure:

1. Define new page sets to correspond to each existing page set in your queue manager. See [Task 15: Define your page sets](#) for information about page set definition.
Ensure that each new page set is larger than the corresponding source page set.
2. Use the FORMAT function of CSQUTIL to format the destination page set. See [Formatting page sets](#) for more details.
3. Use the RESETPAGE function of CSQUTIL to copy the existing page sets or reset them in place, and reset the log RBA in each page. See [Copying a page set and resetting the log](#) for more information about this function.
4. Redefine your queue manager log data sets and BSDS using CSQJU003 (see [The change log inventory utility](#)).
5. Restart the queue manager using the new page sets. To do this, you do one of the following:
 - Change the queue manager started task procedure to reference the new page sets. See [Task 6: Create procedures for the IBM MQ queue manager](#) for more information.
 - Use Access Method Services to delete the old page sets and then rename the new page sets, giving them the same names as the old page sets.

Attention: Before you delete any IBM MQ page set, ensure that you have made the required backup copies.

If the queue manager is a member of a queue sharing group, GROUP and SHARED object definitions are not normally affected by lost or damaged logs. However, if any shared-queue messages are involved in a unit of work that was covered by the lost or damaged logs, the effect on such uncommitted messages is unpredictable.

Note: If logs are damaged and the queue manager is a member of a queue sharing group, the ability to recover shared persistent messages might be lost. Issue a BACKUP CFSTRUCT command immediately on another active queue manager in the queue sharing group for all CF structures with the RECOVER(YES) attribute.

Restarting if you have lost your CF structures

You do not need to restart if you lose your CF structures, because the queue manager does not terminate.

Alternative site recovery on z/OS

You can recover a single queue manager or a queue sharing group, or consider disk mirroring.

See the following sections for more details:

- [Recovering a single queue manager at an alternative site](#)
- [Recovering a queue sharing group.](#)
 - [CF structure media recovery](#)
 - [Backing up the queue sharing group at the prime site](#)
 - [Recovering a queue sharing group at the alternative site](#)
- [Using disk mirroring](#)

Recovering a single queue manager at an alternative site

If a total loss of an IBM MQ computing center occurs, you can recover on another queue manager or queue sharing group at a recovery site. (See “[Recovering a queue sharing group at the alternative site](#)” on page 542 for the alternative site recovery procedure for a queue sharing group.)

To recover on another queue manager at a recovery site, you must regularly back up the page sets and the logs. As with all data recovery operations, the objectives of disaster recovery are to lose as little data, workload processing (updates), and time, as possible.

At the recovery site:

- The recovery queue managers **must** have the same names as the lost queue managers.
- The system parameter module (for example, CSQZPARM) used on each recovery queue manager must contain the same parameters as the corresponding lost queue manager.

When you have done this, reestablish all your queue managers as described in the following procedure. This can be used to perform disaster recovery at the recovery site for a single queue manager. It assumes that all that is available are:

- Copies of the archive logs and BSDSs created by normal running at the primary site (the active logs will have been lost along with the queue manager at the primary site).
- Copies of the page sets from the queue manager at the primary site that are the same age or older than the most recent archive log copies available.

You can use dual logging for the active and archive logs, in which case you need to apply the BSDS updates to both copies:

1. Define new page set data sets and load them with the data in the copies of the page sets from the primary site.
2. Define new active log data sets.
3. Define a new BSDS data set and use Access Method Services REPRO to copy the most recent archived BSDS into it.
4. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have would have just been truncated as an active log, and does not appear as an archived log. Record the STARTRBA and ENDRBA of this log.

5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the STARTRBA and ENDRBA recorded in Step “4” on page 539.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. Use the NEWLOG option of CSQJU003 to add active logs to the BSDS, do not specify STARTRBA or ENDRBA.
8. Use CSQJU003 to add a restart control record to the BSDS. Specify CRESTART CREATE, ENDRBA=highrba, where highrba is the high RBA of the most recent archive log available (found in Step “4” on page 539), plus 1.

The BSDS now describes all active logs as being empty, all the archived logs you have available, and no checkpoints beyond the end of your logs.

9. Restart the queue manager with the START QMGR command. During initialization, an operator reply message such as the following is issued:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.
REPLY Y TO CONTINUE, N TO CANCEL
```

Type Y to start the queue manager. The queue manager starts, and recovers data up to ENDRBA specified in the CRESTART statement.

See [IBM MQ utilities on z/OS reference](#) for information about using CSQJU003 and CSQJU004.

The following example shows sample input statements for CSQJU003 for steps 6, 7, and 8:

```
* Step 6
DELETE DSNAME=MQM2.LOGCOPY1.DS01
DELETE DSNAME=MQM2.LOGCOPY1.DS02
DELETE DSNAME=MQM2.LOGCOPY1.DS03
DELETE DSNAME=MQM2.LOGCOPY1.DS04
DELETE DSNAME=MQM2.LOGCOPY2.DS01
DELETE DSNAME=MQM2.LOGCOPY2.DS02
DELETE DSNAME=MQM2.LOGCOPY2.DS03
DELETE DSNAME=MQM2.LOGCOPY2.DS04

* Step 7
NEWLOG DSNAME=MQM2.LOGCOPY1.DS01,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS02,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS03,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS04,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY2.DS01,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS02,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS03,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS04,COPY2

* Step 8
CRESTART CREATE,ENDRBA=063000
```

The things you need to consider for restarting the channel initiator at the recovery site are like those faced when using ARM to restart the channel initiator on a different z/OS image. See [“Using ARM in an IBM MQ network” on page 548](#) for more information. Your recovery strategy should also cover recovery of the IBM MQ product libraries and the application programming environments that use IBM MQ (CICS, for example).

Other functions of the change log inventory utility (CSQJU003) can also be used in disaster recovery scenarios. The HIGHRBA function allows the update of the highest RBA written and highest RBA offloaded values within the bootstrap data set. The CHECKPT function allows the addition of new checkpoint queue records or the deletion of existing checkpoint queue records in the BSDS.

Attention: These functions might affect the integrity of your IBM MQ data. Only use them in disaster recovery scenarios under the guidance of IBM service personnel.

Fast copy techniques

If copies of all the page sets and logs are made while the queue manager is frozen, the copies will be a consistent set that can be used to restart the queue manager at an alternative site. They typically enable a much faster restart of the queue manager, as there is little media recovery to be performed.

Use the SUSPEND QMGR LOG command to freeze the queue manager. This command flushes buffer pools to the page sets, takes a checkpoint, and stops any further log write activity. Once log write activity has been suspended, the queue manager is effectively frozen until you issue a RESUME QMGR LOG command. While the queue manager is frozen, the page sets and logs can be copied.

By using copying tools such as FLASHCOPY or SNAPSHOT to rapidly copy the page sets and logs, the time during which the queue manager is frozen can be reduced to a minimum.

Within a queue sharing group, however, the SUSPEND QMGR LOG command might not be such a good solution. To be effective, the copies of the logs must all contain the same point in time for recovery, which means that the SUSPEND QMGR LOG command must be issued on all queue managers within the queue sharing group simultaneously, and therefore the entire queue sharing group will be frozen for some time.

Recovering a queue sharing group

In the event of a prime site disaster, you can restart a queue sharing group at a remote site using backup data sets from the prime site. To recover a queue sharing group you need to coordinate the recovery across all the queue managers in the queue sharing group, and coordinate with other resources, primarily Db2. This section describes these tasks in detail.

- [CF structure media recovery](#)
- [Backing up the queue sharing group at the prime site](#)
- [Recovering a queue sharing group at the alternative site](#)

CF structure media recovery

Media recovery of a CF structure used to hold persistent messages on a shared queue, relies on having a backup of the media that can be forward recovered by the application of logged updates. Take backups of your CF structures periodically using the MQSC BACKUP CFSTRUCT command. All updates to shared queues (MQGETs and MQPUTs) are written on the log of the queue manager where the update is performed. To perform media recovery of a CF structure you must apply logged updates to that backup from the logs of all the queue managers that have used that CF structure. When you use the MQSC RECOVER CFSTRUCT command, IBM MQ automatically merges the logs from relevant queue managers, and applies the updates to the most recent backup.

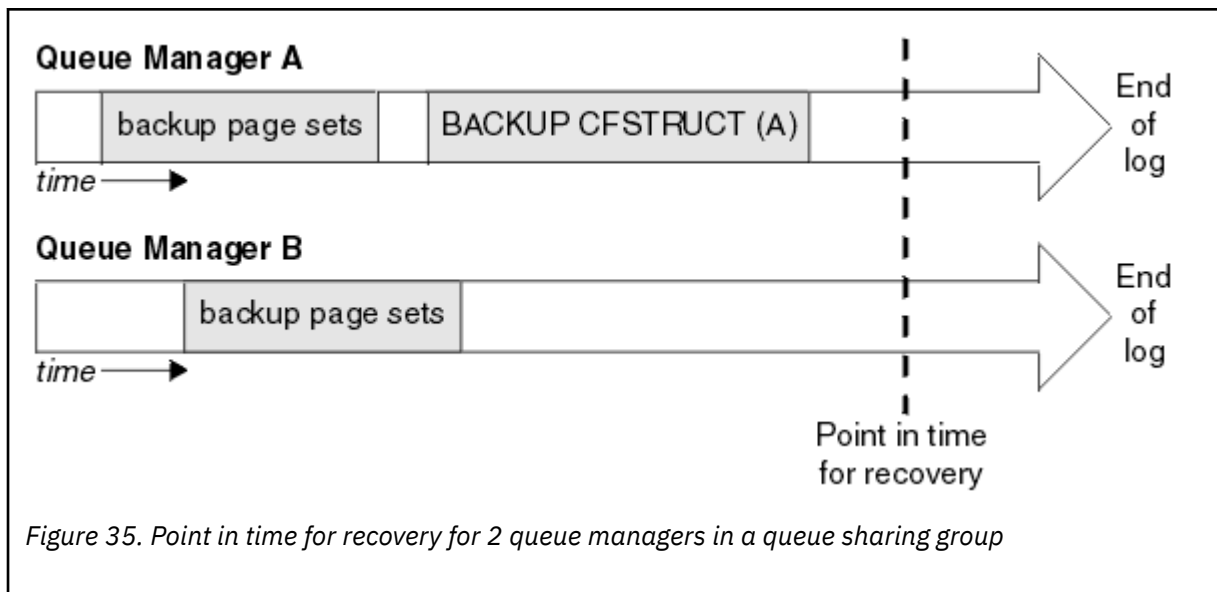
The CF structure backup is written to the log of the queue manager that processed the BACKUP CFSTRUCT command, so there are no additional data sets to be collected and transported to the alternative site.

Backing up the queue sharing group at the prime site

At the prime site you need to establish a consistent set of backups on a regular basis, which can be used in the event of a disaster to rebuild the queue sharing group at an alternative site. For a single queue manager, recovery can be to an arbitrary point in time, typically the end of the logs available at the remote site. However, where persistent messages have been stored on a shared queue, the logs of all the queue managers in the queue sharing group must be merged to recover shared queues, as any queue manager in the queue sharing group might have performed updates (MQPUTs or MQGETs) on the queue.

For recovery of a queue sharing group, you need to establish a point in time that is within the log range of the log data of all queue managers. However, as you can only **forward** recover media from the log, this point in time must be after the BACKUP CFSTRUCT command has been issued and after any page set backups have been performed. (Typically, the point in time for recovery might correspond to the end of a business day or week.)

The following diagram shows time lines for two queue managers in a queue sharing group. For each queue manager, fuzzy backups of page sets are taken (see [Method 2: Fuzzy backup](#)). On queue manager A, a BACKUP CFSTRUCT command is issued. Subsequently, an ARCHIVE LOG command is issued on each queue manager to truncate the active log, and copy it to media offline from the queue manager, which can be transported to the alternative site. End of log identifies the time at which the ARCHIVE LOG command was issued, and therefore marks the extent of log data typically available at the alternative site. The point in time for recovery must lie between the end of any page set or CF structure backups, and the earliest end of log available at the alternative site.



IBM MQ records information associated with the CF structure backups in a table in Db2. Depending on your requirements, you might want to coordinate the point in time for recovery of IBM MQ with that for Db2, or it might be sufficient to take a copy of the IBM MQ CSQ.ADMIN_B_STRBACKUP table after the BACKUP CFSTRUCT commands have finished.

To prepare for a recovery:

1. Create page set backups for each queue manager in the queue sharing group.
2. Issue a BACKUP CFSTRUCT command for each CF structure with the RECOVER(YES) attribute. You can issue these commands from a single queue manager, or from different queue managers within the queue sharing group to balance the workload.
3. Once all the backups have completed, issue an ARCHIVE LOG command to switch the active log and create copies of the logs and BSDS of each queue manager in the queue sharing group.
4. Transport the page set backups, the archived logs, the archived BSDS of all the queue managers in the queue sharing group, and your chosen Db2 backup information, off-site.

Recovering a queue sharing group at the alternative site

Before you can recover the queue sharing group, you need to prepare the environment:

1. If you have old information in your coupling facility from practice startups when you installed the queue sharing group, you need to clean this out first:

Note: If you do not have old information in the coupling facility, you can omit this step.

- a. Enter the following z/OS command to display the CF structures for this queue sharing group:

```
D XCF,STRUCTURE,STRNAME= qsgname
```

- b. For all structures that start with the queue sharing group name, use the z/OS command SETXCF FORCE CONNECTION to force the connection off those structures:

```
SETXCF FORCE,CONNECTION,STRNAME= strname,CONNAME=ALL
```

- c. Delete all the CF structures using the following command for each structure:

```
SETXCF FORCE,STRUCTURE,STRNAME= strname
```

2. Restore Db2 systems and data-sharing groups.
3. Recover the CSQ.ADMIN_B_STRBACKUP table so that it contains information about the most recent structure backups taken at the prime site.

Note: It is important that the STRBACKUP table contains the most recent structure backup information. Older structure backup information might require data sets that you have discarded as a result of the information given by a recent DISPLAY USAGE TYPE(DATASET) command, which would mean that your recovered CF structure would not contain accurate information.

4. Run the ADD QMGR command of the CSQ5PQSG utility for every queue manager in the queue sharing group. This will restore the XCF group entry for each queue manager.

When you run the utility in this scenario, the following messages are normal:

```
CSQU566I Unable to get attributes for admin structure, CF not found  
or not allocated  
CSQU546E Unable to add QMGR queue_manager_name entry,  
already exists in DB2 table CSQ.ADMIN_B_QMGR  
CSQU148I CSQ5PQSG Utility completed, return code=4
```

To recover the queue managers in the queue sharing group:

1. Define new page set data sets and load them with the data in the copies of the page sets from the primary site.
2. Define new active log data sets.
3. Define a new BSDS data set and use Access Method Services REPRO to copy the most recent archived BSDS into it.
4. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have would have just been truncated as an active log, and does not appear as an archived log. Record the STARTRBA, STARTLRSN, ENDRBA, and ENDLRSN values of this log.
5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the values recorded in Step “4” on page 543.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. Use the NEWLOG option of CSQJU003 to add active logs to the BSDS, do not specify STARTRBA or ENDRBA.
8. Calculate the *recoverylrsn* for the queue sharing group. The *recoverylrsn* is the lowest of the ENDLRSNs across all queue managers in the queue sharing group (as recorded in Step “4” on page 543), minus 1. For example, if there are two queue managers in the queue sharing group, and the ENDLRSN for one of them is B713 3C72 22C5, and for the other is B713 3D45 2123, the *recoverylrsn* is B713 3C72 22C4.
9. Use CSQJU003 to add a restart control record to the BSDS. Specify:

```
CRESTART CREATE,ENDLRSN= recoverylrsn
```

where *recoverylrsn* is the value you recorded in Step “8” on page 543.

The BSDS now describes all active logs as being empty, all the archived logs you have available, and no checkpoints beyond the end of your logs.

You must add the CRESTART record to the BSDS for each queue manager within the queue sharing group.

10. Restart each queue manager in the queue sharing group with the START QMGR command. During initialization, an operator reply message such as the following is issued:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Reply Y to start the queue manager. The queue manager starts, and recovers data up to ENDRBA specified in the CRESTART statement.

The first IBM MQ queue manager started can rebuild the admin structure partitions for other members of the queue sharing group as well as its own, and it is no longer necessary to restart each queue manager in the queue sharing group at this stage.

11. When the admin structure data for all queue managers has been rebuilt, issue a RECOVER CFSTRUCT command for each CF application structure.

If you issue the RECOVER CFSTRUCT command for all structures on a single queue manager, the log merge process is only performed once, so is quicker than issuing the command on a different queue manager for each CF structure, where each queue manager has to perform the log merge step.

When conditional restart processing is used in a queue sharing group, IBM MQ queue managers, performing peer admin rebuild, check that peers BSDS contain the same CRESTART LRSN as their own. This is to ensure the integrity of the rebuilt admin structure. It is therefore important to restart other peers in the QSG, so they can process their own CRESTART information, before the next unconditional restart of any member of the group.

Using disk mirroring

Many installations now use disk mirroring technologies such as IBM Metro Mirror (formerly PPRC) to make synchronous copies of data sets at an alternative site. In such situations, many of the steps detailed become unnecessary as the IBM MQ page sets and logs at the alternative site are effectively identical to those at the prime site. Where such technologies are used, the steps to restart a queue sharing group at an alternative site may be summarized as:

- Clear IBM MQ CF structures at the alternative site. (These often contain residual information from any previous disaster recovery exercise).
- Restore Db2 systems and all tables in the database used by the IBM MQ queue sharing group.
- Restart queue managers. Before IBM WebSphere MQ 7.0.1, it is necessary to restart each queue manager defined in the queue sharing group as each queue manager recovers its own partition of the admin structure during queue manager restart. After each queue manager has been restarted, those not on their home LPAR can be shut down again. The first IBM MQ queue manager started rebuilds the admin structure partitions for other members of the queue sharing group as well as its own, and it is no longer necessary to restart each queue manager in the queue sharing group.
- After the admin structure has been rebuilt, recover the application structures.

IBM MQ for z/OS supports use of zHyperWrite when writing to active logs mirrored using Metro Mirror. zHyperWrite can help reduce the performance impact of using Metro Mirror; see [Using Metro Mirror with IBM MQ](#) for more information.

Reinitializing a queue manager

If the queue manager has terminated abnormally you might not be able to restart it. This could be because your page sets or logs have been lost, truncated, or corrupted. If this has happened, you might have to reinitialize the queue manager (perform a cold start).

Attention

Only perform a cold start if you cannot restart the queue manager any other way. Performing a cold start enables you to recover your queue manager and your object definitions; you will **not** be able to recover your message data. Check that none of the other restart scenarios described in this topic work for you before you do this.

When you have restarted, all your IBM MQ objects are defined and available for use, but there is no message data.

Note: Do not reinitialize a queue manager while it is part of a cluster. You must first remove the queue manager from the cluster (using RESET CLUSTER commands on the other queue managers in the cluster), then reinitialize it, and finally reintroduce it to the cluster as a new queue manager.

This is because during reinitialization, the queue manager identifier (QMID) is changed, so any cluster object with the old queue manager identifier must be removed from the cluster.

For further information see the following sections:

- [Reinitializing a queue manager that is not in a queue sharing group](#)
- [Reinitializing queue managers in a queue sharing group](#)

Reinitializing a queue manager that is not in a queue sharing group

To reinitialize a queue manager, follow this procedure:

1. Prepare the object definition statements that to be used when you restart the queue manager. To do this, either:
 - If page set zero is available, use the CSQUTIL SDEFS function (see [Producing a list of IBM MQ define commands](#)). You must get definitions for all object types (authentication information objects, CF structures, channels, namelists, processes, queues, and storage classes).
 - If page set zero is not available, use the definitions from the last time you backed up your object definitions.
2. Redefine your queue manager data sets (do not do this until you have completed step “1” on page [545](#)).
See [creating the bootstrap and log data sets](#) and [defining your page sets](#) for more information.
3. Restart the queue manager using the newly defined and initialized log data sets, BSDS, and page sets. Use the object definition input statements that you created in step “1” on page [545](#) as input in the CSQINP2 initialization input data set.

Reinitializing queue managers in a queue sharing group

In a queue sharing group, reinitializing a queue manager is more complex. It might be necessary to reinitialize one or more queue managers because of page set or log problems, but there might also be problems with Db2 or the coupling facility to deal with. Because of this, there are a number of alternatives:

Cold start

Reinitializing the entire queue sharing group involves forcing all the coupling facilities structures, clearing all object definitions for the queue sharing group from Db2, deleting or redefining the logs and BSDS, and formatting page sets for all the queue managers in the queue sharing group.

Shared definitions retained

Delete or redefine the logs and BSDS, format page sets for all queue managers in the queue sharing group, and force all the coupling facilities structures. On restart, all messages will have been deleted. The queue managers re-create COPY objects that correspond to GROUP objects that still exist in the Db2 database. Any shared queues still exist and can be used.

Single queue manager reinitialized

Delete or redefine the logs and BSDS, and format page sets for the single queue manager (this deletes all its private objects and messages). On restart, the queue manager re-creates COPY objects that correspond to GROUP objects that still exist in the Db2 database. Any shared queues still exist, as do the messages on them, and can be used.

Point in time recovery of a queue sharing group

This is the alternative site disaster recovery scenario.

Shared objects are recovered to the point in time achieved by Db2 recovery (described in [A Db2 system fails](#)). Each queue manager can be recovered to a point in time achievable from the backup copies available at the alternative site.

Persistent messages can be used in queue sharing groups, and can be recovered using the MQSC RECOVER CFSTRUCT command. Note that this command recovers to the time of failure. However, there is no recovery of nonpersistent shared queue messages; they are lost unless you have made backup copies independently using the COPY function of the CSQUTIL utility program.

It is not necessary to try to restore each queue manager to the same point in time because there are no interdependencies between the local objects on different queue managers (which are what is actually being recovered), and the queue manager resynchronization with Db2 on restart creates or deletes COPY objects as necessary on a queue manager by queue manager basis.

Using the z/OS Automatic Restart Manager (ARM)

Use this topic to understand how you can use ARM to automatically restart your queue managers.

This section contains information about the following topics:

- [“What is the ARM?” on page 546](#)
- [“ARM policies” on page 547](#)
- [“Using ARM in an IBM MQ network” on page 548](#)

What is the ARM?

The z/OS Automatic Restart Manager (ARM) is a z/OS recovery function that can improve the availability of your queue managers. When a job or task fails, or the system on which it is running fails, ARM can restart the job or task without operator intervention.

If a queue manager or a channel initiator has failed, ARM restarts it on the same z/OS image. If z/OS, and hence a whole group of related subsystems and applications have failed, ARM can restart all the failed systems automatically, in a predefined order, on another z/OS image within the sysplex. This is called a *cross-system restart*.

Restart the channel initiator by ARM only in exceptional circumstances. If the queue manager is restarted by ARM, restart the channel initiator from the CSQINP2 initialization data set (see [“Using ARM in an IBM MQ network” on page 548](#)).

You can use ARM to restart a queue manager on a different z/OS image within the sysplex in the event of z/OS failure. The network implications of IBM MQ ARM restart on a different z/OS image are described in [“Using ARM in an IBM MQ network” on page 548](#).

To enable automatic restart:

- Set up an ARM couple data set.
- Define the automatic restart actions that you want z/OS to perform in an *ARM policy*.
- Start the ARM policy.

Also, IBM MQ must register with ARM at startup (this happens automatically).

Note: If you want to restart queue managers in different z/OS images automatically, you must define every queue manager as a subsystem in each z/OS image on which that queue manager might be restarted, with a sysplex wide unique four character subsystem name.

ARM couple data sets

Ensure that you define the couple data sets required for ARM, and that they are online and active before you start any queue manager for which you want ARM support. IBM MQ automatic ARM registration fails if the couple data sets are not available at queue manager startup. In this situation, IBM MQ assumes that the absence of the couple data set means that you do not want ARM support, and initialization continues.

See *z/OS MVS Setting up a Sysplex* for information about ARM couple data sets.

ARM policies

The Automatic Restart Manager policies are user-defined rules that control ARM functions that can control any restarts of a queue manager.

ARM functions are controlled by a user-defined *ARM policy*. Each z/OS image running a queue manager instance that is to be restarted by ARM must be connected to an ARM couple data set with an active ARM policy.

IBM provides a default ARM policy. You can define new policies, or override the policy defaults by using the *administrative data utility* (IXCMIAPU) provided with z/OS. *z/OS MVS Setting up a Sysplex* describes this utility, and includes full details of how to define an ARM policy.

Figure 36 on page 547 shows an example of an ARM policy. This sample policy restarts any queue manager within a sysplex, if either the queue manager failed, or a whole system failed.

```
//IXCMIAPU EXEC PGM=IXCMIAPU,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(ARM)
DEFINE POLICY NAME(ARMPOL1) REPLACE(YES)
RESTART_GROUP(DEFAULT)
ELEMENT(*)
RESTART_ATTEMPTS(0) /* Jobs not to be restarted by ARM */
RESTART_GROUP(GROUP1)
ELEMENT(SYSMQGRMQ*) /* These jobs to be restarted by ARM */
/*
```

Figure 36. Sample ARM policy

For more information see:

- [Defining an ARM policy](#)
- [Activating an ARM policy](#)
- [Registering with ARM](#)

Defining an ARM policy

Set up your ARM policy as follows:

- Define RESTART_GROUPS for each queue manager instance that also contain any CICS or IMS subsystems that connect to that queue manager instance. If you use a subsystem naming convention, you might be able to use the '?' and '*' wild-card characters in your element names to define RESTART_GROUPS with minimum definition effort.
- Specify TERMTYPE(ELEMTERM) for your channel initiators to indicate that they will be restarted only if the channel initiator has failed and the z/OS image has not failed.

- Specify `TERMTYPE(ALLTERM)` for your queue managers to indicate that they will be restarted if either the queue manager has failed or the z/OS image has failed.
- Specify `RESTART_METHOD(BOTH, PERSIST)` for both queue managers and channel initiators. This tells ARM to restart using the JCL it saved (after resolution of system symbols) during the last startup. It tells ARM to do this irrespective of whether the individual element failed, or the z/OS image failed.
- Accept the default values for all the other ARM policy options.

Activating an ARM policy

To start your automatic restart management policy, issue the following z/OS command:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME= mypol
```

When the policy is started, all systems connected to the ARM couple data set use the same active policy. Use the `SETXCF STOP` command to disable automatic restarts.

Registering with ARM

IBM MQ registers automatically as an *ARM element* during queue manager startup (subject to ARM availability). It deregisters during its shutdown phase, unless requested not to.

At startup, the queue manager determines whether ARM is available. If it is, IBM MQ registers using the name `SYSMQMGR ssid`, where *ssid* is the four character queue manager name, and `SYSMQMGR` is the element type.

The `STOP QMGR MODE(QUIESCE)` and `STOP QMGR MODE(FORCE)` commands deregister the queue manager from ARM (if it was registered with ARM at startup). This prevents ARM restarting this queue manager. The `STOP QMGR MODE(RESTART)` command does not deregister the queue manager from ARM, so it is eligible for immediate automatic restart.

Each channel initiator address space determines whether ARM is available, and if so registers with the element name `SYSMQCH ssid`, where *ssid* is the queue manager name, and `SYSMQCH` is the element type.

The channel initiator is always deregistered from ARM when it stops normally, and remains registered only if it ends abnormally. The channel initiator is always deregistered if the queue manager fails.

Using ARM in an IBM MQ network

You can set up your queue manager so that the channel initiators and associated listeners are started automatically when the queue manager is restarted.

To ensure fully automatic queue manager restart on the same z/OS image for both LU 6.2 and TCP/IP communication protocols:

- Start your listeners automatically by adding the appropriate `START LISTENER` command to the `CSQINPX` data set.
- Start your channel initiator automatically by adding the appropriate `START CHINIT` command to the `CSQINP2` data set.

For restarting a queue manager with TCP/IP or LU6.2, see

- [“Restarting on a different z/OS image with TCP/IP” on page 549](#)
- [“Restarting on a different z/OS image with LU 6.2” on page 550](#)

See [Task 13: Customize the initialization input data sets](#) for information about the `CSQINP2` and `CSQINPX` data sets.

Restarting on a different z/OS image with TCP/IP

If you are using TCP/IP as your communication protocol, and you are using virtual IP addresses, you can configure these to recover on other z/OS images, allowing channels connecting to that queue manager to reconnect without any changes. Otherwise, you can reallocate a TCP/IP address after moving a queue manager to a different z/OS image only if you are using clusters or if you are connecting to a queue sharing group using a WLM dynamic Domain Name System (DNS) logical group name.

- [When using clustering](#)
- [When connecting to a queue sharing group](#)

When using clustering

z/OS ARM responds to a system failure by restarting the queue manager on a different z/OS image in the same sysplex; this system has a different TCP/IP address to the original z/OS image. The following explains how you can use IBM MQ clusters to reassign a queue manager's TCP/IP address after it has been moved by ARM restart to a different z/OS image.

When a client queue manager detects the queue manager failure (as a channel failure), it responds by reallocating suitable messages on its cluster transmission queue to a different server queue manager that hosts a different instance of the target cluster queue. However, it cannot reallocate messages that are bound to the original server by affinity constraints, or messages that are in doubt because the server queue manager failed during end-of-batch processing. To process these messages, do the following:

1. Allocate a different cluster-receiver channel name and a different TCP/IP port to each z/OS queue manager. Each queue manager needs a different port so that two systems can share a single TCP/IP stack on a z/OS image. One of these is the queue manager originally running on that z/OS image, and the other is the queue manager that ARM will restart on that z/OS image following a system failure. Configure each port on each z/OS image, so that ARM can restart any queue manager on any z/OS image.
2. Create a different channel initiator command input file (CSQINPX) for each queue manager and z/OS image combination, to be referenced during channel initiator startup.

Each CSQINPX file must include a START LISTENER PORT(port) command specific to that queue manager, and an ALTER CHANNEL command for a cluster-receiver channel specific to that queue manager and z/OS image combination. The ALTER CHANNEL command needs to set the connection name to the TCP/IP name of the z/OS image on which it is restarted. It must include the port number specific to the restarted queue manager as part of the connection name.

The start-up JCL of each queue manager can have a fixed data set name for this CSQINPX file, and each z/OS image must have a different version of each CSQINPX file on a non-shared DASD volume.

If an ARM restart occurs, IBM MQ advertises the changed channel definition to the cluster repository, which in turn publishes it to all the client queue managers that have expressed an interest in the server queue manager.

The client queue manager treats the server queue manager failure as a channel failure, and tries to restart the failed channel. When the client queue manager learns the new server connection-name, the channel restart reconnects the client queue manager to the restarted server queue manager. The client queue manager can then resynchronize its messages, resolve any in-doubt messages on the client queue manager's transmission queue, and normal processing can continue.

When connecting to a queue sharing group

When connecting to a queue sharing group through a TCP/IP dynamic Domain Name System (DNS) logical group name, the connection name in your channel definition specifies the logical group name of your queue sharing group, not the host name or IP address of a physical machine. When this channel starts, it connects to the dynamic DNS and is then connected to one of the queue managers in the queue sharing group. This process is explained in [Setting up communication for IBM MQ for z/OS using queue sharing groups](#).

In the unlikely event of an image failure, one of the following occurs:

- The queue managers on the failing image de-register from the dynamic DNS running on your sysplex. The channel responds to the connection failure by entering RETRYING state and then connects to the dynamic DNS running on the sysplex. The dynamic DNS allocates the inbound request to one of the remaining members of the queue sharing group that is still running on the remaining images.
- If no other queue manager in the queue sharing group is active and ARM restarts the queue manager and channel initiator on a different image, the group listener registers with dynamic DNS from this new image. This means that the logical group name (from the connection name field of the channel) connects to the dynamic DNS and is then connected to the same queue manager, now running on a different image. No change was required to the channel definition.

For this type of recovery to occur, the following points must be noted:

- On z/OS, the dynamic DNS runs on one of the z/OS images in the sysplex. If this image were to fail, the dynamic DNS needs to be configured so that there is a secondary name server active in the sysplex, acting as an alternative to the primary name server. Information about primary and secondary dynamic DNS servers can be found in the [OS/390® SecureWay CS IP Configuration](#) manual
- The TCP/IP group listener might have been started on a particular IP address that might not be available on this z/OS image. If so, the listener might need to be started on a different IP address on the new image. If you are using virtual IP addresses, you can configure these to recover on other z/OS images so that no change to the START LISTENER command is required.

Restarting on a different z/OS image with LU 6.2

If you use only LU 6.2 communication protocols, carry out the following procedure to enable network reconnect after automatic restart of a queue manager on a different z/OS image within the sysplex:

- Define each queue manager within the sysplex with a unique subsystem name.
- Define each channel initiator within the sysplex with a unique LUNAME. This is specified in both the queue manager attributes and in the START LISTENER command.

Note: The LUNAME names an entry in the APPC side table, which in turn maps this to the actual LUNAME.

- Set up a shared APPC side table, which is referenced by each z/OS image within the sysplex. This should contain an entry for each channel initiator's LUNAME. See [z/OS MVS Planning: APPC/MVS Management](#) for information about this.
- Set up an APPCPM xx member of SYS1.PARMLIB for each channel initiator within the sysplex to contain an LUADD to activate the APPC side table entry for that channel initiator. These members should be shared by each z/OS image. The appropriate SYS1.PARMLIB member is activated by a z/OS command SET APPC= xx, which is issued automatically during ARM restart of the queue manager (and its channel initiator) on a different z/OS image, as described in the following text.
- Use the LU62ARM queue manager attribute to specify the xx suffix of this SYS1.PARMLIB member for each channel initiator. This causes the channel initiator to issue the required z/OS command SET APPC= xx to activate its LUNAME.

Define your ARM policy so that it restarts the channel initiator only if it fails while its z/OS image stays up; the user ID associated with the XCFAS address space must be authorized to issue the IBM MQ command START CHINIT. Do not restart the channel initiator automatically if its z/OS image also fails, instead use commands in the CSQINP2 and CSQINPX data sets to start the channel initiator and listeners.

Recovering units of work manually

You can manually recover units of work CICS, IMS, RRS, or other queue managers in a queue sharing group. You can use queue manager commands to display the status of the units of work associated with each connection to the queue manager.

This topic contains information about the following subjects:

- [“Displaying connections and threads” on page 551](#)
- [“Recovering CICS units of recovery manually” on page 551](#)
- [“Recovering IMS units of recovery manually” on page 555](#)
- [“Recovering RRS units of recovery manually” on page 556](#)
- [“Recovering units of recovery on another queue manager in the queue sharing group” on page 557](#)

Displaying connections and threads

You can use the `DISPLAY CONN` command to get information about connections to queue managers and their associated units of work. You can display active units of work to see what is currently happening, or to see what needs to be terminated to allow the queue manager to shut down, and you can display unresolved units of work to help with recovery.

Active units of work

To display only active units of work, use

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)
```

Unresolved units of work

An unresolved unit of work, also known as an "in-doubt thread", is one that is in the second pass of the two-phase commit operation. Resources are held in IBM MQ on its behalf. To display unresolved units of work, use

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

External intervention is needed to resolve the status of unresolved units of work. This might only involve starting the recovery coordinator (CICS, IMS, or RRS) or might involve more, as described in the following sections.

Recovering CICS units of recovery manually

Use this topic to understand what happens when the CICS adapter restarts, and then explains how to deal with any unresolved units of recovery that arise.

What happens when the CICS adapter restarts

Whenever a connection is broken, the adapter has to go through a *restart phase* during the *reconnect process*. The restart phase resynchronizes resources. Resynchronization between CICS and IBM MQ enables in-doubt units of work to be identified and resolved.

Resynchronization can be caused by:

- An explicit request from the distributed queuing component
- An implicit request when a connection is made to IBM MQ

If the resynchronization is caused by connecting to IBM MQ, the sequence of events is:

1. The connection process retrieves a list of in-doubt units of work (UOW) IDs from IBM MQ.
2. The UOW IDs are displayed on the console in CSQC313I messages.
3. The UOW IDs are passed to CICS.
4. CICS initiates a resynchronization task (CRSY) for each in-doubt UOW ID.

5. The result of the task for each in-doubt UOW is displayed on the console.

You need to check the messages that are displayed during the connect process:

CSQC313I

Shows that a UOW is in doubt.

CSQC400I

Identifies the UOW and is followed by one of these messages:

- CSQC402I or CSQC403I shows that the UOW was resolved successfully (committed or backed out).
- CSQC404E, CSQC405E, CSQC406E, or CSQC407E shows that the UOW was not resolved.

CSQC409I

Shows that all UOWs were resolved successfully.

CSQC408I

Shows that not all UOWs were resolved successfully.

CSQC314I

Warns that UOW IDs highlighted with a * are not resolved automatically. These UOWs must be resolved explicitly by the distributed queuing component when it is restarted.

Figure 37 on page 552 shows an example set of restart messages displayed on the z/OS console.

```
CSQ9022I +CSQ1 CSQYASCP ' START QMGR' NORMAL COMPLETION
+CSQC323I VICIC1 CSQCQCON CONNECT received from TERMID=PB62 TRANID=CKCN
+CSQC303I VICIC1 CSQCQCON CSQCSERV loaded. Entry point is 850E8918
+CSQC313I VICIC1 CSQCQCON UOWID=VICIC1.A6E5A6F0E2178D25 is in doubt
+CSQC313I VICIC1 CSQCQCON UOWID=VICIC1.A6E5A6F055B2AC25 is in doubt
+CSQC313I VICIC1 CSQCQCON UOWID=VICIC1.A6E5A6EFFD60D425 is in doubt
+CSQC313I VICIC1 CSQCQCON UOWID=VICIC1.A6E5A6F07AB56D22 is in doubt
+CSQC307I VICIC1 CSQCQCON Successful connection to subsystem VC2
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAD18) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAA10) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BA708) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAE88) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAB80) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA878) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA570) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA268) connect
successful
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F0E2178D25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F055B2AC25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F07AB56D22
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6EFFD60D425
+CSQC409I VICIC1 CSQCTRUE Resynchronization completed successfully
```

Figure 37. Example restart messages

The total number of CSQC313I messages should equal the total number of CSQC402I plus CSQC403I messages. If the totals are not equal, there are UOWs that the connection process cannot resolve. Those UOWs that cannot be resolved are caused by problems with CICS (for example, a cold start) or with IBM MQ, or by distributing queuing. When these problems have been fixed, you can initiate another resynchronization by disconnecting and then reconnecting.

Alternatively, you can resolve each outstanding UOW yourself using the RESOLVE INDOUBT command and the UOW ID shown in message CSQC400I. You must then initiate a disconnect and a connect to clean

up the *unit of recovery descriptors* in CICS. You need to know the correct outcome of the UOW to resolve UOWs manually.

All messages that are associated with unresolved UOWs are locked by IBM MQ and no Batch, TSO, or CICS task can access them.

If CICS fails and an emergency restart is necessary, *do not* vary the GENERIC APPLID of the CICS system. If you do and then reconnect to IBM MQ, data integrity with IBM MQ cannot be guaranteed. This is because IBM MQ treats the new instance of CICS as a different CICS (because the APPLID is different). In-doubt resolution is then based on the wrong CICS log.

How to resolve CICS units of recovery manually

If the adapter ends abnormally, CICS and IBM MQ build in-doubt lists either dynamically or during restart, depending on which subsystem caused the abend.

Note: If you use the DFH\$INDB sample program to show units of work, you might find that it does not always show IBM MQ UOWs correctly.

When CICS connects to IBM MQ, there might be one or more units of recovery that have not been resolved.

One of the following messages is sent to the console:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E
- CSQC408I

For details of what these messages mean, see the [CICS adapter and Bridge messages](#) messages.

CICS retains details of units of recovery that were not resolved during connection startup. An entry is purged when it no longer appears on the list presented by IBM MQ.

Any units of recovery that CICS cannot resolve must be resolved manually using IBM MQ commands. This manual procedure is rarely used within an installation, because it is required only where operational errors or software problems have prevented automatic resolution. *Any inconsistencies found during in-doubt resolution must be investigated.*

To resolve the units of recovery:

1. Obtain a list of the units of recovery from IBM MQ using the following command:

```
+CSQ1 DISPLAY CONN( * ) WHERE(UOWSTATE EQ UNRESOLVED)
```

You receive the following message:

```

CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN (BC85772CBE3E0001)
EXTCONN (C3E2D8C3C7D9F0F940404040404040)
TYPE (CONN)
CONNOPTS (
MQCNO_STANDARD_BINDING
)
UOWLOGDA (2005-02-04)
UOWLOGTI (10.17.44)
UOWSTDA (2005-02-04)
UOWSTTI (10.17.44)
UOWSTATE (UNRESOLVED)
NID (IYRCSQ1 .BC8571519B60222D)
EXTURID (BC8571519B60222D)
QMURID (0000002BDA50)
URTYPE (CICS)
USERID (MQTEST)
APPLTAG (IYRCSQ1)
ASID (0000)
APPLTYPE (CICS)
TRANSID (GP02)
TASKNO (0000096)
END CONN DETAILS

```

For CICS connections, the NID consists of the CICS applid and a unique number provided by CICS at the time the syncpoint log entries are written. This unique number is stored in records written to both the CICS system log and the IBM MQ log at syncpoint processing time. This value is referred to in CICS as the *recovery token*.

2. Scan the CICS log for entries related to a particular unit of recovery.

Look for a PREPARE record for the task-related installation where the recovery token field (JCSRMTKN) equals the value obtained from the network ID. The network ID is supplied by IBM MQ in the DISPLAY CONN command output.

The PREPARE record in the CICS log for the units of recovery provides the CICS task number. All other entries on the log for this CICS task can be located using this number.

You can use the CICS journal print utility DFHJUP when scanning the log. For details of using this program, see the *CICS Operations and Utilities Guide*.

3. Scan the IBM MQ log for records with the NID related to a particular unit of recovery. Then use the URID from this record to obtain the rest of the log records for this unit of recovery.

When scanning the IBM MQ log, note that the IBM MQ startup message CSQJ001I provides the start RBA for this session.

The print log records program (CSQ1LOGP) can be used for that purpose.

4. If you need to, do in-doubt resolution in IBM MQ.

IBM MQ can be directed to take the recovery action for a unit of recovery using an IBM MQ [RESOLVE INDOUBT](#) command.

To recover all threads associated with a specific *connection-name*, use the NID(*) option.

The command produces one of the following messages showing whether the thread is committed or backed out:

```

CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED

```

When performing in-doubt resolution, CICS and the adapter are not aware of the commands to IBM MQ to commit or back out units of recovery, because only IBM MQ resources are affected. However, CICS keeps details about the in-doubt threads that could not be resolved by IBM MQ. This information is purged

either when the list presented is empty, or when the list does not include a unit of recovery of which CICS has details.

Recovering IMS units of recovery manually

Use this topic to understand what happens when the IMS adapter restarts, and then explains how to deal with any unresolved units of recovery that arise.

What happens when the IMS adapter restarts

Whenever the connection to IBM MQ is restarted, either following a queue manager restart or an IMS / START SUBSYS command, IMS initiates the following resynchronization process:

1. IMS presents the list of unit of work (UOW) IDs that it believes are in doubt to the IBM MQ IMS adapter one at a time with a resolution parameter of Commit or Backout.
2. The IMS adapter passes the resolution request to IBM MQ and reports the result back to IMS.
3. Having processed all the IMS resolution requests, the IMS adapter gets from IBM MQ a list of all UOWs that IBM MQ still holds in doubt that were initiated by the IMS system. These are reported to the IMS master terminal in message CSQQ008I.

Note: While a UOW is in doubt, any associated IBM MQ message is locked by IBM MQ and is not available to any application.

How to resolve IMS units of recovery manually

When IMS connects to IBM MQ, IBM MQ might have one, or more in-doubt units of recovery that have not been resolved.

If IBM MQ has in-doubt units of recovery that IMS did not resolve, the following message is issued at the IMS master terminal:

```
CSQQ008I nn units of recovery are still in doubt in queue manager qmgr-name
```

If this message is issued, IMS was either cold-started or it was started with an incomplete log tape. This message can also be issued if IBM MQ or IMS terminates abnormally because of a software error or other subsystem failure.

After receiving the CSQQ008I message:

- The connection remains active.
- IMS applications can still access IBM MQ resources.
- Some IBM MQ resources remain locked out.

If the in-doubt thread is not resolved, IMS message queues can start to build up. If the IMS queues fill to capacity, IMS terminates. You must be aware of this potential difficulty, and you must monitor IMS until the in-doubt units of recovery are fully resolved.

Recovery procedure

Use the following procedure to recover the IMS units of work:

1. Force the IMS log closed, using /SWI OLDS, and then archive the IMS log. Use the utility, DFSERA10, to print the records from the previous IMS log tape. Type X ' 3730 ' log records indicate a phase-2 commit request and type X ' 38 ' log records indicate an abort request. Record the requested action for the last transaction in each dependent region.
2. Run the DL/I batch job to back out each PSB involved that has not reached a commit point. The process might take some time because transactions are still being processed. It might also lock

up a number of records, which could affect the rest of the processing and the rest of the message queues.

3. Produce a list of the in-doubt units of recovery from IBM MQ using the following command:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

You receive the following message:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3E2C5C3F240404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2005-02-15)
UOWLOGTI(16.39.43)
UOWSTDA(2005-02-15)
UOWSTTI(16.39.43)
UOWSTATE(UNRESOLVED)
NID(IM8F .BC45A794D3810344)
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0000)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

For IMS, the NID consists of the IMS connection name and a unique number provided by IMS. The value is referred to in IMS as the *recovery token*. For more information, see the [IMS documentation](#).

4. Compare the NIDs (IMSID plus OASN in hexadecimal) displayed in the DISPLAY THREAD messages with the OASNs (4 bytes decimal) shown in the DFSERA10 output. Decide whether to commit or back out.
5. Perform in-doubt resolution in IBM MQ with the [RESOLVE INDOUBT](#) command, as follows:

```
RESOLVE INDOUBT( connection-name )
ACTION(COMMIT|BACKOUT)
NID( network-id )
```

To recover all threads associated with *connection-name*, use the NID(*) option. The command results in one of the following messages to indicate whether the thread is committed or backed out:

```
CSQV414I  THREAD network-id COMMIT SCHEDULED
CSQV415I  THREAD network-id BACKOUT SCHEDULED
```

When performing in-doubt resolution, IMS and the adapter are not aware of the commands to IBM MQ to commit or back out in-doubt units of recovery because only IBM MQ resources are affected.

Recovering RRS units of recovery manually

Use this topic to understand the how to determine if there are in-doubt RRS units of recovery, and how to manually resolve those units of recovery.

When RRS connects to IBM MQ, IBM MQ might have one, or more in-doubt units of recovery that have not been resolved. If IBM MQ has in-doubt units of recovery that RRS did not resolve, one of the following messages is issued at the z/OS console:

- CSQ3011I
- CSQ3013I

- CSQ3014I
- CSQ3016I

Both IBM MQ and RRS provide tools to display information about in-doubt units of recovery, and techniques for manually resolving them.

In IBM MQ, use the DISPLAY CONN command to display information about in-doubt IBM MQ threads. The output from the command includes RRS unit of recovery IDs for those IBM MQ threads that have RRS as a coordinator. This can be used to determine the outcome of the unit of recovery.

Use the RESOLVE INDOUBT command to resolve the IBM MQ in-doubt thread manually. This command can be used to either commit or back out the unit of recovery after you have determined what the correct decision is.

Recovering units of recovery on another queue manager in the queue sharing group

Use this topic to identify, and manually recover units of recovery on other queue managers in a queue sharing group.

If a queue manager that is a member of a queue sharing group fails and cannot be restarted, other queue managers in the group can perform peer recovery, and take over from it. However, the queue manager might have in-doubt units of recovery that cannot be resolved by peer recovery because the final disposition of that unit of recovery is known only to the failed queue manager. These units of recovery are resolved when the queue manager is eventually restarted, but until then, they remain in doubt.

This means that certain resources (for example, messages) might be locked, making them unavailable to other queue managers in the group. In this situation, you can use the DISPLAY THREAD command to display these units of work on the inactive queue manager. If you want to resolve these units of recovery manually to make the messages available to other queue managers in the group, you can use the RESOLVE INDOUBT command.

When you issue the DISPLAY THREAD command to display units of recovery that are in doubt, you can use the QMNAME keyword to specify the name of the inactive queue manager. For example, if you issue the following command:

```
+CSQ1 DISPLAY THREAD(*) TYPE(INDOUBT) QMNAME(QM01)
```

You receive the following messages:

```
CSQV436I +CSQ1 INDOUBT THREADS FOR QM01 -
NAME   THREAD-XREF   URID NID
USER1  0000000000000000000000000000 CSQ:0001.0
USER2  0000000000000000000000000000 CSQ:0002.0
DISPLAY THREAD REPORT COMPLETE
```

If the queue manager specified is active, IBM MQ does not return information about in-doubt threads, but issues the following message:

```
CSQV435I CANNOT USE QMNAME KEYWORD, QM01 IS ACTIVE
```

Use the IBM MQ command RESOLVE INDOUBT to resolve the in-doubt threads manually. Use the QMNAME keyword to specify the name of the inactive queue manager in the command.

This command can be used to commit or back out the unit of recovery. The command resolves the shared portion of the unit of recovery only; any local messages are unaffected and remain locked until the queue manager restarts, or reconnects to CICS, IMS, or RRS batch.

z/OS IBM MQ and IMS

IBM MQ provides two components to interface with IMS, the IBM MQ - IMS adapter, and the IBM MQ - IMS bridge. These components are commonly called the IMS adapter, and the IMS bridge.

z/OS Operating the IMS adapter

Use this topic to understand how to operate the IMS adapter, which connects IBM MQ to IMS systems.

Note: The IMS adapter does not incorporate any operations and control panels.

This topic contains the following sections:

- [“Controlling IMS connections” on page 558](#)
- [“Connecting from the IMS control region” on page 558](#)
- [“Displaying in-doubt units of recovery” on page 560](#)
- [“Controlling IMS dependent region connections” on page 562](#)
- [“Disconnecting from IMS” on page 564](#)
- [“Controlling the IMS trigger monitor” on page 565](#)

z/OS Controlling IMS connections

Use this topic to understand the IMS operator commands which control and monitor the connection to IBM MQ.

IMS provides the following operator commands to control and monitor the connection to IBM MQ:

/CHANGE SUBSYS

Deletes an in-doubt unit of recovery from IMS.

/DISPLAY OASN SUBSYS

Displays outstanding recovery elements.

/DISPLAY SUBSYS

Displays connection status and thread activity.

/START SUBSYS

Connects the IMS control region to a queue manager.

/STOP SUBSYS

Disconnects IMS from a queue manager.

/TRACE

Controls the IMS trace.

For more information about these commands, see the *IMS/ESA® Operator's Reference* manual for the level of IMS that you are using.

IMS command responses are sent to the terminal from which the command was issued. Authorization to issue IMS commands is based on IMS security.

z/OS Connecting from the IMS control region

Use this topic to understand the mechanisms available to connect from IMS to IBM MQ.

IMS makes one connection from its control region to each queue manager that uses IMS. IMS must be enabled to make the connection in one of these ways:

- Automatically during either:
 - A cold start initialization.
 - A warm start of IMS, if the IBM MQ connection was active when IMS was shut down.
- In response to the IMS command:

```
/START SUBSYS sysid
```

where *sysid* is the queue manager name.

The command can be issued regardless of whether the queue manager is active.

The connection is not made until the first IBM MQ API call to the queue manager is made. Until that time, the IMS command /DIS SUBSYS shows the status as 'NOT CONN'.

The order in which you start IMS and the queue manager is not significant.

IMS cannot re-enable the connection to the queue manager automatically if the queue manager is stopped with a STOP QMGR command, the IMS command /STOP SUBSYS, or an abnormal end. Therefore, you must make the connection by using the IMS command /START SUBSYS.

If an IMS command is seen in the queue manager console log similar to this:

```
MODIFY IMS*,SS*
```

check the IMS master log and ensure that IBM MQ has RACF authority to issue IMS Adapter MODIFY commands.

Initializing the adapter and connecting to the queue manager

The adapter is a set of modules loaded into the IMS control and dependent regions, using the IMS external Subsystem Attach Facility.

This procedure initializes the adapter and connects to the queue manager:

1. Read the subsystem member (SSM) from IMS.PROCLIB. The SSM chosen is an IMS EXEC parameter. There is one entry in the member for each queue manager to which IMS can connect. Each entry contains control information about an IBM MQ adapter.

2. Load the IMS adapter.

Note: IMS loads one copy of the adapter modules for each IBM MQ instance that is defined in the SSM member.

3. Attach the external subsystem task for IBM MQ.
4. Run the adapter with the CTL EXEC parameter (IMSID) as the connection name.

The process is the same whether the connection is part of initialization or a result of the IMS command /START SUBSYS.

If the queue manager is active when IMS tries to make the connection, the following messages are sent:

- to the z/OS console:

```
DFS3613I ESS TCB INITIALIZATION COMPLETE
```

- to the IMS master terminal:

```
CSQQ000I IMS/TM imsid connected to queue manager ssnm
```

When IMS tries to make the connection and *the queue manager is not active*, the following messages are sent to the IMS master terminal each time an application makes an MQI call:

```
CSQQ001I IMS/TM imsid not connected to queue manager ssnm.  
Notify message accepted  
DFS3607I MQM1 SUBSYSTEM ID EXIT FAILURE, FC = 0286, RC = 08,  
JOBNAME = IMSEMPR1
```

If you get DFS3607I messages when you start the connection to IMS or on system startup, this indicates that the queue manager is not available. To prevent a large number of messages being generated, you must do one of the following:

1. Start the relevant queue manager.
2. Issue the IMS command:

```
/STOP SUBSYS
```

so that IMS does not expect to connect to the queue manager.

If you do neither, a DFS3607I message and the associated CSQQ001I message are issued each time a job is scheduled in the region and each time a connection request to the queue manager is made by an application.

Thread attachment

In an MPP or IFP region, IMS makes a thread connection when the first application program is scheduled into that region, even if that application program does not make an IBM MQ call. In a BMP region, the thread connection is made when the application makes its first IBM MQ call (MQCONN or MQCONNX). This thread is retained for the duration of the region or until the connection is stopped.

For both the message driven and non-message driven regions, the recovery thread cross-reference identifier, *Thread-xref*, associated with the thread is:

```
PSTid + PSBname
```

where:

PSTid

Partition specification table region identifier

PSBname

Program specification block name

You can use connection IDs as unique identifiers in IBM MQ commands, in which case IBM MQ automatically inserts these IDs into any operator message that it generates.

z/OS Displaying in-doubt units of recovery

You can display in-doubt units of recovery and attempt to recover them.

The operational steps used to list and recover in-doubt units of recovery in this topic are for relatively simple cases only. If the queue manager ends abnormally while connected to IMS, IMS might commit or back out work without IBM MQ being aware of it. When the queue manager restarts, that work is termed *in doubt*. A decision must be made about the status of the work.

To display a list of in-doubt units of recovery, issue the command:


```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

IBM MQ responds with a message like the following:

```
CSQM201I +CSQ1 CSQMDRTC DIS CONN DETAILS
CONN(BC0F6125F5A30001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-11-02)
UOWLOGTI(12.27.58)
UOWSTDA(2004-11-02)
UOWSTTI(12.27.58)
UOWSTATE(UNRESOLVED)
NID(CSQ1CHIN.BC0F5F1C86FC0766)
EXTURID(00000000000001F000000007472616E5F6964547565204E6F762020...)
QMURID(0000000026232)
URTYPE(XA)
USERID( )
APPLTAG(CSQ1CHIN)
ASID(0000)
APPLTYPE(CHINIT)
CHANNEL( )
CONNNAME( )
END CONN DETAILS
```

For an explanation of the attributes in this message, see the description of the [DISPLAY CONN](#) command.

Recovering in-doubt units of recovery

To recover in-doubt units of recovery, issue this command:

```
+CSQ1 RESOLVE INDOUBT( connection-name ) ACTION(COMMIT|BACKOUT)
NID( net-node.number )
```

where:

connection-name

The IMS system ID.

ACTION

Indicates whether to commit (COMMIT) or back out (BACKOUT) this unit of recovery.

net-node.number

The associated net-node.number.

When you have issued the RESOLVE INDOUBT command, one of the following messages is displayed:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id BACKOUT SCHEDULED
```

Resolving residual recovery entries

At given times, IMS builds a list of residual recovery entries (RREs). RREs are units of recovery about which IBM MQ might be in doubt. They arise in several situations:

- If the queue manager is not active, IMS has RREs that cannot be resolved until the queue manager is active. These RREs are not a problem.
- If the queue manager is active and connected to IMS, and if IMS backs out the work that IBM MQ has committed, the IMS adapter issues message CSQQ010E. If the data in the two systems must be consistent, there is a problem. For information about resolving this problem, see [“Recovering IMS units of recovery manually”](#) on page 555.
- If the queue manager is active and connected to IMS, there might still be RREs even though no messages have informed you of this problem. After the IBM MQ connection to IMS has been established, you can issue the following IMS command to find out if there is a problem:

```
/DISPLAY OASN SUBSYS sysid
```

To purge the RRE, issue one of the following IMS commands:

```
/CHANGE SUBSYS sysid RESET  
/CHANGE SUBSYS sysid RESET OASN nnnn
```

where *nnnn* is the originating application sequence number listed in response to your +CSQ1 DISPLAY command. This is the schedule number of the program instance, giving its place in the sequence of invocations of that program since the last IMS cold start. IMS cannot have two in-doubt units of recovery with the same schedule number.

These commands reset the status of IMS ; they do not result in any communication with IBM MQ.

Controlling IMS dependent region connections

You can control, monitor, and, when necessary, terminate connections between IMS and IBM MQ.

Controlling IMS dependent region connections involves the following activities:

- [Connecting from dependent regions](#)
- [Region error options](#)
- [Monitoring the activity on connections](#)
- [Disconnecting from dependent regions](#)

Connecting from dependent regions

The IMS adapter used in the control region is also loaded into dependent regions. A connection is made from each dependent region to IBM MQ. This connection is used to coordinate the commitment of IBM MQ and IMS work. To initialize and make the connection, IMS does the following:

1. Reads the subsystem member (SSM) from IMS.PROCLIB.

A subsystem member can be specified on the dependent region EXEC parameter. If it is not specified, the control region SSM is used. If the region is never likely to connect to IBM MQ, to avoid loading the adapter, specify a member with no entries.

2. Loads the IBM MQ adapter.

For a batch message program, the load is not done until the application issues its first messaging command. At that time, IMS tries to make the connection.

For a message-processing program region or IMS fast-path region, the attempt is made when the region is initialized.

Region error options

If the queue manager is not active, or if resources are not available when the first messaging command is sent from application programs, the action taken depends on the error option specified on the SSM entry. The options are:

R

The appropriate return code is sent to the application.

Q

The application ends abnormally with abend code U3051. The input message is re-queued.

A

The application ends abnormally with abend code U3047. The input message is discarded.

Monitoring the activity on connections

A thread is established from a dependent region when an application makes its first successful IBM MQ request. You can display information about connections and the applications currently using them by issuing the following command from IBM MQ:

```
+CSQ1 DISPLAY CONN(*) ALL
```

The command produces a message like the following:

```
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-12-15)
UOWLOGTI(16.39.43)
UOWSTDA(2004-12-15)
UOWSTTI(16.39.43)
UOWSTATE(ACTIVE)
NID( )
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0049)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

For the control region, *thread-xref* is the special value CONTROL. For dependent regions, it is the PSTid concatenated with the PSBname. *auth-id* is either the user field from the job card, or the ID from the z/OS started procedures table.

For an explanation of the displayed list, see the description of message CSQV402I in the [mensagens, conclusão e códigos de razão do IBM MQ for z/OS](#) documentation.

IMS provides a display command to monitor the connection to IBM MQ. It shows which program is active on each dependent region connection, the LTERM user name, and the control region connection status. The command is:

```
/DISPLAY SUBSYS name
```

The status of the connection between IMS and IBM MQ is shown as one of:

```
CONNECTED  
NOT CONNECTED  
CONNECT IN PROGRESS  
STOPPED  
STOP IN PROGRESS  
INVALID SUBSYSTEM NAME= name  
SUBSYSTEM name NOT DEFINED BUT RECOVERY OUTSTANDING
```

The thread status from each dependent region is one of the following:

```
CONN  
CONN, ACTIVE (includes LTERM of user)
```

Disconnecting from dependent regions

To change values in the SSM member of IMS.PROCLIB, you disconnect a dependent region. To do this, you must:

1. Issue the IMS command:

```
/STOP REGION
```

2. Update the SSM member.
3. Issue the IMS command:

```
/START REGION
```

Disconnecting from IMS

The connection is ended when either IMS or the queue manager terminates. Alternatively, the IMS master terminal operator can explicitly break the connection.

To terminate the connection between IMS and IBM MQ, use the following IMS command:

```
/STOP SUBSYS sysid
```

The command sends the following message to the terminal that issued it, typically the master terminal operator (MTO):

```
DFS058I STOP COMMAND IN PROGRESS
```

The IMS command:

```
/START SUBSYS sysid
```

is required to reestablish the connection.

Note: The IMS command /STOP SUBSYS is not completed if an IMS trigger monitor is running.

Controlling the IMS trigger monitor

You can use the CSQQTRMN transaction to stop, and start the IMS trigger monitor.

The IMS trigger monitor (the CSQQTRMN transaction) is described in the [Setting up the IMS trigger monitor](#).

To control the IMS trigger monitor see:

- [Starting CSQQTRMN](#)
- [Stopping CSQQTRMN](#)

Starting CSQQTRMN

1. Start a batch-oriented BMP that runs the program CSQQTRMN for each initiation queue you want to monitor.
2. Modify your batch JCL to add a DDname of CSQQUT1 that points to a data set containing the following information:

```
QMGRNAME=q_manager_name    Comment: queue manager name
INITQUEUEUENAME=init_q_name  Comment: initiation queue name
LTERM=lterm                  Comment: LTERM to remove error messages
CONSOLEMESSAGES=YES         Comment: Send error messages to console
```

where:

q_manager_name	The name of the queue manager (if this is blank, the default nominated in CSQQDEFV is assumed)
init_q_name	The name of the initiation queue to be monitored
lterm	The IMS LTERM name for the destination of error messages (if this is blank, the default value is MASTER).
CONSOLEMESSAGES= YES	Requests that messages sent to the nominated IMS LTERM are also sent to the z/OS console. If this parameter is omitted or misspelled, the default is NOT to send messages to the console.

3. Add a DD name of CSQQUT2 if you want a printed report of the processing of CSQQUT1 input.

Note:

1. The data set CSQQUT1 is defined with LRECL=80. Other DCB information is taken from the data set. The DCB for data set CSQQUT2 is RECFM=VBA and LRECL=125.

2. You can put only one keyword on each record. The keyword value is delimited by the first blank following the keyword; this means that you can include comments. An asterisk in column 1 means that the whole input record is a comment.
3. If you misspell either of the QMGRNAME or LTERM keywords, CSQQTRMN uses the default for that keyword.
4. Ensure that the subsystem is started in IMS (by the /START SUBSYS command) before submitting the trigger monitor BMP job. If it is not started, your trigger monitor job terminates with abend code U3042.

Stopping CSQQTRMN

Once started, CSQQTRMN runs until either the connection between IBM MQ and IMS is broken due to one of the following events:

- the queue manager ending
 - IMS ending
- or a z/OS STOP **jobname** command is entered.

Controlling the IMS bridge

Use this topic to understand the IMS commands that you can use to control the IMS bridge.

There are no IBM MQ commands to control the IBM MQ-IMS bridge. However, you can stop messages being delivered to IMS in the following ways:

- For non-shared queues, by using the ALTER QLOCAL(xxx) GET(DISABLED) command for all bridge queues.
- For clustered queues, by using the SUSPEND QMGR CLUSTER(xxx) command. This is effective only when another queue manager is also hosting the clustered bridge queue.
- For clustered queues, by using the SUSPEND QMGR FACILITY(IMSBRIDGE) command. No further messages are sent to IMS, but the responses for any outstanding transactions are received from IMS.

To start sending messages to IMS again, issue the RESUME QMGR FACILITY(IMSBRIDGE) command.

You can also use the MQSC command DISPLAY SYSTEM to display whether the bridge is suspended.

See [MQSC commands](#) for details of these commands.

For further information see:

- [“Starting and stopping the IMS bridge” on page 566](#)
- [“Controlling IMS connections” on page 567](#)
- [Controlling bridge queues](#)
- [“Resynchronizing the IMS bridge” on page 568](#)
- [Working with tpipe names](#)
- [Deleting messages from IMS](#)
- [Deleting tpipes](#)
- [“IMS Transaction Expiration” on page 570](#)

Starting and stopping the IMS bridge

Start the IBM MQ bridge by starting OTMA. Either use the IMS command:

```
/START OTMA
```

or start it automatically by specifying OTMA=YES in the IMS system parameters. If OTMA is already started, the bridge starts automatically when queue manager startup has completed. An IBM MQ event message is produced when OTMA is started.

Use the IMS command:

```
/STOP OTMA
```

to stop OTMA communication. When this command is issued, an IBM MQ event message is produced.

Controlling IMS connections

IMS provides these operator commands to control and monitor the connection to IBM MQ:

/DEQUEUE TMEMBER *tmember* TPIPE *tpipe*

Removes messages from a Tpipe. Specify PURGE to remove all messages or PURGE1 to remove the first message only.

/DISPLAY OTMA

Displays summary information about the OTMA server and clients, and client status.

/DISPLAY TMEMBER *name*

Displays information about an OTMA client.

/DISPLAY TRACE TMEMBER *name*

Displays information about what is being traced.

/SECURE OTMA

Sets security options.

/START OTMA

Enables communications through OTMA.

/START TMEMBER *tmember* TPIPE *tpipe*

Starts the named Tpipe.

/STOP OTMA

Stops communications through OTMA.

/STOP TMEMBER *tmember* TPIPE *tpipe*

Stops the named Tpipe.

/TRACE

Controls the IMS trace.

For more information about these commands, see the *IMS/ESA Operators Reference* manual for the level of IMS that you are using.

IMS command responses are sent to the terminal from which the command was issued. Authorization to issue IMS commands is based on IMS security.

Controlling bridge queues

To stop communicating with the queue manager with XCF member name *tmember* through the bridge, issue the following IMS command:

```
/STOP TMEMBER tmember TPIPE ALL
```

To resume communication, issue the following IMS command:

```
/START TMEMBER tmember TPIPE ALL
```

The Tpipes for a queue can be displayed using the MQ DISPLAY QUEUE command.

To stop communication with the queue manager on a single Tpipe, issue the following IMS command:

```
/STOP TMEMBER tmember TPIPE tpipe
```

One or two Tpipes are created for each active bridge queue, so issuing this command stops communication with the IBM MQ queue. To resume communication, use the following IMS command :

```
/START TMEMBER tmember TPIPE tpipe
```

Alternatively, you can alter the attributes of the IBM MQ queue to make it get inhibited.

Resynchronizing the IMS bridge

The IMS bridge is automatically restarted whenever the queue manager, IMS, or OTMA are restarted.

The first task undertaken by the IMS bridge is to resynchronize with IMS. This involves IBM MQ and IMS checking sequence numbers on every synchronized Tpipe. A synchronized Tpipe is used when persistent messages are sent to IMS from an IBM MQ - IMS bridge queue using commit mode zero (commit-then-send).

If the bridge cannot resynchronize with IMS, the IMS sense code is returned in message CSQ2023E and the connection to OTMA is stopped. If the bridge cannot resynchronize with an individual IMS Tpipe, the IMS sense code is returned in message CSQ2025E and the Tpipe is stopped. If a Tpipe has been cold started, the recoverable sequence numbers are automatically reset to 1.

If the bridge discovers mismatched sequence numbers when resynchronizing with a Tpipe, message CSQ2020E is issued. Use the IBM MQ command RESET TPIPE to initiate resynchronization with the IMS Tpipe. You need to provide the XCF group and member name, and the name of the Tpipe; this information is provided by the message.

You can also specify:

- A new recoverable sequence number to be set in the Tpipe for messages sent by IBM MQ, and to be set as the partner's receive sequence number. If you do not specify this, the partner's receive sequence number is set to the current IBM MQ send sequence number.
- A new recoverable sequence number to be set in the Tpipe for messages received by IBM MQ, and to be set as the partner's send sequence number. If you do not specify this, the partner's send sequence number is set to the current IBM MQ receive sequence number.

If there is an unresolved unit of recovery associated with the Tpipe, this is also notified in the message. Use the IBM MQ command RESET TPIPE to specify whether to commit the unit of recovery, or back it out. If you commit the unit of recovery, the batch of messages has already been sent to IMS, and is deleted from the bridge queue. If you back the unit of recovery out, the messages are returned to the bridge queue, to be later sent to IMS.

Commit mode 1 (send-then-commit) Tpipes are not synchronized.

Considerations for Commit mode 1 transactions

In IMS, commit mode 1 (CM1) transactions send their output replies before sync point.

A CM1 transaction might not be able to send its reply, for example because:

- The Tpipe on which the reply is to be sent is stopped
- OTMA is stopped
- The OTMA client (that is, the queue manager) has gone away
- The reply-to queue and dead-letter queue are unavailable

For these reasons, the IMS application sending the message pseudo-abends with code U0119. The IMS transaction and program are not stopped in this case.

These reasons often prevent messages being sent into IMS, as well as replies being delivered from IMS. A U0119 abend can occur if:

- The Tpipe, OTMA, or the queue manager is stopped while the message is in IMS
- IMS replies on a different Tpipe to the incoming message, and that Tpipe is stopped
- IMS replies to a different OTMA client, and that client is unavailable.

Whenever a U0119 abend occurs, both the incoming message to IMS and the reply messages to IBM MQ are lost. If the output of a CMO transaction cannot be delivered for any of these reasons, it is queued on the Tpipe within IMS.

Working with tpipe names

Many of the commands used to control the IBM MQ - IMS bridge require the *tpipe* name. Use this topic to understand how you can find further details of the tpipe name.

You need *tpipe* names for many of the commands that control the IBM MQ - IMS bridge. You can get the tpipe names from DISPLAY QUEUE command and note the following points:

- tpipe names are assigned when a local queue is defined
- a local queue is given two tpipe names, one for sync and one for non-sync
- tpipe names will not be known to IMS until after some communication between IMS and IBM MQ specific to that particular local queue takes place
- For a tpipe to be available for use by the IBM MQ - IMS bridge its associated queue must be assigned to a Storage Class that has the correct XCF group and member name fields completed

Deleting messages from IMS

A message that is destined for IBM MQ through the IMS bridge can be deleted if the Tmember/Tpipe is stopped. To delete one message for the queue manager with XCF member name *tmember*, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE1
```

To delete all the messages on the Tpipe, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE
```

Deleting tpipes

You cannot delete IMS tpipes yourself. They are deleted by IMS at the following times:

- Synchronized tpipes are deleted when IMS is cold started.

- Non-synchronized tpipes are deleted when IMS is restarted.

IMS Transaction Expiration

An expiration time is associated with a transaction; any IBM MQ message can have an expiration time associated with it. The expiration interval is passed from the application, to IBM MQ, using the MQMD.Expiry field. The time is the duration of a message before it expires, expressed as a value in tenths of a second. An attempt to perform the MQGET of a message, later than it has expired, results in the message being removed from the queue and expiry processing performed. The expiration time decreases as a message flows between queue managers on an IBM MQ network. When an IMS message is passed across the IMS bridge to OTMA, the remaining message expiry time is passed to OTMA as a transaction expiration time.

If a transaction has an expiration time specified, OTMA expires the input transactions in three different places in IMS:

- input message receiving from XCF
- input message enqueueing time
- application GU time

No expiration is performed after the GU time.

The transaction EXPRTIME can be provided by:

- IMS transaction definition
- IMS OTMA message header
- IMS DFSINSX0 user exit
- IMS CREATE or UPDATE TRAN commands

IMS indicates that it has expired a transaction by abending a transaction with 0243, and issuing a message. The message issued is either DFS555I in the non-shared-queues environment, or DFS2224I in the shared-queues environment.

z/OS

Operating Advanced Message Security on z/OS

The Advanced Message Security address space accepts commands using the z/OS MODIFY command.

Procedure

- Modify Advanced Message Security on z/OS.

To enter commands for the Advanced Message Security (AMS) address space, use the z/OS MODIFY command.

For example:

```
F qmgrAMSM, cmd
```

where *qmgr* is the prefix of the started task name.

The following table describes the MODIFY commands that are accepted:

Table 29. Advanced Message Security address space MODIFY commands		
Command	Option	Description
DISPLAY		Display version information

Table 29. Advanced Message Security address space MODIFY commands (continued)		
Command	Option	Description
REFRESH	KEYRING POLICY ALL	Refresh the key ring certificates, security policies, or both.
SMFAUDIT	SUCCESS FAILURE ALL	Set whether SMF auditing is required when AMS successfully protects or unprotects messages, when AMS fails to protect or unprotect messages, or both.
SMFTYPE	0 - 255	Set the SMF record type to be generated when AMS protects or unprotects messages. To disable SMF auditing specify a record type of 0.

Note: To specify an option it must be separated by a comma. For example:

```
F qmgrAMSM,REFRESH KEYRING
F qmgrAMSM,SMFAUDIT ALL
F qmgrAMSM,SMFTYPE 180
```

- Refresh Advanced Message Security on z/OS.

Changes that are made effective by issuing the **REFRESH** command apply to applications that issue MQOPEN after the **REFRESH** command has completed. Existing applications that have a queue open, continue to use the options from when the application opened the queue. To use the new values, the application has to close and reopen the queue.

- Start and stop AMS on z/OS.

You do not need to enter a command to start or stop the Advanced Message Security address space. The AMS address space is started automatically when the queue manager is started if AMS has been enabled with the **SPLCAP** parameter of CSQ6SYSP, and is stopped when the queue manager is stopped.

Administrando IBM MQ Internet Pass-Thru

Esta seção descreve como administrar o IBM MQ Internet Pass-Thru (MQIPT).




Configure MQIPT fazendo alterações no arquivo de configuração do `mqipt.conf` conforme descrito em [Configurando IBM MQ Internet Pass-Thru](#). Para administrar o MQIPT, incluindo a atualização do MQIPT para que as mudanças de configuração entrem em vigor sem reiniciar o MQIPT, use o comando **mqiptAdmin**. Para obter informações sobre a administração do MQIPT usando o comando **mqiptAdmin**, consulte [“Administrando MQIPT usando a linha de comandos”](#) na página 574.

Iniciando e Parando o MQIPT

É possível iniciar MQIPT a partir da linha de comandos ou fazer com que ele inicie automaticamente quando o sistema for iniciado. É possível parar o MQIPT usando o comando **mqiptAdmin**.

Iniciando MQIPT a partir da linha de comandos

MQIPT é instalado em um diretório de instalação, tal como:

-  **Windows** C:\MQIPT em sistemas Windows, com scripts executáveis no C:\MQIPT\bin
-  **Linux**  **AIX** /opt/mqipt em sistemas AIX and Linux, com scripts executáveis no /opt/mqipt/bin

MQIPT também usa um diretório inicial, que contém o arquivo de configuração `mcipt.conf` e quaisquer arquivos que são produzidos pela MQIPT quando está em execução. Os subdiretórios a seguir do diretório inicial MQIPT são criados automaticamente quando MQIPT é chamado pela primeira vez:

- Um diretório `errors` no qual qualquer First Failure Support Technology (FFST) e arquivos de rastreamento são escritos
- Um diretório `logs` no qual o log de conexão é mantido

O ID do usuário sob o qual o MQIPT é executado deve ter permissão para criar esses diretórios ou, como alternativa, os diretórios já deverão existir e o ID do usuário deverá ter permissão para criar, ler e gravar arquivos neles. Além disso, se você estiver usando uma política do Java security manager, a política de segurança deverá conceder as permissões necessárias para esses diretórios. Para obter mais informações sobre as configurações de política do Security Manager, consulte [Java security manager](#).

É possível usar o diretório de instalação como um diretório inicial. Se você usar esse diretório, deverá assegurar que o ID do usuário sob o qual MQIPT é executado tenha as permissões apropriadas e que qualquer política do Security Manager esteja configurada corretamente.

Para iniciar o MQIPT, use o comando `mcipt`, que está localizado no diretório `bin` do diretório de instalação do MQIPT. Por exemplo, o comando a seguir inicia uma instância de MQIPT que usa o diretório `C:\mciptHome` como o diretório inicial:

```
mcipt C:\mciptHome
```

Para obter mais informações sobre o comando `mcipt`, consulte [mcipt \(iniciar o MQIPT\)](#).




É possível usar o comando `mcipt` para especificar um nome a ser fornecido para a instância do MQIPT que está sendo iniciada. O nome da instância do MQIPT é usado para administrar as instâncias locais do MQIPT com o comando `mciptAdmin` sem precisar usar uma porta de comando. Se esse parâmetro não for especificado, o nome do diretório inicial do MQIPT será usado como o nome da instância do MQIPT.

As mensagens do console mostram o status de MQIPT. Se ocorrer um erro, consulte a [Resolução de problemas do IBM MQ Internet Pass-Thru](#). As mensagens a seguir são um exemplo da saída quando MQIPT é iniciado com êxito:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mcipt.conf
MQCPI152 MQIPT name is C:\mciptHome
MQCPI021 Password checking has been enabled on the command port
MQCPI144 MQ Advanced capabilities not enabled
MQCPI011 The path C:\mciptHome\logs will be used to store the log files
MQCPI006 Route 1414 is starting and will forward messages to :
MQCPI034 ...examplehost(1414)
MQCPI035 ...using MQ protocols
MQCPI057 ...trace level 5 enabled
MQCPI078 Route 1414 ready for connection requests
```

Iniciando MQIPT automaticamente

É possível instalar o MQIPT como um serviço do sistema que é iniciado automaticamente quando o sistema é iniciado. Use o comando `mciptService` para instalar e desinstalar o serviço do MQIPT.

-  Em sistemas Windows, o comando `mciptService` instala o MQIPT como um serviço do Windows.
-   Em sistemas AIX and Linux, o comando `mciptService` instala MQIPT como um serviço de inicialização do System V que inicia quando o sistema é inicializado. Em sistemas Linux que não suportam inicialização do System V, use outro método, como `systemd`, para gerenciar MQIPT como um serviço.

Quando o serviço MQIPT é iniciado, todas as rotas MQIPT ativas são iniciadas. Quando o serviço é interrompido, todas as rotas estão sujeitas a um encerramento imediato.

É possível instalar apenas um serviço do MQIPT em um sistema, mesmo se há mais de uma instalação do MQIPT no sistema.

Para obter mais informações sobre o comando **mqiPTService**, consulte [mqiPTService \(gerenciar o serviço do MQIPT\)](#).

Parando MQIPT

É possível parar o MQIPT usando o comando **mqiPTAdmin** com o parâmetro **-stop**.

Por exemplo, o comando a seguir para uma instância do MQIPT com o nome **mqiPT1** em execução localmente sob o mesmo ID de usuário que o comando **mqiPTAdmin**:

```
mqiPTAdmin -stop -n ipt1
```

O comando **mqiPTAdmin** se conecta à instância ativa do MQIPT para a administração usando um dos métodos a seguir:

- conectando-se a uma instância local do MQIPT sem usar a porta de comando.
- fazendo uma conexão de rede com uma porta de comando.

O encerramento remoto deve ser ativado configurando a propriedade **RemoteShutDown** como **true** antes que o comando **mqiPTAdmin** possa ser usado para parar o MQIPT enviando um comando para uma porta de comando.

Para obter mais informações sobre a administração do MQIPT usando o comando **mqiPTAdmin**, consulte [“Administrando MQIPT usando a linha de comandos”](#) na página 574.

Especificando a chave de criptografia de senha

Se a configuração do MQIPT contiver senhas criptografadas usando uma chave de criptografia diferente da chave padrão, deve-se fornecer a chave de criptografia de senha em um arquivo que o MQIPT possa ler quando ele for iniciado.

O arquivo de chave de criptografia de senha

As senhas criptografadas para serem armazenadas e usadas pelo MQIPT podem ser criptografadas usando uma chave de criptografia fornecida por você. Se você não fornecer uma chave de criptografia, a chave de criptografia padrão será usada. Não é necessário especificar uma chave de criptografia de senha, no entanto, é mais seguro fazer isso. Se você não especificar a sua própria chave de criptografia, a chave de criptografia padrão será usada.

Se você fornecer uma chave de criptografia de senha, ela deverá ser armazenada em um arquivo que possa ser acessado pelo comando **mqiPTPW** usado para criptografar senhas e o MQIPT. As únicas restrições sobre o conteúdo do arquivo são que ele deve conter pelo menos um caractere e apenas uma linha de texto.

Nota: Deve-se assegurar que permissões de arquivo apropriadas sejam configuradas no arquivo de chave de criptografia de senha para evitar que qualquer usuário não autorizado leia a chave de criptografia. Apenas o usuário que executa o comando **mqiPTPW** e o usuário sob o qual o MQIPT é executado precisam de autoridade para ler a chave de criptografia de senha.

A mesma chave de criptografia de senha é usada para criptografar e descriptografar todas as senhas armazenadas para uma instância do MQIPT. Portanto, é necessário apenas um único arquivo de chave de criptografia de senha para cada instalação do MQIPT.

Se a chave de criptografia de senha de uma instalação do MQIPT mudar, todas as senhas criptografadas deverão ser criptografadas novamente usando a nova chave de criptografia.

Iniciando o MQIPT

O nome padrão do arquivo de chaves de criptografia de senha é `MQIPT_HOME_DIR/mqipt_cred.key`, em que `MQIPT_HOME_DIR` é o diretório onde o arquivo de configuração `mqipt.conf` é armazenado. Se você estiver planejando executar o MQIPT como um serviço iniciado automaticamente, o arquivo de chave de criptografia de senha deverá ser criado com o nome padrão.

Se o arquivo de chave de criptografia de senha for criado com um nome diferente do nome padrão, o nome do arquivo deverá ser fornecido para o MQIPT quando ele for iniciado. O nome do arquivo de chave de criptografia de senha pode ser especificado usando qualquer um dos métodos a seguir, em ordem de preferência:

1. o parâmetro **-sf** no comando **mqipt** usado para iniciar o MQIPT.
2. a variável de ambiente `MQS_MQIPTCRED_KEYFILE`.
3. a propriedade `com.ibm.mq.ipc.cred.keyfile` Java.

Se nenhum nome de arquivo de chave de criptografia de senha for fornecido, o nome do arquivo padrão será usado, se o arquivo existir. Se o arquivo de chave de criptografia de senha padrão não existir, a chave de criptografia de senha padrão será usada.

Administrando MQIPT usando a linha de comandos

É possível usar o comando **mqiptAdmin** na linha de comandos para administrar o MQIPT.

É possível usar o comando **mqiptAdmin** para executar as funções administrativas a seguir:

- Listar instâncias locais ativas do MQIPT.
- Atualizar uma instância do MQIPT depois de fazer mudanças no arquivo de configuração.
- Parar uma instância do MQIPT.

O comando **mqiptAdmin** está localizado no subdiretório `bin` do diretório de instalação do MQIPT.

O comando **mqiptAdmin** se conecta à instância ativa do MQIPT para a administração usando um dos métodos a seguir:

- fazendo uma conexão de rede com uma porta de comando.
- conectando-se a uma instância local do MQIPT sem usar a porta de comando.

O comando **mqiptAdmin** é compatível com versões anteriores do MQIPT, mas não é possível usar o comando para administrar versões do MQIPT que sejam mais recentes do que a versão do comando **mqiptAdmin**. Em um ambiente que inclui versões diferentes do MQIPT, deve-se usar a versão mais recente do comando **mqiptAdmin**.

Para obter mais informações sobre a sintaxe do comando **mqiptAdmin**, consulte [mqiptAdmin \(administrar o MQIPT\)](#).

Administração local sem uma porta de comando

As instâncias locais do MQIPT podem ser administradas sem usar uma porta de comandos. A administração local permite administrar o MQIPT usando o comando **mqiptAdmin** apenas quando ele é executado no mesmo sistema que a instância do MQIPT que você deseja administrar.

Para que **mqiptAdmin** seja autorizado a administrar uma instância local do MQIPT sem usar a porta de comando, a instância do MQIPT deve estar em execução no mesmo sistema e sob o mesmo ID de usuário que **mqiptAdmin**. Como alternativa, em AIX and Linux, **mqiptAdmin** pode ser executado como `root`.

A administração local é ativada por padrão. Para desativar a administração local, use a propriedade de configuração **LocalAdmin**. Para obter mais informações sobre a propriedade **LocalAdmin**, consulte [LocalAdmin](#).

Para administrar instâncias locais do MQIPT, deve-se dar a cada instância um nome. É possível atribuir um nome a uma instância do MQIPT usando o parâmetro **-n** ao iniciar o MQIPT com o comando **mqipt**. Se você não especificar um nome ao iniciar o MQIPT, o nome do diretório inicial será usado como o nome

da instância do MQIPT. Por exemplo, o comando a seguir inicia o MQIPT e designa o nome `ipt1` para a instância:

```
mqiPT /opt/mqiPT1 -n ipt1
```

Uma vez que a instância tem um nome, é possível administrá-la especificando o nome no comando **mqiPTAdmin** com o parâmetro **-n**. Por exemplo, o comando a seguir interrompe a instância local do MQIPT com o nome `ipt1`:

```
mqiPTAdmin -stop -n ipt1
```

É possível listar todas as instâncias ativas locais do MQIPT que o comando **mqiPTAdmin** está autorizado a administrar sem usar uma porta de comando usando o comando **mqiPTAdmin** com o parâmetro **-list**. Por exemplo, o comando a seguir lista todas as instâncias ativas locais do MQIPT que o usuário que iniciou o comando **mqiPTAdmin** está autorizado a administrar:

```
mqiPTAdmin -list
```

Administração usando uma porta de comando

É possível configurar o MQIPT com uma porta de comando que não seja segura, e uma porta de comando que seja protegida com TLS (TLS). É possível usar essas portas de comando para administrar o MQIPT como qualquer usuário que esteja no mesmo sistema que a instância do MQIPT que você deseja administrar ou de um sistema remoto.

Versões anteriores do MQIPT apenas aceitavam comandos administrativos emitidos para a porta de comando não segura.

Nota: As conexões com a porta de comando não segura não são criptografadas, portanto, os dados enviados pela rede para a porta de comando não segura, incluindo a senha de acesso do MQIPT, podem ser visíveis para outros usuários na rede.

Para que o MQIPT atenda em uma porta de comando para comandos emitidos pelo comando **mqiPTAdmin**, um valor deve ser especificado para as propriedades **CommandPort** ou **SSLCommandPort** na seção global do arquivo de configuração `mqiPT.conf`.

Revise as considerações de segurança em [Outras considerações de segurança](#) antes de ativar qualquer uma das portas de comando do MQIPT. Considere ativar a autenticação para comandos recebidos pelas portas de comando. Para obter mais informações sobre autenticação de porta de comando, consulte [“Autenticação de porta de comando”](#) na página 579.

Para administrar uma instância do MQIPT usando uma porta de comando, especifique o endereço de rede do host no qual o MQIPT está em execução e o número da porta de comando, como os parâmetros para o comando **mqiPTAdmin**. Por exemplo, para atualizar a instância do MQIPT que está em execução no `mqiPT.example.com` e possui a porta de comando não segura configurada para atender na porta 1890, emita o comando a seguir:

```
mqiPTAdmin -refresh -r mqiPT.example.com:1890
```

Se você não especificar o nome do host e o número da porta, **mqiPTAdmin** tentará se conectar a `localhost`, a porta 1881.

Para obter mais informações sobre a administração do MQIPT usando a porta de comando TLS, consulte [“Administrando o MQIPT por meio da porta de comando TLS”](#) na página 575.



Administrando o MQIPT por meio da porta de comando TLS

MQIPT pode ser configurado para usar uma porta de comandos TLS para atender comandos administrativos emitidos pelo comando **mqiPTAdmin**. O uso da porta de comando TLS protege dados sensíveis, como a senha de acesso do MQIPT, na rede entre o **mqiPTAdmin** e o MQIPT. Use este procedimento para configurar a porta de comando TLS e administrar o MQIPT por meio dela.

Sobre esta tarefa

A porta de comando TLS deve ser configurada com um certificado do servidor armazenado em um keystore PKCS #12 ou em um hardware criptográfico que suporte a Interface do Token Criptográfico PKCS #11. O certificado do servidor da porta de comando é enviado para o comando **mqiptAdmin** durante o handshake TLS. Essa tarefa supõe que você solicite um novo certificado de servidor a uma autoridade de certificação (CA) confiável e que ele seja retornado a você em um arquivo. O comando **mqiptAdmin** valida o certificado da porta de comando usando o certificado da autoridade de certificação que assinou o certificado do servidor. O certificado CA deve ser armazenado em um keystore PKCS #12 que possa ser acessado pelo comando **mqiptAdmin**.



A autenticação de certificado de cliente não é suportada pela porta de comando TLS. Para ativar a autenticação para comandos administrativos emitidos para uma porta de comando, consulte [“Autenticação de porta de comando” na página 579](#).

Este procedimento descreve como gerenciar os keystores e certificados digitais que são necessários para usar a porta de comando TLS usando o comando   **mqiptKeytool**. Para obter mais informações sobre como gerenciar keystores que o MQIPT usa, consulte [Gerenciando MQIPT keystores](#).

Procedimento

1. Siga estas etapas para configurar a porta de comando TLS para a instância do MQIPT.

- a) Crie um par de chave pública e privada, e um certificado do servidor da porta de comando TLS associado em um keystore PKCS #12

  Para criar o keystore que contém o certificado do servidor da porta de comando TLS, insira o comando a seguir:

```
mqiptKeytool -genkeypair -keystore filename -storetype pkcs12 -storepass password
             -dname distinguished_name -alias label
             -keyalg key_algorithm -keysize key_size -sigalg sig_algorithm
```

em que:

-keystore nome do arquivo

Especifica o nome do keystore.

-storepass senha

Especifica a senha do keystore.

-alias label

Especifica o rótulo certificado.

-keyalg key_algorithm

Especifica o algoritmo usado para criar o par de chaves.

-keysize key_size

Especifica o tamanho da chave.

-sigalg algoritmo

Especifica o algoritmo usado para assinar o certificado.

-dname distinguished_name

Especifica o nome distinto X.500 colocado entre aspas duplas.

- b) Crie uma solicitação de certificado para o certificado do servidor da porta de comando TLS assinado pela autoridade de certificação.

  Para criar uma solicitação de certificado, insira o comando a seguir:

```
mqiptKeytool -certreq -keystore filename -storetype pkcs12 -storepass password
             -alias label -file certreq_filename
```

em que:

-keystore nome do arquivo

Especifica o nome do keystore.

-storepass senha

Especifica a senha do keystore.


-alias label

Especifica o rótulo certificado.

-file certreq_filename

Especifica o nome do arquivo para a solicitação de certificado.

- c) Envie o arquivo de solicitação de certificado criado na etapa [“1.b”](#) na página 576 para ser assinado por sua autoridade de certificação.
- d) Depois que a CA enviar o certificado assinado, receba o certificado assinado no keystore.

 Para receber o certificado assinado no keystore, insira o comando a seguir:

```
mqiptKeytool -importcert -keystore cert_filename -storetype pkcs12 -storepass password  
-file cert_filename
```

em que *cert_filename* é o nome do arquivo que contém o certificado, *filename* é o nome do keystore e *password* é a senha do keystore.

- e) Criptografe a senha do keystore usando o comando **mqiptPW**.

Insira o seguinte comando:

```
mqiptPW -sf encryption_key_file
```

em que *encryption_key_file* é o nome de um arquivo que contém a chave de criptografia de senha para sua instalação do MQIPT. Não é preciso especificar o parâmetro **-sf** caso sua instalação MQIPT esteja usando a chave de criptografia de senha padrão. Digite a senha do keystore para criptografar quando solicitado.

Para obter mais informações sobre o comando **mqiptPW**, consulte [Criptografando a senha de um conjunto de chaves](#).

- f) Edite o arquivo de configuração do `mqipt.conf` e especifique as propriedades a seguir para configurar a porta de comando do TLS:
 - i) Configure o valor da propriedade **SSLCommandPort** para o número da porta de comando TLS.
 - ii) Configure o valor da propriedade **SSLCommandPortKeyRing** para o nome de arquivo do keystore criado na etapa [“1.a”](#) na página 576.
 - iii) Configure o valor de **SSLCommandPortKeyRingPW** como a saída de sequência pelo comando **mqiptPW** na etapa [“1.e”](#) na página 577.
 - iv) Configure o valor da propriedade **SSLCommandPortSiteLabel** para o nome do rótulo do certificado de porta de comando TLS, especificado ao criar a solicitação de certificado na etapa [“1.b”](#) na página 576.
 - v) Se você deseja restringir as conexões de entrada para a porta de comando TLS àquelas de uma interface de rede específica, configure o valor da propriedade **SSLCommandPortListenerAddress** para um endereço de rede pertencente a uma das interfaces de rede no sistema em que MQIPT está em execução. Por exemplo, para restringir as conexões de entrada à porta de comando TLS àquelas apenas da máquina local, defina o valor da propriedade **SSLCommandPortListenerAddress** como `localhost`.
- g) Inicie ou atualize o MQIPT para ativar a porta de comando TLS.


O MQIPT emite mensagens de console como a seguinte para exibir a configuração da porta de comando TLS que está em vigor:

```
MQCPI155 Listening for control commands on port 1882 on local address * using TLS  
MQCPI139 .....secure socket protocols <NULL>  
MQCPI031 .....cipher suites <NULL>
```

```
MQCPI032 .....key ring file c:\\iptHome\\ssl\\commandport.p12
MQCPI072 .....and certificate label mqiptadmin
```

2. No sistema no qual o comando **mqiptAdmin** é usado para administrar o MQIPT, siga estas etapas para ativar **mqiptAdmin** a fim de se conectar à porta de comando TLS.

- a) Importe o certificado de autoridade de certificação da autoridade de certificação que assinou o certificado da porta de comando TLS para um keystore PKCS #12 a ser usado como um armazenamento confiável por um comando **mqiptAdmin**

 Para importar o certificado de CA, insira o comando a seguir:

```
mqiptKeytool -importcert -keystore filename -storetype pkcs12 -storepass password
              -file cert_filename -alias certlabel
```

em que:

nome_do_arquivo

Especifica o nome do keystore a ser criado

senha

Especifica a senha do keystore

certlabel

Especifica o rótulo a ser fornecido ao certificado de autoridade de certificação

cert_filename

Especifica o nome do arquivo que contém o certificado de autoridade de certificação.

- b) Criptografe a senha do keystore usando o comando **mqiptPW**.

Insira o seguinte comando:

```
mqiptPW -sf encryption_key_file
```

em que *encryption_key_file* é o nome do arquivo que contém a chave de criptografia de senha. O arquivo de chave de criptografia de senha pode ser diferente do usado para criptografar senhas na configuração do MQIPT. A chave de criptografia de senha padrão é usada se você não especificar um arquivo de chave de criptografia com o parâmetro **-sf**. Digite a senha do keystore para criptografar quando solicitado.

Para obter mais informações sobre o comando **mqiptPW**, consulte [Criptografando a senha de um conjunto de chaves](#).

- c) Crie um arquivo de propriedades para ser usado pelo comando **mqiptAdmin** e especifique as propriedades a seguir:

```
SSLClientCAKeyRing=key_ring_file_name
SSLClientCAKeyRingPW=key_ring_password
PasswordProtectionKeyFile=encryption_key_file
```

em que:

key_ring_file_name

é o nome do keystore criado na etapa “2.a” na página 578.

key_ring_password

é a saída da senha criptografada pelo comando **mqiptPW** na etapa “2.b” na página 578.

encryption_key_file

é o nome do arquivo que contém a chave de criptografia de senha. Será necessário especificar a propriedade **PasswordProtectionKeyFile** apenas se um arquivo de chave de criptografia tiver sido usado para criptografar a senha do keystore na etapa “2.b” na página 578

- d) Emita o comando **mqiptAdmin** para administrar o MQIPT, especificando o parâmetro **-s** para indicar que uma conexão TLS é necessária e o parâmetro **-p** para especificar o nome do arquivo de propriedades que foi criado na etapa “2.c” na página 578.

Por exemplo, insira o comando a seguir para atualizar uma instância do MQIPT enviando um comando de atualização para a porta de comando TLS:

```
mciptAdmin -refresh -r hostname:port -s -p properties_file
```

O comando **mciptAdmin** emite uma mensagem como a seguinte para confirmar se a conexão com o MQIPT está protegida com TLS:

```
MQCAI109 The connection to MQIPT is secured with TLSv1.2.
```

Como proceder a seguir

Para ativar a autenticação para comandos recebidos pela porta de comando TLS, siga as etapas em [“Autenticação de porta de comando”](#) na página 579.

Autenticação de porta de comando

O MQIPT pode ser configurado para autenticar comandos recebidos pela porta de comando não segura e pela porta de comando TLS usando uma senha. Use este procedimento para ativar a autenticação da porta de comando.

Sobre esta tarefa

O comando **mciptAdmin** solicita que os usuários insiram uma senha quando o comando se conecta à porta de comando de uma instância do MQIPT que tem a autenticação da porta de comando ativada. O MQIPT valida a senha inserida no comando **mciptAdmin** com relação à senha de acesso especificada na configuração do MQIPT.

As propriedades que você configurou para a autenticação da porta de comando se aplicam tanto à porta de comando TLS quanto à porta de comando não segura.

Procedimento

1. Criptografe a senha de acesso do MQIPT usando o comando **mciptPW**.

Insira o seguinte comando:

```
mciptPW -sf encryption_key_file
```

em que *encryption_key_file* é o nome do arquivo que contém a chave de criptografia de senha para a sua instalação do MQIPT. Não é preciso especificar o parâmetro **-sf** caso sua instalação MQIPT esteja usando a chave de criptografia de senha padrão. Quando solicitado, digite a senha de acesso para a criptografia.

Para obter mais informações sobre como criptografar senhas na configuração do MQIPT, consulte [Criptografando senhas armazenadas](#).

2. Edite o arquivo de configuração `mcipt.conf` e especifique as seguintes propriedades:

```
AccessPW=encrypted_password  
RemoteCommandAuthentication=auth_setting
```

em que:

encrypted_password

é a saída da senha criptografada pelo comando **mciptPW** na etapa [“1”](#) na página 579.

auth_setting

é o requisito de autenticação. A autenticação da porta de comando será ativada se essa propriedade for configurada para um dos valores a seguir:

opcional

Não é necessário fornecer uma senha, mas se ela for fornecida, deverá ser válida. Esta opção pode ser útil durante a migração, por exemplo.

requeridos

Uma senha válida deve ser fornecida com cada comando recebido por uma porta de comando.

Para obter mais informações sobre essas propriedades, consulte [Propriedades globais do MQIPT](#).

3. Inicie ou atualize o MQIPT para que as mudanças entrem em vigor.

O MQIPT emite uma mensagem indicando se a autenticação da porta de comando está ativada. Por exemplo, se o MQIPT estiver configurado para requerer que uma senha válida seja inserida sempre que o comando **mqiptAdmin** for executado, a mensagem a seguir será emitida:

```
MQCPI021 Password checking has been enabled on the command port
```

Fazendo backups

Há um número de arquivos MQIPT que você deve fazer backup como parte de seus procedimentos de backup regulares.

Faça backup dos seguintes arquivos em uma base regular:

- O arquivo de configuração, `mqipt.conf`
- Os arquivos do conjunto de chaves SSL/TLS especificados pelas propriedades a seguir em `mqipt.conf`:
 - **SSLClientKeyRing**
 - **SSLClientCAKeyRing**
 - **SSLServerKeyRing**
 - **SSLServerCAKeyRing**
 - **SSLCommandPortKeyRing**
- Os arquivos de senha do conjunto de chaves SSL/TLS especificados pelas propriedades a seguir em `mqipt.conf`:
 - **SSLClientKeyRingPW**
 - **SSLClientCAKeyRingPW**
 - **SSLServerKeyRingPW**
 - **SSLServerCAKeyRingPW**
- O arquivo de chave de criptografia de senha, se a configuração do MQIPT contiver senhas criptografadas com uma chave de criptografia diferente da chave padrão.
- O arquivo de políticas especificado por **SecurityManagerPolicy**, se essa propriedade tiver sido configurada.
- Os arquivos de saída de segurança e os arquivos de saída de certificado especificados pelas propriedades a seguir em `mqipt.conf`:
 - **SecurityExitName**
 - **SSLExitName**
- Arquivos de log de conexão no subdiretório `log` do diretório inicial MQIPT, se forem necessários para fins de auditoria.

Ajuste de desempenho

É possível ajustar o desempenho relativo de cada rota do MQIPT usando uma combinação de conjunto de encadeamentos e uma especificação de tempo limite inativo.

Linhas de conexão

Cada rota MQIPT é designada a um conjunto de trabalho de encadeamentos em execução simultânea que manipulam solicitações de comunicação recebidas. Na inicialização, um conjunto de

encadeamentos é criado (do tamanho especificado no atributo `MinConnectionThreads` da rota) e um encadeamento é designado para manipular a primeira solicitação recebida. Quando essa solicitação chega, outro encadeamento é designado, pronto para a próxima solicitação recebida. Quando todos os encadeamentos são designados para o trabalho, um novo encadeamento é criado, incluído no conjunto de trabalho e designado para o trabalho.

Desta maneira, o conjunto cresce até que o número máximo de encadeamentos (especificado em **MaxConnectionThreads**) seja atingido. Os encadeamentos são liberados de volta para o conjunto quando uma conversa termina, ou o período de tempo limite inativo especificado foi decorrido. Quando o número máximo de encadeamentos de trabalho é atingido, a próxima solicitação recebida aguarda até que um encadeamento seja liberado de volta para o conjunto de trabalho.

É possível reduzir o tempo em que as solicitações podem ter que esperar aumentando o número de encadeamentos disponíveis. No entanto, você deve equilibrar esse aumento com os recursos do sistema que estão disponíveis.

Tempo Limite de Inatividade

Por padrão, os encadeamentos de trabalho não são finalizados devido à inatividade. Quando um encadeamento tiver sido designado a uma conversa, ela permanecerá designada a essa conversa até que ela seja fechada normalmente, a rota será desativada ou MQIPT será encerrado. Opcionalmente, é possível especificar um intervalo de tempo limite inativo (em minutos) na propriedade **IdleTimeout** para que os encadeamentos que foram inativos para o período de tempo especificado sejam reciclados. Os encadeamentos são reciclados para uso colocando-os de volta no conjunto de trabalho.

Se a atividade IBM MQ for intermitente, configure seu intervalo de pulsação para um valor menor que o do tempo de espera MQIPT para que os encadeamentos não sejam constantemente reciclados.

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte seu representante local do IBM para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja nenhum direito de propriedade intelectual da IBM. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou aplicativos de patentes pendentes relativas aos assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum sobre tais patentes. É possível enviar pedidos de licença, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
U.S.A.

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. IBM pode aperfeiçoar e/ou alterar no produto(s) e/ou programa(s) descritos nesta publicação a qualquer momento sem aviso prévio.

Todas as referências nessas informações a websites não IBM são fornecidas somente por conveniência e de forma alguma são um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Os licenciados deste programa que desejarem obter informações sobre este assunto com o propósito de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações trocadas, deverão entrar em contato com:

Av. Pasteur, 138-146
Av. Pasteur, 138-146

Botafogo
Rio de Janeiro, RJ
U.S.A.

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do IBM Customer Agreement, IBM Contrato de Licença do Programa Internacional ou qualquer contrato equivalente entre as partes.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disto, algumas medidas podem ter sido estimadas através de extrapolação. Os resultados reais podem variar. usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam somente metas e objetivos.

Essas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas.

Se estiver visualizando estas informações em formato eletrônico, as fotografias e ilustrações coloridas poderão não aparecer.

Informações sobre a Interface de Programação

As informações da interface de programação, se fornecidas, destinam-se a ajudá-lo a criar software aplicativo para uso com este programa.

Este manual contém informações sobre as interfaces de programação desejadas que permitem que o cliente grave programas para obter os serviços do IBM MQ

No entanto, estas informações também podem conter informações sobre diagnósticos, modificações e ajustes. As informações sobre diagnósticos, modificações e ajustes são fornecidas para ajudá-lo a depurar seu software aplicativo.

Importante: Não use essas informações de diagnóstico, modificação e ajuste como uma interface de programação, pois elas estão sujeitas a mudanças

Marcas comerciais

IBM, o logotipo IBM , ibm.com, são marcas registradas da IBM Corporation, registradas em várias jurisdições no mundo todo Uma lista atual de marcas registradas da IBM está disponível na Web em "Informações de copyright e marca registrada" www.ibm.com/legal/copytrade.shtml. Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas.

Microsoft e Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Este produto inclui software desenvolvido pelo Projeto Eclipse (<https://www.eclipse.org/>).

Java e todas as marcas registradas e logotipos baseados em Java são marcas ou marcas registradas da Oracle e/ou de suas afiliadas.



Part Number:

(1P) P/N: