

9.3

Visão geral técnica do IBM MQ

IBM

Nota

Antes de usar estas informações e o produto que elas suportam, leia as informações em [“Avisos” na página 319](#).

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

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

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

Índice

Visão geral técnica.....	5
Introdução ao Enfileiramento de Mensagens.....	5
Principais Recursos e Benefícios do Enfileiramento de Mensagens.....	7
Terminologia de Enfileiramento de Mensagens.....	9
Mensagens e Filas.....	13
Objetos do IBM MQ.....	14
Tipos de objeto.....	16
Nomeando os Objetos IBM MQ.....	37
Enfileiramento distribuído e clusters.....	44
Componentes de Enfileiramento Distribuído.....	48
Componentes do Cluster.....	58
Sistema de Mensagens de Publicação/Assinatura.....	64
Componentes publicar/assinar.....	65
Exemplo da Configuração de um Único Gerenciador de Filas de Publicação/Assinatura.....	91
Redes publicar/assinar distribuídas.....	91
IBM MQ Multicast.....	109
Conceitos Iniciais e Multicast.....	110
Visão geral do MQ Telemetry.....	111
Introdução ao MQ Telemetry.....	112
Casos de Uso de Telemetria.....	114
Conectando Dispositivos de Telemetria a um Gerenciador de Filas.....	120
Protocolos de Conexão do Telemetry.....	121
Serviço de telemetria (MQXR).....	121
Canais de Telemetria.....	121
Protocolo IBM MQ Telemetry Transport.....	122
Clientes MQTT.....	122
Envie uma mensagem para um cliente MQTT.....	123
Enviando uma mensagem para um aplicativo IBM MQ por meio de um cliente do MQTT.....	132
Aplicativos de publicação/assinatura do MQTT.....	133
Aplicativos de Telemetria.....	133
Integração do MQ Telemetry com gerenciadores de filas.....	134
Sessões stateless e stateful do MQTT.....	136
Quando um MQTT do cliente não está conectado.....	137
Loose coupling entre clientes MQTT e aplicativos IBM MQ.....	137
MQ Telemetrysegurança.....	138
Globalização do MQ Telemetry.....	139
Desempenho e escalabilidade do MQ Telemetry.....	139
Dispositivos suportados pelo MQ Telemetry.....	141
Segurança no IBM MQ.....	142
Suporte de TLS do cliente gerenciado pelo IBM MQ.NET.....	143
IBM MQ MQI clients.....	144
Por que usar clientes IBM MQ?.....	146
O que É um Cliente Transacional Estendido?.....	148
Como o Cliente se Conecta ao Servidor.....	149
Gerenciamento de Transação e Suporte.....	150
Estendendo as instalações do gerenciador de filas.....	152
Interfaces de linguagem do IBM MQ Java.....	153
IBM MQ classes for JMS/Jakarta Messaging.....	154
Provedor de sistema de mensagens do IBM MQ.....	165
Conceitos do IBM MQ for z/OS.....	165
O gerenciador de filas no z/OS.....	167
O inicializador de canais no z/OS.....	168

Termos e tarefas para gerenciar o IBM MQ for z/OS.....	169
Filas compartilhadas e grupos de filas compartilhadas.....	172
Fila do intra-grupo.....	220
Gerenciamento de armazenamento no z/OS.....	234
Efetuação de login no IBM MQ for z/OS.....	239
Definição do sistema em z/OS.....	250
Recuperação e reinicialização no z/OS.....	261
Conceitos de segurança no IBM MQ for z/OS.....	280
Disponibilidade no z/OS.....	286
Monitoramento e estatísticas no IBM MQ for z/OS.....	290
Disposição de unidade de recuperação no z/OS.....	291
IBM MQ e outros Produtos z/OS.....	294
IBM MQ e CICS.....	294
IBM MQ e IMS.....	295
Adaptadores IBM MQ e z/OS Batch, TSO e RRS.....	299
IBM MQ for z/OS e WebSphere Application Server.....	301
Managed File Transfer.....	301
Como o MFT trabalha com o IBM MQ?.....	304
Visão Geral da Topologia do MFT.....	304
Visão Geral do MFT REST API.....	305
IBM MQ Internet Pass-Thru.....	306
Usos do MQIPT.....	307
Como o MQIPT funciona.....	309
Possíveis configurações do MQIPT.....	310
Configurações compatíveis.....	312
Configurações de Canal Suportadas.....	314
Condições de término e falha do canal.....	315
Segurança de Mensagens.....	315
Gerenciadores de filas de várias instâncias e alta disponibilidade.....	315
O IBM MQ Console e REST API.....	316
Avisos.....	319
Informações sobre a Interface de Programação.....	320
Marcas comerciais.....	321

Visão Geral Técnica do IBM MQ

Use o IBM MQ para conectar seus aplicativos e gerenciar a distribuição de informações em sua organização.

O IBM MQ permite que programas se comuniquem uns com os outros por meio de uma rede de componentes diferentes (processadores, sistemas operacionais, subsistemas e protocolos de comunicação) usando uma interface de programação de aplicativos consistente. Os aplicativos projetados e gravados usando esta interface são conhecidos como aplicativos de enfileiramento de mensagens.

Use os subtópicos a seguir para descobrir sobre o enfileiramento de mensagens e outros recursos fornecidos pelo IBM MQ.

Conceitos relacionados

[Introdução ao IBM MQ](#)

[Onde localizar informações de requisitos e suporte do produto](#)

Tarefas relacionadas

[Planejando uma arquitetura do IBM MQ](#)

Referências relacionadas

[“Principais Recursos e Benefícios do Enfileiramento de Mensagens” na página 7](#)

Essas informações destacam alguns recursos e benefícios do enfileiramento de mensagens. Descreve recursos como segurança e integridade de dados do enfileiramento de mensagens.

Introdução ao Enfileiramento de Mensagens

Os produtos IBM MQ permitem que os programas se comuniquem entre si em uma rede de componentes diferentes (processadores, sistemas operacionais, subsistemas e protocolos de comunicação) usando uma interface de programação de aplicativos consistente.

Os aplicativos designados e gravados usando esta interface são mostrados como aplicativos de *enfileiramento de mensagens*, porque usam o estilo de *sistema de mensagens* e de *enfileiramento*:

- Sistema de mensagens significa que os programas se comunicam enviando uns aos outros dados em mensagens, em vez de chamar uns aos outros diretamente.
- Enfileiramento significa que as mensagens são colocadas em filas no armazenamento, permitindo que os programas sejam executados independentemente uns dos outros, em velocidades e horários diferentes, em locais diferentes e sem que haja uma conexão lógica entre eles.

O enfileiramento de mensagens tem sido usado no processamento de dados por vários anos. Hoje ele é frequentemente usado no correio eletrônico. Sem o enfileiramento, o envio de uma mensagem eletrônica a longas distâncias requer que cada nó na rota fique disponível para o encaminhamento de mensagens e que os endereços sejam registrados e reconheçam que você está tentando enviar uma mensagem a eles. Em um sistema de enfileiramento, as mensagens são armazenadas nos nós intermediários até que o sistema fique pronto para encaminhá-las. Em seus destinos finais, elas são armazenadas em uma caixa de correio eletrônico até que o destinatário esteja pronto para lê-las.

Mesmo assim, muitas transações de negócios complexas são processadas atualmente sem o enfileiramento. Em uma rede ampla, o sistema pode estar mantendo milhares de conexões em um estado pronto para uso. Se uma parte do sistema sofrer um problema, várias partes do sistema podem ficar inutilizáveis.

É possível imaginar um enfileiramento de mensagens como sendo o correio eletrônico para os programas. Em um ambiente de enfileiramento de mensagens, cada programa que faz parte de um conjunto de aplicativos executa uma função autocontida e bem definida em resposta a uma solicitação específica. Para comunicar-se com outro programa, um programa deve colocar uma mensagem em uma fila predefinida. O outro programa recupera a mensagem da fila e processa as solicitações e as informações contidas na mensagem. Portanto, o enfileiramento de mensagens é um estilo de comunicação programa-a-programa.

O enfileiramento é o mecanismo pelo qual as mensagens são retidas até que um aplicativo esteja pronto para processá-las. O enfileiramento permite:

- Comunicar-se entre os programas (que podem estar em execução em ambientes diferentes) sem ter que gravar o código de comunicação.
- Selecionar a ordem na qual um programa processa as mensagens.
- Equilibrar os carregamentos em um sistema organizando mais de um programa para atender uma fila quando o número de mensagens exceder um limite.
- Aumentar a disponibilidade dos seus aplicativos organizando um sistema alternativo para atender as filas se o seu sistema primário estiver indisponível.

O que é uma fila de mensagens?

Uma fila de mensagens, conhecida simplesmente como uma fila, é um destino nomeado para o qual as mensagens podem ser enviadas. As mensagens se acumulam em filas até que sejam recuperadas pelos programas que atendem essas filas.

Filas residem em, e são gerenciadas por, um gerenciador de filas (consulte [“Terminologia de Enfileiramento de Mensagens”](#) na página 9). A natureza física de uma fila depende do sistema operacional no qual o gerenciador de filas está sendo executado. Uma fila pode ser área de buffer volátil na memória de um computador ou um conjunto de dados em um dispositivo de armazenamento permanente (como um disco). O gerenciamento físico de filas é a responsabilidade do gerenciador de filas e não se torna aparente aos programas de aplicativo participantes.

Os programas acessam as filas apenas por meio dos serviços externos do gerenciador de filas. Eles podem abrir uma fila, colocar as mensagens nela, obter as mensagens a partir dela e fechar a fila. Eles também podem configurar, e perguntar sobre, os atributos das filas.

Diferentes Estilos de Enfileiramento de Mensagens

Ponto a ponto

Uma mensagem é colocada na fila e um aplicativo recebe essa mensagem.

No sistema de mensagens ponto a ponto, um aplicativo de envio deve ter informações sobre o aplicativo de recebimento antes de poder enviar uma mensagem para ele. Por exemplo, o aplicativo de envio precisará saber o nome da fila para a qual enviar as informações e também poderá especificar um nome do gerenciador de filas.

Publicação/Assinatura

Uma cópia de cada mensagem publicada por um aplicativo de publicação é entregue para cada aplicativo interessado. Pode haver vários, um ou nenhum aplicativo interessado. Na publicação/assinatura, um aplicativo interessado é conhecido como assinante, e as mensagens são enfileiradas em uma fila identificada por uma assinatura.

O sistema de mensagens de publicação/assinatura permite separar o provedor de informações dos consumidores dessas informações. O aplicativo de envio e o aplicativo de recebimento não precisam saber nada um do outro para que as informações sejam enviadas e recebidas. Para obter mais informações, consulte [“Sistema de Mensagens de Publicação/Assinatura”](#) na página 64.

Benefícios do Enfileiramento de Mensagens para o Editor de Telas e Desenvolvedor

O IBM MQ permite que os programas de aplicativo usem o *enfileiramento de mensagens* para participar no processamento orientado à mensagem. Os programas de aplicativo podem se comunicar entre diferentes plataformas usando os produtos de software de enfileiramento de mensagens apropriados. Por exemplo, os aplicativos z/OS podem se comunicar por meio de IBM MQ for z/OS. Os aplicativos são blindados a partir das mecânicas das comunicações subjacentes. Alguns outros benefícios do enfileiramento de mensagens são:

- É possível projetar os aplicativos usando pequenos programas que podem ser compartilhados entre vários aplicativos.
- É possível construir novos aplicativos reutilizando esses blocos de construção.
- Os aplicativos gravados para usar as técnicas de enfileiramento de mensagens não são afetados por mudanças na maneira como esses gerenciadores de filas trabalham.
- Você não precisa usar qualquer protocolo de comunicação. O gerenciador de filas lida com todos os aspectos da comunicação.
- Os programas que recebem as mensagens não precisam estar em execução no momento em que as mensagens são enviadas. As mensagens são retidas nas filas.

Os designers podem reduzir o custo de seus aplicativos porque o desenvolvimento é mais rápido, menos desenvolvedores são necessários e as demandas na qualificação de programação são menores do que para os aplicativos que não usam o enfileiramento de mensagens.

IBM MQ implementa uma interface de programação de aplicativos comum conhecida como a *interface de fila de mensagens* (ou MQI) sempre que os aplicativos são executados. Isso facilita portar os programas de aplicativos de uma plataforma a outra.

Para obter detalhes sobre o MQI, consulte [Visão geral do Message Queue Interface](#).

Principais Recursos e Benefícios do Enfileiramento de Mensagens

Essas informações destacam alguns recursos e benefícios do enfileiramento de mensagens. Descreve recursos como segurança e integridade de dados do enfileiramento de mensagens.

Os principais recursos de aplicativos que usam as técnicas de enfileiramento de mensagens são:

- [“Sem conexões diretas entre os programas” na página 7](#)
- [“Comunicação independente de horário” na página 8](#)
- [“Programas pequenos” na página 8](#)
- [“Processamento orientado a mensagens” na página 8](#)
- [“Processamento Direcionado a Eventos” na página 9](#)
- [“Prioridade da mensagem” na página 9](#)
- [“Segurança” na página 9](#)
- [“Integridade de dados” na página 9](#)
- [“Suporte à recuperação” na página 9](#)

Nota: Ao considerar clientes e servidores IBM MQ, não é necessário alterar um aplicativo do servidor para suportar IBM MQ MQI clients adicionais em novas plataformas. Da mesma forma, o IBM MQ MQI client pode, sem mudança, funcionar com tipos adicionais de servidores.

Sem conexões diretas entre os programas

O enfileiramento de mensagem é uma técnica para comunicação indireta programa a programa. Ele pode ser usado em qualquer aplicativo no qual os programas se comunicam entre si. A comunicação ocorre por um programa que coloca as mensagens em uma fila (de propriedade de um gerenciador de filas) e outro programa que recebe as mensagens da fila.

Os programas podem receber as mensagens que foram colocadas em uma fila por outros programas. Os outros programas podem ser conectados no mesmo gerenciador de filas como o programa de recebimento ou em outro gerenciador de filas. Este outro gerenciador de filas pode estar em outro sistema, um sistema de computador diferente ou até em um negócio ou empreendimento diferente.

Não existem conexões físicas entre os programas que se comunicam usando as filas de mensagens. Um programa envia mensagens em uma fila de propriedade de um gerenciador de filas e outro programa recupera as mensagens da fila (consulte [Figura 1 na página 8](#)).

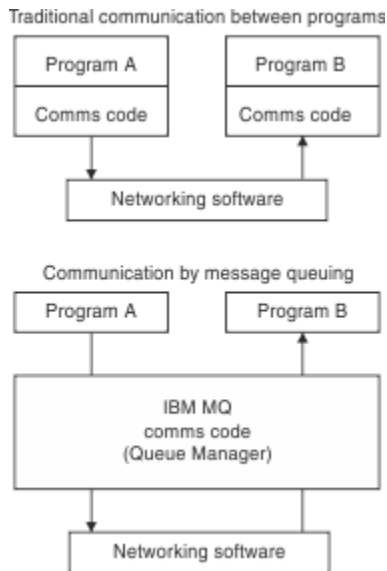


Figura 1. Enfileiramento de Mensagens Comparado à Comunicação Tradicional

Assim como no correio eletrônico, as mensagens individuais que fazem parte de uma transação viajam por meio de uma rede com base em um armazenamento e encaminhamento. Se um link entre os nós falhar, a mensagem será mantida até que o link seja restaurado ou o operador ou o programa redirecione a mensagem.

O mecanismo pelo qual uma mensagem move de fila para fila é oculto dos programas. Portanto, os programas são mais simples.

Comunicação independente de horário

Os programas que precisam de outros para executar o trabalho não precisam aguardar pela resposta para uma solicitação. Eles podem realizar outros trabalhos e processar a resposta quando chegar ou mais recentemente. Ao gravar um aplicativo de mensagens, não é necessário saber (ou se preocupar) quando um programa enviar uma mensagem ou quando o destino conseguir receber a mensagem. A mensagem não é perdida; ela é retida pelo gerenciador de filas até que o destino esteja pronto para processá-la. A mensagem permanece na fila até que seja removida por um programa. Isso significa que os programas de aplicativo de envio e recebimento são separados; o emissor pode continuar processando sem aguardar que o receptor reconheça o recebimento da mensagem. O aplicativo de destino não precisa estar em execução quando a mensagem for enviada. Ele pode recuperar a mensagem depois que ela tiver sido iniciada.

Programas pequenos

O enfileiramento de mensagens permite as vantagens de usar programas pequenos e autocontidos. Em vez de um único programa grande executando duas partes de uma tarefa sequencialmente, é possível espalhar a tarefa em diversos programas independentes menores. O programa de solicitação envia as mensagens a cada um dos programas separados, pedindo que executem suas funções; quando cada programa for concluído, os resultados serão enviados de volta como uma ou mais mensagens.

Processamento orientado a mensagens

Quando as mensagens chegarem a uma fila, elas poderão iniciar automaticamente um aplicativo usando *triggering*. Se necessário, os aplicativos podem ser interrompidos quando a mensagem (ou mensagens) tiverem sido processadas.

Processamento Direcionado a Eventos

Os programas podem ser controlados de acordo com o estado das filas. Por exemplo, é possível organizar um programa para que seja iniciado assim que uma mensagem chegar a uma fila ou é possível especificar que o programa não seja iniciado até que existam, por exemplo, 10 mensagens acima de uma certa prioridade na fila ou 10 mensagens de qualquer prioridade na fila.

Prioridade da mensagem

Um programa pode designar uma prioridade a uma mensagem quando colocar a mensagem em uma fila. Isso determina a posição na fila na qual a nova mensagem é incluída.

Os programas podem receber mensagens de uma fila na ordem em que as mensagens ficam na fila ou recebendo uma mensagem específica. (Talvez um programa queira obter uma mensagem específica, se estiver procurando pela resposta a uma solicitação que enviou anteriormente.)

Segurança

São fornecidos recursos de segurança, incluindo autenticação de aplicativos quando eles usam um gerenciador de filas, verificações de autorização quando eles usam recursos, como uma fila no gerenciador de filas e criptografia de dados da mensagem conforme eles circulam pela rede e enquanto residem em filas. Para obter informações adicionais sobre segurança, consulte [Visão geral de segurança](#).

Integridade de dados

A integridade de dados é fornecida por unidades de trabalho. A sincronização do início e do término das unidades de trabalho é totalmente suportada como uma opção em cada MQGET ou MQPUT, permitindo que os resultados da unidade de trabalho sejam confirmados ou retrocedidos. O suporte de ponto de sincronização opera interna ou externamente no IBM MQ, dependendo da forma de coordenação de ponto de sincronização selecionada para o aplicativo.

Suporte à recuperação



Para que a recuperação seja possível, todas as atualizações persistentes do IBM MQ são registradas. Se a recuperação for necessária, todas as mensagens persistentes serão restauradas, todas as transações em trânsito serão retrocedidas e qualquer confirmação de ponto de sincronização e restaurações serão tratadas da maneira normal no gerenciador de ponto de sincronização em controle. Para obter informações adicionais sobre mensagens persistentes, consulte [Mensagens persistentes](#).

Terminologia de Enfileiramento de Mensagens

Essas informações fornecem um insight em alguns termos usados no enfileiramento de mensagens.

São eles:

- [Canais](#)
- [Cluster](#)
- [IBM MQ MQI client](#)
-  [Enfileiramento intragrupo](#)
- [Mensagem](#)
- [Agente do canal de mensagens](#)
- [Descritor de mensagens](#)
- [Ponto a ponto](#)
- [Publicação/Assinatura](#)
- [Fila](#)
- [Gerenciador de Filas](#)

-  [Grupo de filas compartilhadas](#)
-  [Fila compartilhada](#)
- [Assinatura](#)
- [Tópico](#)

Canais

Os canais são usados para mover as mensagens de um gerenciador de filas para outro e protegem os aplicativos dos protocolos de comunicações subjacentes. Os gerenciadores de filas podem existir no mesmo sistema ou em sistemas diferentes na mesma plataforma ou ainda em plataformas diferentes. As mensagens que são enviadas podem se originar de vários locais:

- Programas de aplicativo gravados pelo usuário que transferem dados de um nó para outro.
- Aplicativos de administração gravados pelo usuário que usam comandos PCF ou a MQAI.
- O IBM MQ Explorer.
- Gerenciadores de filas que enviam mensagens de evento de instrumentação para outro gerenciador de filas.
- Gerenciadores de filas que enviam comandos de administração remota para outro gerenciador de filas. Por exemplo, usando comandos MQSC ou o administrative REST API.

Para obter mais informações sobre os canais, consulte [“Definições de canal”](#) na página 33

Cluster

Um *cluster* é uma rede de gerenciadores de filas que estão associados logicamente de alguma maneira.

Em uma rede do IBM MQ que usa enfileiramento distribuído sem cluster, cada gerenciador de filas é independente. Se um gerenciador de filas precisar enviar as mensagens para outro, ele deve ter definir uma fila de transmissão e um canal para outro gerenciador de filas remotas.

Existem duas razões diferentes para usar os clusters: reduzir a administração do sistema e melhorar a disponibilidade e o balanceamento de carga de trabalho.





Assim que estabelecer até mesmo o menor de todos os clusters, você será beneficiado pela administração de sistema simplificada. Os gerenciadores de fila que fazem parte de um cluster precisarão de menos definições e assim o risco de cometer um erro nas suas definições é reduzido.

Para obter informações adicionais sobre armazenamento em cluster, consulte [Clusters](#).

IBM MQ MQI client

Os IBM MQ clientes MQI são componentes instaláveis independentemente do IBM MQ Um cliente do MQI permite que você execute aplicativos IBM MQ com um protocolo de comunicação, para interagir com um ou mais servidores do Message Queue Interface (MQI) em outras plataformas e para conectar-se a seus gerenciadores de filas.

Para obter detalhes completos sobre como instalar e usar os componentes do IBM MQ MQI client, veja os tópicos a seguir:

-  [Instalando um cliente do IBM MQ no AIX](#)
-  [Instalando um cliente do IBM MQ no Linux®](#)
-  [Instalando um cliente do IBM MQ no Windows](#)
-  [Instalando um cliente do IBM MQ no IBM i](#)

e [Configurando conexões entre o servidor e o cliente](#).

Fila do intra-grupo



Os gerenciadores de filas em um grupo de filas compartilhadas podem se comunicar usando canais normais ou é possível usar uma técnica chamada *enfileiramento intragrupo* (IGQ), o que permite executar a transferência de mensagem rápida sem definir os canais. Isso é aplicável apenas ao IBM MQ for z/OS.

Para obter mais informações sobre o enfileiramento intragrupo, consulte [“Fila do intra-grupo” na página 220](#)

Mensagem

No enfileiramento de mensagens, uma mensagem é uma coleta de dados enviados por um programa e pretendido para outro programa. Consulte [mensagens do IBM MQ](#).

Para obter informações sobre os tipos de mensagens, consulte [Tipos de mensagem](#).

Agente do canal de mensagens

Um agente do canal de mensagens é uma extremidade de um canal. Um par de agentes do canal de mensagens, um envio e um recebimento formam um canal e movem mensagens de um gerenciador de filas para outro.

Para obter informações sobre como os agentes do canal de mensagens são usados, consulte [Introdução ao gerenciamento de filas distribuídas](#).

Descritor de Mensagens

Uma mensagem do IBM MQ consiste em informações de controle e dados de aplicativos.

As informações de controle são definidas em uma estrutura de descritor de mensagens (MQMD) e contêm coisas como:

- O tipo da mensagem
- Um identificador para a mensagem
- A prioridade para a entrega da mensagem

A estrutura e o conteúdo dos dados do aplicativo são determinados pelos programas participantes, não pelo IBM MQ.

Para obter mais informações, consulte [MQMD](#).

Sistema de mensagens ponto a ponto

No sistema de mensagens ponto a ponto, cada mensagem passa de um aplicativo produtor para um aplicativo consumidor. As mensagens são transferidas por meio do aplicativo produtor, que as coloca em uma fila da qual o aplicativo consumidor as retira.

Sistema de Mensagens de Publicação/Assinatura

No sistema de mensagens de publicação/assinatura, uma cópia de cada mensagem publicada por um aplicativo de publicação é entregue a cada aplicativo interessado. Pode haver vários, um ou nenhum aplicativo interessado. Na publicação/assinatura, um aplicativo interessado é conhecido como assinante, e as mensagens são enfileiradas em uma fila identificada por uma assinatura.

Para obter informações adicionais, consulte [“Sistema de Mensagens de Publicação/Assinatura” na página 64](#).

Fila

Um destino nomeado para o qual mensagens podem ser enviadas. As mensagens se acumulam em filas até que sejam recuperadas pelos programas que atendem essas filas.

Para obter informações adicionais, consulte [“Filas” na página 20](#).

Gerenciador de filas

Um *gerenciador de filas* é um programa do sistema que fornece serviços de enfileiramento para os aplicativos.

Fornecer uma interface de programação de aplicativos para que os programas possam colocar as mensagens em, e obter as mensagens de, filas. Um gerenciador de filas fornece funções adicionais para que os administradores possam criar novas filas, alterar as propriedades das filas existentes e controlar a operação do gerenciador de filas.

Para que serviços de enfileiramento de mensagens do IBM MQ estejam disponíveis em um sistema, deve haver um gerenciador de filas em execução. É possível ter mais de um gerenciador de filas em execução em um único sistema (por exemplo, para separar um sistema de teste de um sistema de *produção*). Para um aplicativo, cada gerenciador de filas é identificado por uma *manipulação de conexões (Hconn)*.

Muitos aplicativos diferentes podem usar os serviços do gerenciador de filas ao mesmo tempo, e esses aplicativos podem estar inteiramente desvinculados. Para que um programa use os serviços de um gerenciador de filas, ele deve estabelecer uma conexão a esse gerenciador de filas.

Para que os aplicativos enviem as mensagens aos aplicativos que estão conectados a outros gerenciadores de fila, os gerenciadores de fila devem conseguir se comunicar entre si. IBM MQ implementa um protocolo de *armazenamento e encaminhamento* para assegurar a entrega segura de mensagens entre esses aplicativos.

Para obter informações adicionais, consulte [“Gerenciadores de filas” na página 29](#).

Grupo de filas compartilhadas



Os gerenciadores de filas que podem acessar o mesmo conjunto de filas compartilhadas formam um grupo denominado *grupo de filas compartilhadas (QSG)*. Eles se comunicam uns com os outros com o recurso de acoplamento (CF) que armazena as filas compartilhadas. Isso é aplicável apenas ao IBM MQ for z/OS.

Para obter informações adicionais, consulte [“Filas compartilhadas e grupos de filas compartilhadas” na página 172](#).

Fila compartilhada



Uma *fila compartilhada* é um tipo de fila local com mensagens que podem ser acessadas por um ou mais gerenciadores de filas que estão em um sysplex. Isso não é igual a uma fila sendo compartilhada por mais de um aplicativo, usando o mesmo gerenciador de filas. Isso é aplicável apenas ao IBM MQ for z/OS.

Assinatura

Um aplicativo de publicação/assinatura pode registrar um interesse nas mensagens sobre tópicos específicos. Quando faz isso, o aplicativo fica conhecido como assinante, e o termo assinatura define como as mensagens correspondentes são enfileiradas para o processamento.

Uma assinatura contém informações sobre a identidade do assinante e a identidade da fila de destino na qual as publicações devem ser colocadas. Ela também contém informações sobre como uma publicação deve ser colocada na fila de destino.

Para obter informações adicionais, consulte [“Assinantes e assinaturas” na página 68](#).

Tópico

Um tópico é uma cadeia de caracteres que descreve o assunto das informações que são publicadas em uma mensagem de publicação/assinatura.

Os tópicos são a chave para a entrega de mensagens bem-sucedida em um sistema de Publicação/Assinatura. Em vez de incluir um endereço de destino específico em cada mensagem, um publicador designa um tópico a cada mensagem. O gerenciador de filas corresponde ao tópico com uma lista de assinantes que assinaram esse tópico, e entrega a mensagem para cada um desses assinantes.

Para obter informações adicionais, consulte [“tópicos” na página 70](#).

Mensagens e Filas

As mensagens e as filas são componentes básicos de um sistema de enfileiramento de mensagens.

O que é uma mensagem?

Um *mensagem* é uma sequência de bytes que é significativa para os aplicativos que a usam. As mensagens são usadas para transferir as informações de um programa de aplicativo a outro (ou entre diferentes partes do mesmo aplicativo). Os aplicativos podem estar em execução na mesma plataforma ou em diferentes plataformas.

Uma mensagem do IBM MQ é composta de:

- *Os dados do aplicativo.* O conteúdo e a estrutura dos dados do aplicativo são definidos pelos programas de aplicativo que o usam.
- *Um descritor de mensagens.* O descritor de mensagens identifica a mensagem e contém informações adicionais de controle, como o tipo de mensagem e a prioridade designada à mensagem enviando o aplicativo.

O formato do descritor de mensagens é definido por IBM MQ. Para obter uma descrição completa do descritor de mensagens, consulte [MQMD - Descritor de mensagens](#).

- *Propriedades de mensagem.* Metadados sobre a mensagem. O conteúdo das propriedades de mensagem é definido pelos programas de aplicativo que os utilizam. Para obter informações adicionais, consulte [Propriedades de mensagem](#).

Comprimentos de Mensagens

O comprimento padrão máximo da mensagem é 4 MB, embora seja possível aumentar isso para um comprimento máximo de 100 MB (em que 1 MB é igual a 1 048 576 bytes). Na prática, o comprimento da mensagem pode ser limitado por:

- Comprimento máximo da mensagem definido para a fila de recepção
- Comprimento máximo da mensagem definido para o gerenciador de fila
- Comprimento máximo da mensagem definido pela fila
- Comprimento máximo da mensagem definido pelo envio ou pela recepção do aplicativo
- A quantidade de armazenamento disponível para a mensagem

Pode haver várias mensagens para enviar todas as informações necessárias para um aplicativo.

Como os Aplicativos Envia e Recebem as Mensagens?

Os programas de aplicativo enviam e recebem mensagens usando **chamadas MQI**.

Por exemplo, para colocar uma mensagem em uma fila, um aplicativo:

1. Abre a fila necessária, emitindo uma chamada MQI MQOPEN
2. Emite uma chamada MQI MQPUT para colocar a mensagem na fila

Outro aplicativo pode recuperar a mensagem a partir da mesma fila emitindo uma chamada MQI MQGET

Para obter informações adicionais sobre as chamadas MQI, consulte [Chamadas MQI](#).

O que É uma Fila?

Uma *fila* é uma estrutura de dados usada para armazenar as mensagens.

Cada fila é de propriedade de um *gerenciador de fila*. O gerenciador de filas é responsável por manter as filas que possui e por armazenar todas as mensagens que recebe nas filas apropriadas. As mensagens podem ser colocadas na fila pelos programas aplicativos ou um gerenciador de filas como parte de sua operação normal.

Filas Predefinidas e Filas Dinâmicas

As filas podem ser caracterizadas pela maneira como são criadas:

- As **filas predefinidas** são criadas por um administrador usando comandos MQSC ou PCF apropriados. As filas predefinidas são permanentes; elas existem independentemente dos aplicativos que as usam e sobrevivem aos reinícios do IBM MQ.
- As **filas dinâmicas** são criadas quando um aplicativo emite uma solicitação de MQOPEN especificando o nome de uma *fila modelo*. A fila criada é baseada em uma *definição de fila modelo*, que é denominada fila modelo. É possível criar uma fila modelo usando o comando DEFINE QMODEL do MQSC. Os atributos de uma fila modelo (por exemplo, o número máximo de mensagens que podem ser armazenadas) são herdados por qualquer fila dinâmica que tenha sido criada para ela.

As filas modelos possuem um atributo que especifica se a fila dinâmica deve ser permanente ou temporária. As filas permanentes sobrevivem aos reinício do aplicativo e do gerenciador de filas; as filas temporárias são perdidas no reinício.

Recuperando Mensagens das Filas

Os aplicativos devidamente autorizados podem recuperar as mensagens de uma fila de acordo com os seguintes algoritmos de recuperação:

- Primeiro a entrar, primeiro a sair (FIFO).
- Prioridade da mensagem, conforme definido no descritor de mensagens. As mensagens que possuem a mesma prioridade são recuperadas com base em FIFO.
- Uma solicitação de programa para uma mensagem específica.

A solicitação MQGET do aplicativo determina o método usado.

Objetos do IBM MQ

Gerenciadores de Filas definem as propriedades de objetos do IBM MQ. Os valores dessas propriedades afetam a maneira como o IBM MQ processa esses objetos. É possível criar e gerenciar objetos usando os comandos e interfaces do IBM MQ. A partir de seus aplicativos, você usa a Message Queue Interface (MQI) para controlar objetos. Os objetos são identificados por um IBM MQ *descritor de objeto* (MQOD) quando direcionados de um programa.

Administração de objeto

A administração de objetos inclui as tarefas a seguir:




- Iniciar e parar os gerenciadores de filas.
- Criar objetos, especialmente as filas, para os aplicativos.
- Exibindo ou alterando os atributos de objetos.
- Excluindo objetos.
- Trabalhar com canais para criar caminhos de comunicação para os gerenciadores de fila em outros sistemas (remotos).

- Criar *clusters* dos gerenciadores de fila para simplificar o processo geral de administração e para equilibrar a carga de trabalho.

Com a exceção de filas dinâmicas, objetos devem ser definidos para um gerenciador de filas antes que se possa trabalhar com eles.


Ao utilizar um comando do IBM MQ para transportar uma operação de administração de objeto, o gerenciador de filas verifica se você tem o nível necessário de autoridade para executar a operação. Da mesma forma, quando um aplicativo usa a chamada MQOPEN para abrir um objeto, o gerenciador de filas verifica se o aplicativo possui o nível necessário de autoridade antes que conceda acesso a esse objeto. As verificações são feitas no nome do objeto sendo aberto.


É possível definir e gerenciar objetos usando os seguintes métodos:


- Os comandos PCF descritos em [Referência de formatos de comandos programáveis](#) e [Automatizando tarefas de administração](#)
- Os comandos MQSC descritos em [Os comandos MQSC](#)
-  As operações IBM MQ for z/OS e os painéis de controle, descritos em [Administrando IBM MQ for z/OS](#)
-   O IBM MQ Explorer (somente sistemas Windows e Linux for Intel). Para obter mais informações, veja [Introdução ao MQ Explorer](#).

Você também pode gerenciar objetos usando os seguintes métodos:

- Os comandos de controle, que são digitado a partir de um teclado. Consulte [Administração IBM MQ for Multiplatforms usando comandos de controle](#).
- IBM MQ As chamadas da Interface de Administração (MQAI) em um programa. Consulte [IBM MQ Administration Interface \(MQAI\)](#).

 Para sequências de comandos do IBM MQ no AIX, Linux, and Windows, é possível usar o recurso do MQSC para executar uma série de comandos retidos em um arquivo. Para obter mais informações, consulte [Administrando IBM MQ usando os comandos MQSC](#)

 Para sequências de comandos do IBM MQ for IBM i que você usa regularmente, é possível gravar comandos de CL. Para obter mais informações, veja [Gerenciando o IBM MQ for IBM i usando comandos de CL](#).

 Para sequências de comandos do IBM MQ for z/OS que você usa regularmente, é possível gravar programas de administração que criam mensagens que contêm comandos e que colocam essas mensagens na fila de entrada de comando do sistema. O gerenciador de filas processa as mensagens nesta fila da mesma maneira que processa os comandos inseridos a partir da linha de comandos ou das operações e dos painéis de controle. Esta técnica está descrita em [Gravando programas para administrar o IBM MQ](#), e é demonstrada no aplicativo de amostra do Gerenciador de correio entregue com o IBM MQ for z/OS. Para obter uma descrição desta amostra, consulte [Programas de amostra para o IBM MQ for z/OS](#).

Atributos do Objeto

As propriedades de um objeto são definidas por seus atributos. Algumas pessoas podem especificar e outras podem apenas visualizar.

Por exemplo, o comprimento máximo da mensagem que uma fila pode acomodar é definido por seu atributo **MaxMsgLength**; é possível especificar este atributo ao criar uma fila. O atributo **DefinitionType** especifica como a fila foi criada; é possível exibir apenas este atributo.

No IBM MQ, existem duas maneiras de se referir a um atributo:

- Usando seu nome PCF, por exemplo, **MaxMsgLength**.
- usando seu nome de comando MQSC, por exemplo, MAXMSGL.

Grupos de compartilhamento de filas



Os gerenciadores de filas que podem acessar o mesmo conjunto de filas compartilhadas formam um grupo denominado *grupo de filas compartilhadas* (QSG) e eles se comunicam entre si usando um recurso de acoplamento (CF) que armazena as filas compartilhadas. Observe que um QSG não é estritamente um objeto..

Uma fila compartilhada é um tipo de fila local com mensagens que podem ser acessadas por um ou mais gerenciadores de filas que estão em um grupo de filas compartilhadas. Isso não é igual a uma fila sendo compartilhada por mais de um aplicativo, usando o mesmo gerenciador de filas.

Os grupos de filas compartilhadas têm um nome de até quatro caracteres. O nome deve ser exclusivo em sua rede e deve ser diferente de qualquer nome de gerenciador de filas.

Importante: As filas compartilhadas e os grupos de filas compartilhadas são suportados apenas no IBM MQ for z/OS.

Consulte [“Filas compartilhadas e grupos de filas compartilhadas”](#) na página 172 para obter mais informações.

Objetos Padrão do Sistema

Os *objetos padrão do sistema* são um conjunto de definições de objetos que são criadas automaticamente sempre que um gerenciador de filas é criado. É possível copiar e modificar quaisquer dessas definições de objeto para uso nos aplicativos em sua instalação.

Os nomes de objeto padrão possuem a raiz SYSTEM; por exemplo, a fila local padrão é SYSTEM.DEFAULT.LOCAL.QUEUE e o canal receptor padrão é SYSTEM.DEF.RECEIVER. Não é possível renomear esses objetos; os objetos padrão desses nomes são necessários.

Ao definir um objeto, quaisquer atributos que você não especificar explicitamente serão copiados a partir do objeto padrão apropriado. Por exemplo, se definir uma fila local, esses atributos que você não especificar são extraídos da fila padrão SYSTEM.DEFAULT.LOCAL.QUEUE.

Consulte [Objetos do Sistema e Padrão](#) para obter informações adicionais

Tipos de objeto

Várias das tarefas de administração envolvem a manipulação de vários tipos de IBM MQ *objetos*.

Para obter informações sobre a nomeação de objetos IBM MQ, consulte [“Nomeando os Objetos IBM MQ”](#) na página 37.

Para obter informações sobre os objetos padrão criados em um gerenciador de filas, consulte [“Objetos Padrão do Sistema”](#) na página 16.

Para obter informações sobre os diferentes tipos de objetos IBM MQ , consulte o seguinte:

Objetos das Informações sobre Autenticação

O objeto de informações sobre autenticação fornece as definições necessárias para executar a verificação de revogação de certificado.

O objeto de informações sobre autenticação do gerenciador de filas faz parte do suporte do IBM MQ para Segurança da Camada de Transporte (TLS). Ele fornece as definições necessárias para verificar os certificados revogados. As Autoridades de Certificação revogam os certificados que não podem mais ser confiáveis.

É possível usar o comando MQSC **DEFINE AUTHINFO** para definir um objeto de informações sobre autenticação. Para obter mais informações sobre os atributos de objetos de informação de autenticação, consulte [DEFINE AUTHINFO](#)

É possível usar os comandos de controle do IBM MQ a seguir com um objeto de informações sobre autenticação:

- **setmqaut** (conceder ou revogar autoridade)
- **dspmqa** (exibir autorização de objeto)
- **dmpmqaut** (fazer dump de autorizações)
- **rcrmqobj** (recriar objeto)
- **rcdmqimg** (registrar imagem de mídia)
- **dspmqls** (exibir nomes de arquivos)

Para obter uma visão geral de TLS e o uso dos objetos de informações sobre autenticação, consulte [Conceitos de Segurança da Camada de Transporte \(TLS\)](#) e [Protocolos de segurança TLS no IBM MQ](#).

Canais

Os canais são objetos que fornecem um caminho de comunicação de um gerenciador de filas para outro. Consulte [“Canais” na página 30](#) para obter mais informações.

Objetos de Informações de Comunicação

O IBM MQ Multicast oferece baixa latência, alto fanout e mensagem multicast confiável. Um objeto de informações de comunicação (COMMINFO) é necessário para usar a transmissão Multicast.

Consulte [“IBM MQ Multicast” na página 109](#) para obter mais informações.

Um objeto COMMINFO é um objeto do IBM MQ que contém os atributos associados à transmissão multicast. Para obter informações adicionais sobre estes atributos, consulte [DEFINE COMMINFO](#). Para obter informações adicionais sobre a criação de um objeto COMMINFO, consulte [Introdução ao multicast](#).


Listeners

Os *listeners* são processos que aceitam as solicitações de rede de outros gerenciadores de fila ou aplicativos clientes e iniciam os canais associados.

Os *processos do listener* podem ser iniciados usando o comando de controle **runmqtsr**

Objetos de listener são objetos do IBM MQ que permitem gerenciar o início e a parada dos processos de listener de dentro do escopo de um gerenciador de filas. Ao definir os atributos de um objeto listener, faça o seguinte:

- Configure o processo listener.
- Especifique se o processo listener inicia e para automaticamente quando o gerenciador de filas é iniciado e parado.

Importante:  Os objetos de listener não são suportados no IBM MQ for z/OS. Para obter mais informações sobre como o IBM MQ for z/OS implementa o atendimento, usando o inicializador de canais, consulte [“O inicializador de canais no z/OS” na página 168](#).

Listas de Nomes


Uma *lista de nomes* é um objeto do IBM MQ que contém uma lista de nomes de cluster, nomes de fila ou nomes de objeto de informações de autenticação. Em um cluster, isso pode ser usado para identificar uma lista de clusters para a qual o gerenciador de filas retém os repositórios.

Uma lista de nomes é um objeto do IBM MQ que contém uma lista de outros objetos do IBM MQ. Geralmente, as listas de nomes são usadas por aplicativos como monitores acionadores, em que são usadas para identificar um grupo de filas. A vantagem de usar uma lista de nomes é que ela é mantida independentemente dos aplicativos; ela pode ser atualizada sem parar qualquer um dos aplicativos que a

usam. Além disso, se um aplicativo falhar, a lista de nomes não será afetada e outros aplicativos poderão continuar usando-a.

As listas de nomes também são usadas com clusters do gerenciador de filas para manter uma lista de clusters referidos por mais de um objeto IBM MQ

É possível definir e modificar listas de nomes usando os comandos MQSC [DEFINE NAMELIST](#) e [ALTER NAMELIST](#).

Nota:  Como alternativa, no z/OS, é possível usar as operações IBM MQ for z/OS e os painéis de controle

Os programas podem usar o MQI para localizar quais filas estão incluídas nessas listas de nomes. A organização das listas de nomes é responsabilidade do editor de telas e do administrador do sistema.

Para obter uma lista de atributos de lista de nomes disponíveis para uso, consulte [Atributos para listas de nomes](#),


Definições de Processo

Os objetos de definição de processo permitem que os aplicativos sejam iniciados sem a necessidade da intervenção do operador, definindo os atributos do aplicativo para uso pelo gerenciador de filas.

O objeto de definição de processo define um aplicativo que é iniciado em resposta a um evento acionador em um gerenciador de filas do IBM MQ. Os atributos de definição de processo incluem o ID do aplicativo, o tipo de aplicativo e os dados específicos para o aplicativo. Para obter mais informações, veja *Filas de inicialização* em [“Filas usadas para propósitos específicos por IBM MQ”](#) na página 27.

Para permitir que um aplicativo seja iniciado sem a necessidade de intervenção do operador, conforme descrito em [Iniciando aplicativos IBM MQ usando acionadores](#), os atributos do aplicativo devem ser conhecidos para o gerenciador de filas. Esses atributos são definidos em um *objeto de definição de processo*.

O atributo **ProcessName** é corrigido quando o objeto é criado. No entanto, é possível mudar outros atributos usando os comandos do IBM MQ.

Nota:  Como alternativa, em z/OS é possível usar as operações IBM MQ for z/OS e os painéis de controle.

É possível consultar sobre os valores de todos os atributos usando [MQINQ - Consultar atributos do objeto](#).

Para obter uma lista de atributos de definição de processo disponíveis para uso, consulte [Atributos para definições de processo](#)

Filas

Uma IBM MQ *fila* é um objeto nomeado no qual os aplicativos podem colocar mensagens e por meio do qual os aplicativos podem obter mensagens.

Consulte [“Filas”](#) na página 20 para obter mais informações.

Gerenciadores de filas

IBM MQ O gerenciador de filas fornece serviços de enfileiramento aos aplicativos e gerencia as filas que pertencem a eles.

Consulte [“Gerenciadores de filas”](#) na página 29 para obter mais informações.

Serviços

Os objetos de *serviço* são uma maneira de definir os programas para que sejam executados quando um gerenciador de filas é iniciado ou parado.


Os programas podem ser de um dos tipos a seguir:

Servidores

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 será executado quando um gerenciador de filas especificado for iniciado. Apenas uma instância de um processo do servidor pode ser executada simultaneamente. Durante a execução, o status de um processo do servidor pode ser monitorado usando o comando MQSC, DISPLAY SVSTATUS. Geralmente os objetos de serviço do servidor são definições de programas como manipuladores de devolução ou monitores acionadores; no entanto, os programas que podem ser executados não são limitados aos fornecidos com IBM MQ. Além disso, um objeto de serviço do servidor pode ser definido para incluir um comando que será executado quando o gerenciador de filas especificado for encerrado para finalizar o programa.

Comandos

Um *comando* é um objeto de serviço que possui o parâmetro SERVTYPE especificado como COMMAND. Um objeto de serviço do comando é a definição de um programa que será executado quando um gerenciador de filas especificado for iniciado ou parado. Diversas instâncias de um processo de comando podem ser executadas simultaneamente. Os objetos de serviço de comando são diferentes dos objetos de serviço do servidor porque depois que o programa é executado o gerenciador de filas não monitorará o programa. Geralmente, os objetos de serviço de comando são definições de programas que são de curta duração e executarão uma tarefa específica como iniciar uma ou mais tarefas diferentes.

Importante:  Os objetos de serviço não são suportados no IBM MQ for z/OS.

Para obter mais informações, consulte [Trabalhando com serviços](#).

Classes de Armazenamento



Uma classe de armazenamento mapeia uma ou mais filas para um conjunto de páginas.

Isso significa que as mensagens para essa fila são armazenadas (sujeito ao buffer) nesse conjunto de páginas.

As classes de armazenamento são suportadas somente no IBM MQ for z/OS.

Para obter mais informações sobre as classes de armazenamento, consulte [Planejando seu ambiente do IBM MQ no z/OS](#)

Objetos Topic

Um *objeto do tópico* é um objeto do IBM MQ que permite que você a designar atributos específicos não padrão aos tópicos.

Um *tópico* é definido por um aplicativo publicando ou assinando uma *sequência de tópicos* específica. Uma sequência de tópicos pode especificar uma hierarquia de tópicos, separando-os com um caractere de barra (/). Isso pode ser visualizado por uma *árvore de tópicos*. Por exemplo, se um aplicativo publica nas sequências de tópicos /Sport/American Football e /Sport/Soccer, uma árvore de tópicos é criada que possui um nó pai Sport com dois filhos, American Football e Soccer.

Os tópicos herdam seus atributos do primeiro nó administrativo pai localizado em suas árvores de tópicos. Se não houver nenhum nó de tópico administrativo em uma determinada árvore de tópicos, todos os tópicos herdarão seus atributos do objeto do tópico base, SYSTEM.BASE.TOPIC.

É possível criar um objeto do tópico em qualquer nó em uma árvore de tópicos especificando a sequência de tópicos do nó no atributo TOPICSTR do objeto de tópico. Também é possível definir outros atributos para o nó de tópico administrativo. Para obter mais informações sobre estes atributos, veja [Os comandos MQSC](#) ou [Automatizando a administração usando comandos PCF](#). Cada objeto do tópico, por padrão, herda seus atributos de seu nó de tópico administrativo pai mais próximo.

Os objetos do tópico também podem ser usados para ocultar a árvore de tópicos integral dos desenvolvedores de aplicativos. Se um objeto de tópico denominado FOOTBALL .US for criado para o tópico /Sport/American Football, um aplicativo poderá publicar ou assinar o objeto denominado FOOTBALL .US em vez da cadeia /Sport/American Football com o mesmo resultado.

Se você inserir um caractere #, +, / ou * dentro de uma sequência de tópicos em um objeto do tópico, o caractere é tratado como um caractere normal na sequência e será considerado parte da sequência de tópicos associada a um objeto de tópico.

Para obter mais informações sobre objetos do tópico, consulte [“Sistema de Mensagens de Publicação/ Assinatura” na página 64.](#)

Conceitos relacionados

[“Introdução ao Enfileiramento de Mensagens” na página 5](#)

Os produtos IBM MQ permitem que os programas se comuniquem entre si em uma rede de componentes diferentes (processadores, sistemas operacionais, subsistemas e protocolos de comunicação) usando uma interface de programação de aplicativos consistente.

Referências relacionadas

[Os Comandos MQSC](#)

Filas

Introdução às filas e atributos de filas do IBM MQ.

As mensagens são armazenadas em uma fila, para que se o aplicativo de colocação estiver esperando uma resposta para sua mensagem, ele fique livre para executar outro trabalho enquanto aguarda pela resposta. Os aplicativos acessam uma fila usando a Message Queue Interface (MQI), descrita em [Visão geral do Message Queue Interface.](#)

Antes que uma mensagem ser colocada em uma fila, ela já deve ter sido criada. Uma fila é de propriedade de um gerenciador de filas e esse gerenciador de filas pode ter várias filas. No entanto, cada fila deve ter um nome que seja exclusivo nesse gerenciador de filas.

Uma fila é mantida por meio de um gerenciador de filas. Na maioria dos casos, cada fila é fisicamente gerenciada por seu gerenciador de filas, mas isso não é aparente para um programa de aplicativo. As filas compartilhadas do IBM MQ for z/OS podem ser gerenciadas por qualquer gerenciador de filas no grupo de filas compartilhadas.

Para criar uma fila, você pode utilizar os comandos do IBM MQ (MQSC), comandos PCF ou interfaces específicas de plataforma. Por exemplo, os painéis de operações e de controle do IBM MQ for z/OS são específicos da plataforma.

É possível criar filas locais para tarefas temporárias *dinamicamente* a partir do seu aplicativo. Por exemplo, é possível criar filas de *resposta* (que não são necessárias após o término de um aplicativo). Para obter mais informações, consulte [“Filas Dinâmicas e de Modelo” na página 25.](#)

Antes de usar uma fila, você deve abri-la, especificando o que deseja fazer com ela. Por exemplo, é possível abrir uma fila para:

- Procurar apenas as mensagens (sem recuperá-las)
- Recuperar as mensagens (e compartilhando o acesso com outros programas ou com acesso exclusivo)
- Colocar as mensagens na fila
- Pergunta sobre os atributos da fila
- Configurar os atributos da fila

Para obter uma lista completa das opções que podem ser especificadas quando você abre uma fila, consulte [MQOPEN - Objeto de Abertura.](#)

Atributos das Filas

Alguns dos atributos de uma fila são especificados quando a fila é definida e não podem ser mudados mais tarde (por exemplo, o tipo da fila). Outros atributos de filas podem ser agrupados naqueles que podem ser mudados:

- Pelo gerenciador de filas durante o processamento da fila (por exemplo, a profundidade atual de uma fila)
- Apenas por comandos (por exemplo, a descrição de texto da fila)
- Por aplicativos, usando a chamada MQSET (por exemplo, se as operações de entrada forem permitidas na fila)

É possível localizar os valores de todos os atributos usando a chamada MQINQ.

Os atributos que são comuns a mais de um tipo de fila são:

QName

Nome da fila.

QType

Tipo da fila.

QDesc

A descrição de texto da fila.

InhibitGet

Se os programas têm permissão para obter mensagens da fila. No entanto, nunca é possível obter mensagens de filas remotas.

InhibitPut

Se os programas têm permissão para colocar mensagens na fila.

DefPriority

Prioridade padrão para as mensagens colocadas na fila;

DefPersistence

Persistência padrão para as mensagens colocadas na fila

Escopo

Controla se uma entrada para esta fila também existe em um serviço de nomes.

 O atributo **Scope** não é suportado no z/OS

Para obter uma descrição completa destes atributos, consulte [Atributos para Filas](#).

Maneiras de definir filas

É possível definir filas para o IBM MQ usando o comando MQSC `DEFINE` ou o comando PCF Criar fila. Os comandos especificam o tipo de fila e seus atributos. Por exemplo, um objeto de fila local possui atributos que especificam o que acontece quando os aplicativos fazem referência a essa fila nas chamadas MQI. Os exemplos de atributos são:

- Indica se os aplicativos podem recuperar as mensagens da fila (GET ativado)
- Indica se os aplicativos podem colocar as mensagens na fila (PUT ativado)
- Indica se o acesso à fila é exclusivo a um aplicativo ou compartilhado entre os aplicativos
- O número máximo de mensagens que podem ser armazenadas na fila ao mesmo tempo (profundidade máxima da fila)
- O comprimento máximo de mensagens que podem ser colocadas na fila

Há também várias interfaces específicas de plataforma que é possível usar para definir filas.

Conceitos relacionados

“Filas de Clusters” na página 60

Uma fila de clusters é uma fila que é hospedada por um gerenciador de filas do cluster e disponibilizada para outros gerenciadores de filas no cluster.

“Filas de Devoluções” na página 51

A fila de devoluções (ou fila de mensagens não entregues) é a fila para a qual as mensagens são enviadas se não puderem ser roteadas para seus destinos corretos. Cada gerenciador de filas geralmente possui uma fila de devoluções.


[Automatizando a administração usando comandos PCF](#)

[Trabalhando com filas no IBM MQ Console](#)

Tarefas relacionadas

[Administrando o IBM MQ usando comandos MQSC](#)

[Criando e configurando gerenciadores de filas e objetos com o MQ Explorer](#)

 [Gerenciando IBM MQ for IBM i usando os comandos de CL](#)

 [Origens das quais é possível emitir comandos MQSC e PCF no IBM MQ for z/OS](#)

Referências relacionadas

“Comparação entre Filas Compartilhadas e Filas de Clusters” na página 60

Estas informações são projetadas para ajudá-lo a comparar filas compartilhadas e filas de clusters e decidir o que pode ser mais adequado para seu sistema.

Informações relacionadas

“O que é uma fila compartilhada?” na página 172

Filas locais

Filas de transmissão, inicialização, devoluções, comandos, padrões, canais e eventos são tipos de filas locais.

Uma fila é conhecida em um programa como *local* se for de propriedade do gerenciador de filas ao qual o programa está conectado. É possível obter mensagens de filas locais e colocar as mensagens nelas.

O objeto de definição de fila contém as informações de definição da fila, bem como as mensagens físicas colocadas na fila.

Cada gerenciador de filas pode ter algumas filas locais que usa para propósitos especiais:

Filas de transmissão

Quando um aplicativo envia uma mensagem para uma fila remota, o gerenciador de filas locais armazena a mensagem em uma fila local especial, denominada *fila de transmissão*. Os aplicativos podem colocar mensagens diretamente em uma fila de transmissão ou indiretamente por meio de uma definição de fila remota.


Quando um gerenciador de filas envia mensagens para um gerenciador de filas remotas, ele identifica a fila de transmissão utilizando a seguinte sequência:

1. A fila de transmissão nomeada no atributo XMITQ da definição local de uma fila remota.
2. Uma fila de transmissão com o mesmo nome que o gerenciador de filas remotas. Esse valor é o valor padrão no XMITQ da definição local de uma fila remota.
3. A fila de transmissão nomeada no atributo DEFXMITQ do gerenciador de fila local.

Um *agente do canal de mensagens* é um programa do canal associado à fila de transmissão e ele entrega a mensagem ao seu próximo destino. O próximo destino é o gerenciador de filas ao qual o canal de mensagens está conectado. Ele não é necessariamente o mesmo gerenciador de filas que o destino final da mensagem. Quando a mensagem é entregue para seu próximo destino, ela é excluída da fila de transmissão. A mensagem pode precisar passar por muitos gerenciadores de filas em sua jornada até seu destino final. Você deve definir uma fila de transmissão em cada gerenciador de filas ao longo da rota, cada uma contendo mensagens aguardando para serem transmitidas para o próximo destino. Uma fila de transmissão normal contém mensagens para o próximo destino, embora as mensagens possam ter diferentes destinos eventuais. Uma fila de transmissão do cluster contém mensagens para diversos destinos. O *correIID* de cada mensagem identifica o canal no qual a mensagem é colocada para transferi-la para seu próximo destino.

É possível definir várias filas de transmissão em um gerenciador de filas. Você pode definir várias filas de transmissão para o mesmo destino, com cada um sendo usado para uma classe de serviço diferente. Por exemplo, talvez você deseje criar filas de transmissão diferentes para mensagens pequenas e mensagens grandes indo para o mesmo destino. É possível, então, transferir as mensagens usando diferentes canais de mensagens, para que as mensagens grandes não atrapalhem as mensagens menores. Todas as mensagens nas filas de clusters ou em tópicos de cluster são colocadas em uma única fila de transmissão do cluster SYSTEM . CLUSTER . TRANSMIT . QUEUE, por padrão. Como uma opção, é possível alterar o padrão e separar o tráfego de mensagens que vai para diferentes gerenciadores de filas de clusters para diferentes filas de transmissão do cluster. Se você configurar o atributo do gerenciador de filas DEFCLXQ para CHANNEL, cada canal do emissor de clusters criará uma fila de transmissão do cluster separada. Como uma alternativa, é possível definir manualmente as filas de transmissão do cluster para serem usadas por canais do emissor de clusters.

As filas de transmissão podem acionar um agente do canal de mensagens para enviar mensagens para a frente; consulte [Iniciando aplicativos IBM MQ usando acionadores](#).

 No IBM MQ for z/OS, se você estiver usando enfileiramento dentro do grupo, a fila de transmissão será atendida por um *agente de enfileiramento dentro do grupo*. Uma fila de transmissão compartilhada é usada ao usar o enfileiramento entre grupos no IBM MQ for z/OS.

Filas de inicialização

Uma *fila de inicialização* é uma fila local na qual o gerenciador de filas coloca uma mensagem do acionador quando um evento acionador ocorre em uma fila do aplicativo.

Um evento acionador é um evento destinado a fazer com que um programa inicie o processamento de uma fila. Por exemplo, um evento pode ter mais de 10 mensagens chegando. Para obter mais informações sobre como o acionamento funciona, consulte [Iniciando aplicativos IBM MQ usando acionadores](#).

Fila de Devoluções (mensagem não entregue)


Uma *fila de devoluções (mensagem não entregue)* é uma fila local na qual o gerenciador de filas coloca as mensagens que não pode entregar.

Quando o gerenciador de filas coloca uma mensagem na fila de devoluções, inclui um cabeçalho na mensagem. As informações do cabeçalho incluem a razão pela qual o gerenciador de filas coloca a mensagem na fila de devoluções. Elas também contêm o destino da mensagem original, a data e o horário em que o gerenciador de filas colocou a mensagem na fila de devoluções.

Os aplicativos também podem usar a fila para as mensagens que não podem ser entregues. Para obter informações adicionais, consulte [Usando a fila de devolução \(mensagem não entregue\)](#).

Fila de Comando do Sistema

A *fila de comando do sistema* é uma fila para a qual os aplicativos devidamente autorizados podem enviar os comandos do IBM MQ . Essas filas recebem os comandos PCF, MQSC e CL, conforme suportado em sua plataforma, na prontidão para o gerenciador de filas para agir neles.

 Em IBM MQ for z/OS a fila é chamada SYSTEM . COMMAND . INPUT ; em outras plataformas é chamado de SYSTEM . ADMIN . COMMAND . QUEUE. Os comandos aceitos variam por plataforma. Consulte [Referência de formatos de comandos programáveis](#) para obter detalhes.

Filas Padrão do Sistema

As *filas padrão do sistema* contêm as definições iniciais das filas para o seu sistema. Quando você cria uma definição de fila, o gerenciador de filas copia a definição da fila padrão do sistema apropriada. A criação de uma definição de fila é diferente de criar uma fila dinâmica. A definição da fila dinâmica é baseada na fila modelo escolhida como o modelo para a fila dinâmica.

Filas de Eventos

As *filas de eventos* retêm as mensagens do evento. Essas mensagens são relatadas pelo gerenciador de filas ou por um canal.

Filas Remotas

Para um programa, uma fila é *remota* caso seja de propriedade de um gerenciador de filas diferente daquele ao qual o programa está conectado.

Quando um link de comunicação tiver sido estabelecido, um programa pode enviar uma mensagem a uma fila remota. Um programa nunca pode receber uma mensagem de uma fila remota.

O objeto de definição de fila, criado quando você define uma fila remota, retém apenas as informações necessárias para que o gerenciador de filas locais localize a fila à qual deseja que a sua mensagem acesse. Este objeto é conhecido como a *definição local de uma fila remota*. Todos os atributos da fila remota são retidos pelo gerenciador de filas que o possui, porque é uma fila local para esse gerenciador de filas.

Ao abrir uma fila remota, para identificar a fila, deve-se especificar um dos seguintes:

- O nome da definição local que define a fila remota. Do ponto de vista de um aplicativo, isso é igual a abri uma fila local. Um aplicativo não precisa saber se uma fila é local ou remota.

Para criar uma definição local de uma fila remota em todas as plataformas exceto IBM i, use o comando `DEFINE QREMOTE`.

 No IBM i, use o comando `CRTMQMQ`.

- O nome do gerenciador de filas remotas e o nome da fila como é conhecido nesse gerenciador de filas remotas.

As definições locais das filas remotas possuem três atributos além dos atributos comuns descritos em “Atributos das Filas” na página 20. Esses três atributos são:

RemoteQName

O nome pelo qual o gerenciador de filas proprietário da fila a conhece.

RemoteQMgrName

O nome do gerenciador de filas proprietário.

XmitQName

O nome da fila de transmissão local que é usada ao encaminhar mensagens para outros gerenciadores de filas.

Para obter mais informações sobre esses atributos, veja [Atributos para filas](#).


Se você usar a chamada MQINQ na definição local de uma fila remota, o gerenciador de filas retornará os atributos da definição local apenas, que é o nome da fila remota, o nome do gerenciador de filas remotas e o nome da fila de transmissão, não os atributos da fila local correspondente no sistema remoto.

Consulte também [Filas de Transmissão](#).

Filas de Alias

Uma *fila de alias* é um objeto do IBM MQ que você pode usar para acessar outra fila ou um tópico. Isso significa que mais de um programa pode funcionar com a mesma fila, acessando-a usando diferentes nomes.

A fila que resulta da resolução de um nome de alias, conhecido como a fila base, pode ser qualquer um dos seguintes tipos de filas, conforme suportado pela plataforma:

- Uma fila local
- A definição local de uma fila remota.
-  Uma fila compartilhada, que é um tipo de fila local disponível somente no IBM MQ for z/OS.
- Uma fila predefinida

- Uma fila dinâmica

Um nome alternativo também pode ser resolvido em um tópico. Se um aplicativo colocar atualmente as mensagens em uma fila, ele pode estar preparado para publicar em um tópico, tornando o nome da fila um alias para o tópico. Não é necessário nenhum código do aplicativo.

Nota: Um alias não pode resolver para outro alias diretamente no mesmo gerenciador de filas.

Um exemplo do uso das filas de alias é para que um administrador do sistema forneça diferentes autoridades de acesso ao nome da fila base (ou seja, a fila na qual o alias é resolvido) e ao nome da fila de alias. Isso significa que um programa ou usuário pode estar autorizado a usar a fila de alias, mas não a fila base.

Como alternativa, a autorização pode ser configurada para inibir as operações de entrada para o nome alternativo, mas permiti-las para a fila base.

Em alguns aplicativos, o uso de filas de alias significa que os administradores do sistema podem alterar facilmente a definição de um objeto da fila de alias sem precisar ter o aplicativo mudado.

IBM MQ faz verificações de autorização no nome alternativo quando os programas tentam usar esse nome. Ele não verifica se o programa está autorizado a acessar o nome para o qual o alias é resolvido. Portanto, um programa pode estar autorizado a acessar um nome de fila de alias, mas não o nome de fila resolvido.

Além dos atributos da fila gerais descritos em [“Filas” na página 20](#), as filas de alias possuem um atributo **BaseQName**. Este é o nome da fila base para a qual o nome alternativo é resolvido. Para obter uma descrição mais completa deste atributo, consulte [BaseQName \(MQCHAR48\)](#).

Os atributos *InhibitGet* e **InhibitPut** (consulte [“Filas” na página 20](#)) das filas de alias pertencem ao nome alternativo. Por exemplo, se o nome de fila de alias ALIAS1 for resolvido para o nome de fila base BASE, as inibições no ALIAS1 afetam o ALIAS1 apenas e o BASE não é inibido. No entanto, as inibições no BASE também afetam ALIAS1.

Os atributos *DefPriority* e **DefPersistence** também pertencem ao nome alternativo. Portanto, por exemplo, você pode designar diferentes prioridades padrão a diferentes alias da mesma fila base. Além disso, é possível alterar essas prioridades sem ter que alterar os aplicativos que usam os alias.


Filas Dinâmicas e de Modelo

Essas informações fornecem um insight em filas dinâmicas, propriedades de filas dinâmicas permanentes e temporárias, usos de filas dinâmicas, algumas considerações ao uso de filas dinâmicas e filas de modelo.

Quando um programa de aplicativo emite uma chamada MQOPEN para abrir uma fila de modelo, o gerenciador de filas cria dinamicamente uma instância de uma fila local com os mesmos atributos que a fila de modelo. Dependendo do valor do campo *DefinitionType* da fila modelo, o gerenciador de filas cria uma fila dinâmica permanente ou temporária (consulte [Criando filas dinâmicas](#)).

Propriedades das Fila Dinâmicas Temporárias

As *filas dinâmicas temporárias* possuem as seguintes propriedades:

-  Elas não podem ser filas compartilhadas, acessíveis por meio de gerenciadores de filas em um grupo de filas compartilhadas.

Observe que os grupos de filas compartilhadas estão disponíveis apenas no IBM MQ for z/OS.

- Elas retêm apenas as mensagens não persistentes.
- Elas são irrecuperáveis.
- Elas são excluídas quando o gerenciador de filas é iniciado.
- Elas são excluídas quando o aplicativo que emitiu a chamada MQOPEN que criou a fila fecha a fila ou termina.
 - Se houver alguma mensagem confirmada na fila, ela será excluída.

- Se houver alguma chamada MQGET, MQPUT ou MQPUT1 não confirmada pendente na fila neste momento, a fila será marcada como sendo logicamente excluída e será excluída apenas fisicamente (depois que essas chamadas forem confirmadas) como parte do processo de fechamento ou quando o aplicativo for terminado.
- Se a fila estiver em uso neste momento (pela criação ou outro aplicativo), a fila será marcada como sendo logicamente excluída e será excluída fisicamente apenas quando fechada pelo último aplicativo que usa a fila.
- As tentativas de acessar uma fila logicamente excluída (em vez de fechá-la) falham com o código de razão MQRC_Q_DELETED.
- MQCO_NONE, MQCO_DELETE e MQCO_DELETE_PURGE são todos tratados como MQCO_NONE quando especificado em uma chamada MQCLOSE para a chamada MQOPEN correspondente que criou a fila.

Propriedades de Filas Dinâmicas Permanente

As *filas dinâmicas permanente* possuem as seguintes propriedades:

- Elas retém mensagens persistentes ou não persistentes.
- Elas são recuperáveis no caso de falhas do sistema.
- Elas são excluídas quando um aplicativo (não necessariamente aquele que emitiu a chamada MQOPEN que criou a fila) fechar com êxito a fila usando a opção MQCO_DELETE ou MQCO_DELETE_PURGE.
 - Uma solicitação de fechamento com a opção MQCO_DELETE falhará se houver alguma mensagem (confirmada ou não confirmada) ainda na fila. Uma solicitação de fechamento com a opção MQCO_DELETE_PURGE é bem-sucedida mesmo se houver mensagens confirmadas na fila (as mensagens sendo excluídas como parte do fechamento), mas falha se houver alguma chamada MQGET, MQPUT ou MQPUT1 não confirmada pendente na fila.
 - Se a solicitação de exclusão for bem-sucedida, mas a fila tiver que ficar em uso (pela criação ou outro aplicativo), a fila será marcada como sendo logicamente excluída e será excluída fisicamente apenas quando fechada pelo último aplicativo que usa a fila.
- Elas não são excluídas se fechadas por um aplicativo que não está autorizado a excluir a fila, a menos que o aplicativo de fechamento tenha emitido a chamada MQOPEN que criou a fila. As verificações de autorização são executadas no identificador de usuários (ou identificador de usuários alternativo se MQOO_ALTERNATE_USER_AUTHORITY tiver sido especificado) que foi usado para validar a chamada MQOPEN correspondente.
- Elas podem ser excluídas da mesma maneira que uma fila normal.

Usos das filas dinâmicas

É possível usar as filas dinâmicas para:

- Aplicativos que não requerem que as filas sejam retidas depois que o aplicativo tiver sido terminado.
- Aplicativos que requerem que as respostas para as mensagens sejam processadas por outro aplicativo. Esses aplicativos podem criar dinamicamente uma fila de resposta abrindo uma fila modelo. Por exemplo, um aplicativo cliente pode:
 1. Criar uma fila dinâmica.
 2. Forneça seu nome no campo **ReplyToQ** da estrutura do descritor de mensagens da mensagem de solicitação.
 3. Coloque a solicitação em uma fila sendo processada por um servidor.

O servidor então pode colocar a mensagem de resposta na fila de resposta. Finalmente, o cliente poderia processar a resposta e fechar a fila de resposta com a opção de exclusão.

Considerações ao usar as filas dinâmicas

Considere os seguintes pontos ao usar as filas dinâmicas:

- Em um modelo cliente-servidor, cada cliente deve criar e usar sua própria fila de resposta dinâmica. Se uma fila de resposta dinâmica for compartilhada entre mais de um cliente, a exclusão da fila de resposta pode ser atrasada porque existe uma atividade não confirmada pendente na fila ou porque a fila está em uso por outro cliente. Além disso, a fila pode ser marcada como sendo logicamente excluída e inacessível para as solicitações de API subsequente (que não seja MQCLOSE).
- Se o seu ambiente de aplicativos precisar que as filas dinâmicas sejam compartilhadas entre os aplicativos, certifique-se de que a fila seja apenas fechada (com a opção de exclusão) quando toda a atividade na fila tiver sido confirmada. Isso deve acontecer com o último usuário. Isso assegura que a exclusão da fila não seja atrasada e minimiza o período em que a fila fica inacessível porque foi marcada como sendo logicamente excluída.

Filas Modelo

Uma *fila modelo* é um modelo de uma definição de fila que você usa ao criar uma fila dinâmica.

É possível criar uma fila local dinamicamente a partir de um IBM MQ, nomeando a fila modelo que deseja usar como modelo para os atributos de fila. Nesse ponto, é possível alterar alguns atributos da nova fila. No entanto, não é possível alterar o **DefinitionType**. Se, por exemplo, você precisar de uma fila permanente, selecione uma fila modelo com o tipo de definição configurado como permanente. Alguns aplicativos de conversação podem usar as filas dinâmicas para reter suas consultas porque provavelmente não precisam manter essas filas depois que tiverem processado as respostas.

Especifique o nome de uma fila modelo no *descriptor de objeto* (MQOD) de sua chamada MQOPEN. Usando os atributos da fila modelo, o gerenciador de filas cria dinamicamente uma fila local para você.

É possível especificar um nome (por inteiro) para a fila dinâmica ou raiz de um nome (por exemplo, ABC) e deixar que o gerenciador de filas inclua uma parte exclusiva nisso ou é possível permitir que o gerenciador de filas designe um nome exclusivo completo. Se o gerenciador de filas designar o nome, ele o colocará na estrutura MQOD.

Não é possível emitir uma chamada MQPUT1 diretamente para uma fila de modelo, mas é possível emitir um MQPUT1 para uma fila dinâmica que foi criada, abrindo uma fila modelo.

MQSET e MQINQ não podem ser emitidos em relação a uma fila modelo. Abrir uma fila modelo com MQOO_INQUIRE ou MQOO_SET resulta em chamadas MQINQ e MQSET subsequentes que estão sendo feitas com relação à fila criada dinamicamente.

Os atributos de uma fila modelo são um subconjunto daqueles de uma fila local. Para obter uma descrição integral, consulte [Atributos para Filas](#).

Filas usadas para propósitos específicos por IBM MQ

O IBM MQ usa algumas filas locais para propósitos específicos relacionados à sua operação.

Você deve definir estas filas antes de o IBM MQ poder usá-las.

Filas de inicialização

As filas de inicialização são aquelas usadas no acionamento. Um gerenciador de filas coloca uma mensagem do acionador em uma fila de inicialização quando ocorrer um evento acionador. Um evento acionador é uma combinação lógica das condições detectadas por um gerenciador de filas. Por exemplo, um evento acionador pode ser gerado quando o número de mensagens em uma fila atingir uma profundidade predefinida. Este evento faz com que o gerenciador de filas coloque uma mensagem do acionador em uma fila de inicialização especificada. Esta mensagem do acionador é recuperado por um *monitor acionador*, um aplicativo especial que monitora uma fila de inicialização. O monitor acionador então inicia o programa de aplicativo que foi especificado na mensagem do acionador.

Se um gerenciador de filas tiver que usar o acionamento, pelo menos, uma fila de inicialização deve ser definida para esse gerenciador de filas. Consulte [Gerenciando objetos para acionamento](#), [runmqtrm](#) e [Iniciando aplicativos IBM MQ usando acionadores](#)

Filas de transmissão

As filas de transmissão são filas que armazenam temporariamente as mensagens que são destinadas para um gerenciador de filas remotas. Você deve definir pelo menos uma fila de transmissão para cada gerenciador de filas remotas para a qual o gerenciador de filas locais deve enviar as mensagens diretamente. Essas filas também são usadas na administração remota; consulte [Administração remota a partir de um gerenciador de filas local](#). Para obter informações sobre o uso de filas de transmissão no enfileiramento distribuído, consulte [Técnicas de enfileiramento distribuído do IBM MQ](#).

Cada gerenciador de filas pode ter uma fila de transmissão padrão. Se um gerenciador de filas que não faz parte de um cluster colocar uma mensagem em uma fila remota, a ação padrão é usar a fila de transmissão padrão. Se houver uma fila de transmissão com o mesmo nome que o gerenciador de filas de destino, a mensagem será colocada nessa fila de transmissão. Se houver uma definição de alias do gerenciador de filas, na qual o parâmetro **RQMNAME** corresponde ao gerenciador de filas de destino, e o parâmetro **XMITQ** é especificado, a mensagem é colocada na fila de transmissão nomeada por **XMITQ**. Se não houver o parâmetro **XMITQ**, a mensagem será colocada na fila local nomeada na mensagem.

Filas de Transmissão de Cluster

Cada gerenciador de filas dentro de um cluster possui uma fila de transmissão do cluster chamada `SYSTEM.CLUSTER.TRANSMIT.QUEUE` e uma fila de transmissão do cluster de modelo, `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. As definições destas filas são criadas por padrão quando você define um gerenciador de filas. Se o atributo do gerenciador de filas, **DEFCLXQ**, é configurado como `CHANNEL`, uma fila de transmissão de cluster dinâmico permanente é criada automaticamente para cada canal do emissor de clusters que é criado. As filas são chamadas de `SYSTEM.CLUSTER.TRANSMIT.ChannelName`. Também é possível definir filas de transmissão do cluster manualmente.

Um gerenciador de filas que faz parte do cluster envia mensagens em uma destas filas para outros gerenciadores de filas que estão no mesmo cluster.

Durante a resolução do nome, uma fila de transmissão do cluster tem precedência sobre a fila de transmissão padrão e uma fila de transmissão do cluster específica tem precedência sobre `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Filas de Devoluções

Uma fila de devoluções (mensagens não entregues) é uma fila que armazena as mensagens que não podem ser roteadas para seus destinos corretos. Uma mensagem não pode ser roteada quando, por exemplo, a fila de destino está cheia. A fila de devoluções fornecida é chamada de `SYSTEM.DEAD.LETTER.QUEUE`.

Para o enfileiramento distribuído, defina uma fila de devoluções em cada gerenciador de filas envolvido.

Filas de Comando

A fila de comandos, `SYSTEM.ADMIN.COMMAND.QUEUE`, é uma fila local para a qual aplicativos autorizados de conformidade podem enviar comandos do MQSC para processamento. Esses comandos são então recuperados por um componente do IBM MQ denominado servidor de comandos. O servidor de comandos valida os comandos, encaminha os válidos para processamento pelo gerenciador de filas e retorna qualquer resposta para a fila de resposta apropriada.

Uma fila de comandos é criada automaticamente para cada gerenciador de filas quando este for criado.

Filas de Resposta

Quando um aplicativo envia uma mensagem de solicitação, o aplicativo que recebe a mensagem pode enviar de volta uma mensagem de resposta para o aplicativo de envio. Esta mensagem é colocada em uma fila, denominada fila de resposta, que normalmente é uma fila local para o aplicativo de envio. O nome do fila de resposta é especificado pelo aplicativo de envio como parte do descritor de mensagens.

Filas de Eventos

Os eventos de instrumentação podem ser usados para monitorar os gerenciadores de filas independentemente dos aplicativos MQI.

Quando ocorrer um evento de instrumentação, o gerenciador de filas colocará uma mensagem de evento em uma fila de eventos. Esta mensagem pode então ser lida por um aplicativo de monitoramento, que pode informar um administrador ou iniciar alguma ação reparatória se o evento indicar um problema.

Nota: Eventos acionadores são diferentes de eventos de instrumentação. Os eventos acionadores não são causados pelas mesmas condições e não geram mensagens do evento.

Para obter informações adicionais sobre os eventos de instrumentação, consulte [Eventos de instrumentação](#).

Gerenciadores de filas

Uma instrução para os *gerenciadores de filas* e os serviços de enfileiramento que eles fornecem aos aplicativos.

Um programa deve ter uma conexão a um gerenciador de filas antes que possa usar os serviços desse gerenciador de filas. Um programa pode estabelecer esta conexão explicitamente (usando a chamada MQCONN ou MQCONNX) ou a conexão pode ser estabelecida implicitamente (isso depende da plataforma e do ambiente no qual o programa está sendo executado).

Um gerenciador de filas do IBM MQ assegura as ações a seguir:

- Os atributos do objeto são mudados de acordo com os comandos recebidos.
- Eventos especiais como eventos acionadores ou eventos de instrumentação são gerados quando as condições apropriadas são atendidas.
- As mensagens são colocadas na fila correta, conforme solicitado pelo aplicativo que está fazendo a chamada MQPUT. O aplicativo será informado, se isso não puder ser feito e um código de razão apropriado for fornecido.

Cada fila pertence a um único gerenciador de filas e é considerado como uma *fila local* para esse gerenciador de filas. O gerenciador de filas ao qual um aplicativo está conectado é considerado o *gerenciador de filas locais* para esse aplicativo. Para o aplicativo, as filas que pertencem a seu gerenciador de filas locais são filas locais.

Uma *fila remota* é uma fila que pertence a outro gerenciador de filas. Um *gerenciador de filas remotas* é qualquer gerenciador de fila que não seja o gerenciador de filas locais. Um gerenciador de filas remotas pode existir em uma máquina remota na rede ou pode existir na mesma máquina que o gerenciador de filas locais. IBM MQ suporta diversos gerenciadores de filas na mesma máquina.


Um objeto gerenciador de filas pode ser usado em algumas chamadas MQI. Por exemplo, você pode questionar sobre os atributos do objeto gerenciador de filas usando a chamada MQI MQINQ.

Atributos dos Gerenciadores de Filas


Associado a cada gerenciador de filas está um conjunto de atributos (ou propriedades) que definem suas características. Alguns dos atributos de um gerenciador de filas são corrigidos quando é criado; é possível alterar outros usando os comandos do IBM MQ. É possível consultar os valores de todos os atributos, exceto os usados para a criptografia de Segurança da Camada de Transporte (TLS), usando a chamada MQINQ.

Os atributos corrigidos incluem:

- O nome do gerenciador de filas
- A plataforma na qual o gerenciador de filas é executado (por exemplo, Windows)
- O nível dos comandos de controle do sistema que o gerenciador de filas suporta
- A prioridade máxima que você pode designar às mensagens processadas pelo gerenciador de filas

- O nome da fila para o qual os programas podem enviar os comandos do IBM MQ
- O comprimento máximo de mensagens que o gerenciador de filas pode processar  (corrigido apenas no IBM MQ for z/OS)
- Indica se o gerenciador de filas suporta a indicação de sincronização quando os programas colocam e obtêm as mensagens

Os atributos *alteráveis* incluem:

- Uma descrição do texto do gerenciador de filas
- O identificador do conjunto de caracteres que o gerenciador de filas usa para as sequências de caracteres quando processa chamadas MQI
- O intervalo de tempo que o gerenciador de filas usa para restringir o número de mensagens do acionador
-  O intervalo de tempo que o gerenciador de filas usa para determinar com que frequência as filas devem ser varridas para as mensagens expiradas (IBM MQ for z/OS apenas)
- O nome da fila de devolução do gerenciador de filas (mensagem não entregue)
- O nome da fila de transmissão padrão do gerenciador de filas
- O número máximo de manipulações abertas para qualquer conexão
- A ativação e desativação de várias categorias da geração de relatórios do evento
- O número máximo de mensagens não confirmadas em uma unidade de trabalho

Gerenciadores de Filas e Gerenciamento de Carga de Trabalho

É possível configurar um cluster dos gerenciadores de fila que possui mais de uma definição para a mesma fila (por exemplo, os gerenciadores de fila no cluster deve ser clones uns dos outros). As mensagens para uma determinada fila podem ser tratadas por qualquer gerenciador de filas que hospede uma instância da fila. Um algoritmo de gerenciamento de carga de trabalho decide qual gerenciador de filas manipula a mensagem e, portanto, difunde a carga de trabalho entre seus gerenciadores de filas; consulte [O Algoritmo de Gerenciamento de Carga de Trabalho do Cluster](#) para obter informações adicionais.

Canais

Um *canal* é um link de comunicação lógica, usado por gerenciadores de filas distribuídas, entre um IBM MQ MQI client e um servidor IBM MQ ou entre dois servidores IBM MQ.

Os canais são usados para mover as mensagens de um gerenciador de filas para outro e protegem os aplicativos dos protocolos de comunicações subjacentes. Os gerenciadores de filas podem existir no mesmo sistema ou em sistemas diferentes na mesma plataforma ou ainda em plataformas diferentes. As mensagens que são enviadas podem se originar de vários locais:

- Programas de aplicativo gravados pelo usuário que transferem dados de um nó para outro.
- Aplicativos de administração gravados pelo usuário que usam comandos PCF ou a MQAI.
- O IBM MQ Explorer.
- Gerenciadores de filas que enviam mensagens de evento de instrumentação para outro gerenciador de filas.
- Gerenciadores de filas que enviam comandos de administração remota para outro gerenciador de filas. Por exemplo, usando comandos MQSC ou o administrative REST API.

Um canal possui duas definições: um em cada extremidade da conexão. Para que os gerenciadores de filas se comuniquem entre si, você deve definir um objeto do canal no gerenciador de filas que deve enviar as mensagens e outro, completar no gerenciador de filas que deve recebê-las. O mesmo *nome de canal* deve ser usado em cada extremidade da conexão e o *tipo de canal* usado deve ser compatível.

Existem três categorias de canal no IBM MQ, com diferentes tipos de canal dentro dessas categorias:

- Canais de mensagens, que são unidirecionais e transferem mensagens de um gerenciador de filas para outro.
- Canais MQI, que são bidirecionais e transferem chamadas MQI de um IBM MQ MQI client para um gerenciador de filas e respostas de um gerenciador de filas para um cliente IBM MQ.
- Os canais do AMQP, que são bidirecionais e conectam um cliente do AMQP a um gerenciador de filas em uma máquina servidor. O IBM MQ usa canais do AMQP para transferir chamadas e respostas do AMQP entre os aplicativos e gerenciadores de filas do AMQP

Canais de mensagens

O propósito de um canal de mensagens é transferir as mensagens de um gerenciador de filas para outro. Os canais de mensagens não são necessários para o ambiente do servidor do cliente.

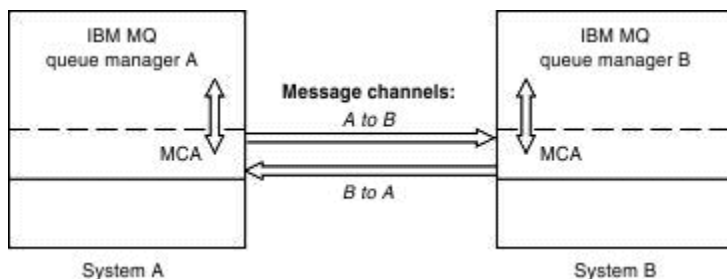


Figura 2. Canais de Mensagens entre Dois Gerenciadores de Filas

Um canal de mensagem é um link unidirecional. Se desejar remover um gerenciador de filas remotas para responder às mensagens enviadas por um gerenciador de filas locais, deve-se configurar um segundo canal para enviar as respostas de volta para o gerenciador de filas locais.

Um canal de mensagens conecta dois gerenciadores de filas usando *agentes do canal de mensagens* (MCAs). Há um agente do canal de mensagens em cada extremidade de um canal. É possível permitir que um MCA transfira mensagens utilizando diversos encadeamentos. Esse processo é conhecido como *enfileiramento*. O enfileirando permite que o MCA transfira mensagens com mais eficiência, aprimorando o desempenho do canal. Para obter mais informações sobre enfileiramento, consulte [Atributos de canais](#).

Para obter mais informações sobre canais, consulte [Chamadas de saída do canal e estruturas de dados e "Componentes de Enfileiramento Distribuído"](#) na página 48.

Canais da MQI

Um canal Message Queue Interface (MQI) conecta um IBM MQ MQI client a um gerenciador de filas em uma máquina servidor e é estabelecido quando você emite uma chamada MQCONN ou MQCONNx de um aplicativo IBM MQ MQI client.

É um link de duas vias e é usado para a transferência de chamadas e respostas de MQI apenas, incluindo chamadas MQPUT que contêm dados da mensagem e chamadas MQGET que resultam no retorno de dados da mensagem. Há maneiras diferentes de criar e usar as definições de canal (consulte [Definindo canais de MQI](#)).

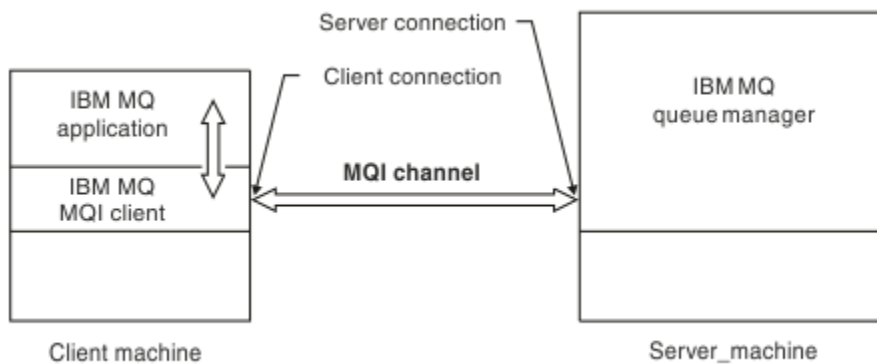


Figura 3. Conexão do Cliente e Conexão do Servidor em um Canal MQI

z/OS Um canal MQI pode ser usado para conectar um cliente a um único gerenciador de filas ou a um gerenciador de filas que faz parte de um grupo de filas compartilhadas (consulte [Conectando um cliente a um grupo de filas compartilhadas](#)).

Existem dois tipos de canais para definições de canal MQI. Eles definem o canal MQI bidirecional.

Canal de conexão do cliente

Esse tipo é para o IBM MQ MQI client.

Canal de Conexão do Servidor

Esse tipo é para o servidor executando o gerenciador de filas, com o qual o IBM MQ aplicativo, em execução em um ambiente do IBM MQ MQI client, deve se comunicar.

Canais AMQP

Multi

Existe somente um tipo de canal AMQP.

É usado um canal para conectar um aplicativo de sistema de mensagens AMQP com um gerenciador de filas, ativando o aplicativo para troca de mensagens com os aplicativos do IBM MQ. Um canal AMQP permite que você desenvolva um aplicativo usando MQ Light e, em seguida, implemente-o como um aplicativo corporativo, tirando vantagem das instalações de nível corporativo fornecidas pelo IBM MQ.

Canais de Conexão do Cliente

Canais de conexão do cliente são objetos que fornecem um caminho de comunicação de um IBM MQ MQI client para um gerenciador de filas.

Os canais de conexão do cliente são usados no enfileiramento distribuído para mover as mensagens entre um gerenciador de filas e um cliente. Eles protegem os aplicativos dos protocolos de comunicação subjacentes. O cliente pode existir em plataformas iguais ou diferentes do gerenciador de filas.

Definições de canal

Consulte “Definições de canal” na página 33 para obter as descrições de cada tipo de canal.

Conceitos relacionados

“Enfileiramento distribuído e clusters” na página 44

Enfileiramento distribuído significa enviar mensagens de um gerenciador de filas para outro. O gerenciador de filas de recebimento pode estar na mesma máquina ou em outra; perto ou do outro lado do mundo. Ele pode estar sendo executado na mesma plataforma que o gerenciador de filas locais ou pode estar em qualquer uma das plataformas suportadas pelo IBM MQ. É possível definir manualmente todas as conexões em um ambiente de enfileiramento distribuído ou você pode criar um cluster e permitir que o IBM MQ defina a maioria dos detalhes da conexão para você.

Tarefas relacionadas

[Administrando objetos remotos do IBM MQ](#)

[Parando canais MQI](#)

[Configurando as Conexões entre o Servidor e o Cliente](#)

Referências relacionadas

[Chamadas de Saída do Canal e Estrutura de Dados](#)

[“Comunicações” na página 36](#)

IBM MQ MQI clients usam canais MQI para se comunicar com o servidor.

Definições de canal

Tabelas que descrevem os diferentes tipos de canais de mensagens e canais de MQI que o IBM MQ usa.

Ao referir-se aos canais de mensagens, a palavra canal é normalmente usada como um sinônimo de definição de canal. Fica claro, pelo contexto, se estamos falando de um canal completo, que tem duas extremidades, ou de uma definição de canal, que tem apenas uma extremidade.

Canais de mensagens

As definições de canal de mensagens podem ser um dos seguintes tipos:

Tipo de definição de canal de mensagens	Descrição
Emissor	Um canal emissor é um canal de mensagens que o gerenciador de filas utiliza para enviar mensagens a outros gerenciadores de filas. Para enviar mensagens utilizando um canal emissor, é necessário criar também, no outro gerenciador de filas, um canal receptor com o mesmo nome do canal emissor. É possível também utilizar canais emissores com canais solicitantes se estiver implementando um mecanismo de "retorno de chamada".
Servidor	Um canal servidor é um canal de mensagens que o gerenciador de filas utiliza para enviar mensagens a outros gerenciadores de filas. Para enviar mensagens utilizando um canal servidor, é necessário criar também, no outro gerenciador de filas, um canal receptor com o mesmo nome do canal servidor. É possível também utilizar os canais do servidor com canais de solicitante. Nesse caso, a definição de canal do solicitante na outra extremidade do canal solicita o início da definição de canal do servidor. O servidor envia mensagens para o solicitante. O servidor também pode iniciar a comunicação assim que o servidor souber o nome da conexão do canal do parceiro.
Receptor	Um canal de receptor é um canal de mensagens que o gerenciador de filas utiliza para receber mensagens de outros gerenciadores de filas. Para receber mensagens utilizando um canal receptor, é necessário criar também, no outro gerenciador de filas, um canal emissor ou servidor com o mesmo nome desse canal receptor.

Tipo de definição de canal de mensagens	Descrição
Solicitante	Um canal do Solicitante é um canal de mensagens que o gerenciador de filas usa para receber mensagens de outros gerenciadores de filas. Um canal do solicitante pode solicitar a inicialização do canal do parceiro que está definido na extremidade remota. Se o canal do parceiro for um canal do servidor, ele aceitará a solicitação de inicialização e começará a enviar mensagens da fila de transmissão identificada na definição do canal do servidor para o canal do solicitante. Se o canal do parceiro for um canal emissor, ele aceitará a solicitação de inicialização. No entanto, ele encerrará a conexão com o solicitante em seguida. Em seguida, o canal emissor é iniciado, negocia uma sessão com o canal do solicitante do parceiro e começa a enviar mensagens da fila de transmissão que é identificada na definição do canal emissor. Este último caso, essencialmente, fornece um mecanismo de retorno de chamada no qual o canal do Solicitante solicita que o canal Emissor retorne a chamada.
Emissor de cluster	Uma definição de canal emissor do cluster (CLUSDR) define a extremidade de envio de um canal no qual um gerenciador de filas do cluster pode enviar informações sobre o cluster para um dos repositórios completos. O canal emissor do cluster é utilizado para notificar o repositório de todas as mudanças no status do gerenciador de filas, por exemplo, a inclusão ou remoção de uma fila. É usado também para transmitir mensagens. Os próprios gerenciadores de filas de repositório completo têm canais do emissor de clusters que apontam um para o outro. Eles os usam para comunicar uns aos outros alterações no status do cluster. É de menor importância para qual repositório completo uma definição de canal CLUSSDR do gerenciador de filas aponta. Depois que o contato inicial foi feito, novos objetos do gerenciador de filas do cluster são definidos automaticamente conforme requerido para que o gerenciador de filas possa enviar informações sobre o cluster a cada repositório completo e mensagens a cada gerenciador de filas.
Receptor de cluster	Uma definição de canal receptor do cluster (CLURCVR) define a extremidade de recebimento de um canal no qual um gerenciador de filas do cluster pode receber mensagens de outros gerenciadores de filas no cluster. Um canal receptor do cluster pode também transportar informações sobre o cluster destinadas ao repositório. Definindo o canal receptor do cluster, o gerenciador de filas indica aos outros gerenciadores de filas do cluster que ele está disponível para receber mensagens. É necessário pelo menos um canal receptor de clusters para cada gerenciador de filas do cluster.

Para cada canal, você deve definir ambas as extremidades para que haja uma definição de canal para cada extremidade do canal. As duas extremidades do canal devem ser de tipos compatíveis.

Você pode ter as seguintes combinações de definições de canal:

- Emissor-Receptor
- Servidor-Receptor
- Solicitante-Servidor
- Solicitante-Servidor (retorno de chamada)
- Cluster-emissor-Cluster-receptor

Agente do canal de mensagens

Cada definição de canal criada pertence a um determinado gerenciador de filas. Um gerenciador de filas pode ter vários canais de tipos iguais ou diferentes. Em cada extremidade do canal, há um programa, o MCA (Message Channel Agent). Em uma extremidade do canal, o MCA responsável pela chamada pega as mensagens da fila de transmissão e as envia por meio do canal. Na outra extremidade do canal, o MCA responsável pelo atendimento recebe as mensagens e as entrega ao gerenciador de filas remotas.

Um MCA responsável pela chamada pode estar associado a um canal de emissor, servidor ou solicitante. Um MCA responsável pelo atendimento pode estar associado a qualquer tipo de canal de mensagens.

O IBM MQ suporta as seguintes combinações de tipo de canal nas duas extremidades de uma conexão:

Responsável pela Chamada		Direcionamento do Fluxo de Mensagens	Respondente	
Tipo de canal	Listener Exigido?		Listener Exigido?	Tipo de canal
Emissor	No	Responsável pela Chamada para Responsável pelo Atendimento	Sim	Receptor
Servidor	No	Responsável pela Chamada para Responsável pelo Atendimento	Sim	Receptor
Servidor	No	Responsável pela Chamada para Responsável pelo Atendimento	Sim	Solicitante
Solicitante	No	Responsável pelo Atendimento para Responsável pela Chamada	Sim	Servidor
Solicitante	Sim	Responsável pelo Atendimento para Responsável pela Chamada	Sim	Emissor

Canais da MQI

Os canais MQI podem ser de um dos seguintes tipos:

Tipo de Canal MQI	Descrição
Conexão do servidor	Um canal de conexão do servidor é um canal MQI bidirecional que é usado para conectar um cliente do IBM MQ a um servidor do IBM MQ. O canal de conexão do servidor é a extremidade de servidor do canal.
Conexão do cliente	Um canal de conexão do cliente é um canal MQI bidirecional que é usado para conectar um cliente do IBM MQ a um servidor do IBM MQ. O IBM MQ Explorer também usa conexões do cliente para se conectar a gerenciadores de filas remotas. O canal de conexão do cliente é o final do cliente do canal. Quando você cria um canal de conexão do cliente, é criado um arquivo no computador que hospeda o gerenciador de filas. Deve-se copiar o arquivo de conexão do cliente no computador cliente do IBM MQ.

Suporte de encadeamento múltiplo-pipelining

Opcionalmente, é possível permitir que um agente do canal de mensagens (MCA) transfira mensagens usando vários encadeamentos. Esse processo, chamado *pipeline*, permite que o MCA transfira mensagens de forma mais eficiente, com menos estados de espera, o que melhora o desempenho do canal. Cada MCA é limitado a um máximo de dois encadeamentos

Você controla o pipeline com o parâmetro *PipeLineLength* no arquivo *qm.ini*. Esse parâmetro é incluído na [Sub-rotina de Canais](#)

Nota: A definição de pipeline é eficaz somente em canais TCP/IP.

Ao usar pipelining, os gerenciadores de filas em ambas as extremidades do canal devem ser configurados para ter um *PipeLineLength* maior que 1.

Considerações de saída do canal

O pipelining pode fazer com que alguns programas de saída falhem, porque:

- As saídas podem não ser chamadas em série.
- As saídas podem ser chamadas alternadamente de diferentes encadeamentos.

Verifique o design de seus programas de saída antes de usar pipelining:

- As saídas devem ser reentrantes em todos os estágios de sua execução
- Ao usar chamadas MQI, lembre-se de que não é possível usar a mesma manipulação MQI quando a saída é chamada de diferentes encadeamentos.

Considere uma saída de mensagens que abre uma fila e usa seu identificador para chamadas MQPUT em todas as chamadas subsequentes da saída. Isso falha no modo de pipelining porque a saída é chamada a partir de diferentes encadeamentos. Para evitar essa falha, mantenha um identificador de fila para cada encadeamento e verifique o identificador de encadeamento toda vez que a saída for chamada




Comunicações

IBM MQ MQI clients usam canais MQI para se comunicar com o servidor.


Uma definição de canal deve ser criada em ambas as extremidades do IBM MQ MQI client e servidor da conexão. Como criar definições de canal é explicado em in [Definindo canais MQI](#).

Os possíveis protocolos de transmissão são mostrados na seguintes tabela:

Tabela 1. Protocolos de Transmissão para Canais MQI

Plataforma do cliente	LU6.2	TCP/IP	NetBIOS	SPX
 IBM i		Sim		
 Sistemas AIX and Linux	Sim ¹	Sim		
 Windows	Sim	Sim	Sim	Sim

Nota:

1.  O LU 6.2 não é suportado nas plataformas a seguir:

- Linux (plataforma POWER)
- Linux (plataforma x86-64)
- Linux (plataforma zSeries s390x)

Protocolos de transmissão – combinação de plataformas de IBM MQ MQI client e do servidor mostra as possíveis combinações de plataformas de IBM MQ MQI client e do servidor, usando esse protocolos de transmissão.

Um aplicativo IBM MQ em um IBM MQ MQI client pode usar todas as chamadas MQI da mesma maneira que quando o gerenciador de filas é local. **MQCONN** ou **MQCONNX** associa o aplicativo IBM MQ com o gerenciador de filas selecionado, criando um *identificador de conexões*. Outras chamadas usando esse identificador de conexões são então processadas pelo gerenciador de filas conectado. A comunicação do IBM MQ MQI client requer uma conexão ativa entre o cliente e o servidor, contrastando com a comunicação entre gerenciadores de filas, que é independente de conexão e de tempo.

O protocolo de transmissão é especificado usando a definição de canal e não afeta o aplicativo. Por exemplo, um aplicativo Windows pode conectar-se a um gerenciador de filas sobre TCP/IP e a outro gerenciador de filas sobre NetBIOS.

Considerações sobre Desempenho

O protocolo de transmissão que você usa pode afetar o desempenho do sistema do cliente e servidor IBM MQ. Em determinadas situações em que a transmissão é lenta, é possível usar a compactação de canal do IBM MQ

Nomeando os Objetos IBM MQ


A convenção de nomenclatura adotada para objetos do IBM MQ depende do objeto. o nome das máquinas e os ID de usuário que você usa com o IBM MQ também estão sujeitos a algumas restrições de nomenclatura.

Cada instância de um gerenciador de filas é conhecida por seu nome. Este nome deve ser exclusivo na rede de gerenciadores de fila interconectados, para que um gerenciador de filas possa identificar precisamente o gerenciador de filas de destino para o qual qualquer mensagem fornecida é enviada.

Para outros tipos de objeto, cada objeto possui um nome associado e pode ser referido por esse nome. Esses nomes devem ser exclusivos em um gerenciador de filas e tipo de objeto. Por exemplo, é possível ter uma fila e um processo com o mesmo nome, mas não ter duas filas com o mesmo nome.

No IBM MQ, os nomes podem ter no máximo 48 caracteres, com exceção dos *canais* que possuem um máximo de 20 caracteres. Para obter informações adicionais sobre a nomeação dos objetos IBM MQ, consulte [“Regras para Nomear os Objetos IBM MQ”](#) na página 38.

O nome das máquinas e os ID de usuário que você usa com o IBM MQ também estão sujeitos a algumas restrições de nomenclatura:

- Certifique-se de que o nome da máquina não contenha nenhum espaço. O IBM MQ não oferece suporte a nomes de máquinas que incluam espaços. Se você instalar o IBM MQ nesse tipo de máquina, não poderá criar nenhum gerenciador de filas.
- Para obter autorizações do IBM MQ, IDs de nomes de usuários e grupos não devem ter mais de 20 caracteres (espaços não são permitidos).
-  Um servidor do IBM MQ for Windows não suporta a conexão de um IBM MQ MQI client se o cliente estiver executando sob um ID de usuário que contém o caractere @, por exemplo, abc@d.

Conceitos relacionados

“Nomes de arquivos do IBM MQ” na página 41

Cada gerenciador de filas, fila, definição de processo, lista de nomes, canal, canal de conexão do cliente, listener, serviço e objeto de informações sobre autenticação do IBM MQ é representado por um arquivo. Como os nomes de objeto não são necessariamente nomes de arquivo válidos, o gerenciador de filas converte o nome do objeto em um nome de arquivo válido onde necessário.

Referências relacionadas

“Regras para Nomear os Objetos IBM MQ” na página 38

Os nomes de objeto IBM MQ possuem comprimentos máximos e fazem distinção entre maiúsculas e minúsculas. Nem todos os caracteres são suportados para cada tipo de objeto e vários objetos possuem regras a respeito da exclusividade de nomes.

Regras para Nomear os Objetos IBM MQ

Os nomes de objeto IBM MQ possuem comprimentos máximos e fazem distinção entre maiúsculas e minúsculas. Nem todos os caracteres são suportados para cada tipo de objeto e vários objetos possuem regras a respeito da exclusividade de nomes.

Existe vários tipos diferentes de objeto IBM MQ e todos os objetos de cada tipo possuem o mesmo nome porque existem nos espaços de nome de objeto separados: Por exemplo, um nome local e um canal emissor podem ter o mesmo nome. No entanto, um objeto não pode ter o mesmo nome que outro objeto no mesmo namespace: Por exemplo, uma fila local não pode ter o mesmo nome que uma fila de modelo e um canal emissor não pode ter o mesmo nome que um canal receptor.

Os seguintes objetos IBM MQ existem em namespaces de objeto separados:

- Informações sobre Autenticação
- Canal
- Canal do cliente
- Listener
- Lista de Nomes
- Processo
- Fila
- Serviço
- Classe de armazenamento
- Assinatura
- Tópico

Comprimento do Caractere de Nomes de Objeto

Em geral, os nomes de objetos do IBM MQ podem ter até 48 caracteres de comprimento. Essa regra se aplica aos seguintes objetos:

- Informações sobre Autenticação
- Cluster
- Listener

- Lista de Nomes
- Definição de processo
- Fila
- Gerenciador de filas
- Serviço
- Assinatura
- Tópico

Existem restrições:

1. **z/OS** Nos sistemas z/OS, os gerenciadores de fila devem ter, no máximo, 4 caracteres e devem estar em maiúsculas e caracteres numéricos apenas.
2. O comprimento máximo de nomes de objeto de canal e os nomes de canal de conexão do cliente é 20 caracteres. Consulte [Definindo os canais](#) para obter informações adicionais sobre canais.
3. As sequências de tópicos podem ter no máximo 10240 bytes. Todos os nomes de objeto IBM MQ fazem distinção entre maiúsculas e minúsculas.
4. Os nomes de assinatura podem ter um máximo de 10240 bytes e podem conter espaços.
5. O comprimento máximo de nomes de classe de armazenamento é de 8 caracteres.
6. O comprimento máximo de nomes de estrutura CF é de 12 caracteres.

Caracteres nos Nomes do Objeto

Os caracteres válidos para os nomes de objeto IBM MQ são:

Caracteres	Restrições
A - Z maiúscula	<ul style="list-style-type: none"> • Nenhum
a - z minúscula	<ul style="list-style-type: none"> • Em scripts do MQSC, nomes com caracteres minúsculos devem ser colocados entre aspas simples. Isso evita que caracteres minúsculos sejam convertidos em maiúsculos. • Os sistemas que usam EBCDIC Katakana não podem usar caracteres a- z minúsculos nos nomes do objeto. • z/OS Podem haver restrições ao usar caracteres minúsculos em sistemas z/OS, por exemplo, os nomes do gerenciador de filas não podem conter caracteres minúsculos. • IBM i Nos sistemas IBM i, durante o uso de comandos da CL, nomes com caracteres minúsculos devem ser colocados entre aspas simples. Isso evita que caracteres minúsculos sejam convertidos em maiúsculos.
Numéricos 0 - 9	<ul style="list-style-type: none"> • Nenhum
Ponto (.)	<ul style="list-style-type: none"> • Nenhum

Caracteres	Restrições
Sublinhado (_)	<ul style="list-style-type: none"> ▶ Multi Nenhum ▶ z/OS Evite usar os nomes com sublinhados iniciais ou finais porque eles não podem ser manipulados pelas operações IBM MQ for z/OS e pelos painéis de controle.
Barra (/)	<ul style="list-style-type: none"> ▶ Windows Em sistemas Windows, o primeiro caractere em um nome do gerenciador de filas não pode ser uma barra. ▶ IBM i Nos sistemas IBM i, durante o uso de comandos da CL, nomes contendo uma barra devem ser colocados entre aspas simples. ▶ z/OS Nenhum
Sinal de percentual (%)	<ul style="list-style-type: none"> ▶ ALW Nenhum ▶ z/OS Se você estiver usando o RACF como o gerenciador de segurança externa para IBM MQ for z/OS, não use % em nomes de objetos porque os nomes não são incluídos nas verificações de segurança quando perfis genéricos do RACF são utilizados. ▶ IBM i Nos sistemas IBM i, durante o uso de comandos da CL, nomes contendo um sinal de porcentagem devem ser colocados entre aspas simples.

Também há algumas regras gerais a respeito dos caracteres nos nomes do objeto:

1. Os espaços em branco integrado ou iniciais não são permitidos.
2. Os caracteres do idioma nacional não são permitidos.
3. Qualquer nome que seja inferior ao comprimento integral do campo pode ser preenchido à direita com espaços em branco. Todos os nomes abreviados que são retornados pelo gerenciador de filas são sempre preenchidos à direita com espaços em branco.

Nomes da Fila

O nome de uma fila tem duas partes:

- O nome de um gerenciador de filas
- O nome local da fila como é conhecido para esse gerenciador de filas

Cada parte do nome da fila tem 48 caracteres de comprimento.

Para consultar uma fila local, é possível omitir o nome do gerenciador de filas (substituindo-o pelos caracteres em branco ou usando um caractere nulo inicial). No entanto, todos os nomes de fila retornados a um programa pelo IBM MQ contêm o nome do gerenciador de filas.

▶ **z/OS** Uma fila compartilhada, acessível a qualquer gerenciador de filas em seu grupo de filas compartilhadas, não pode ter o mesmo nome que qualquer fila local não compartilhada no mesmo grupo de filas compartilhadas. Esta restrição evita a possibilidade de um aplicativo abrindo


equivocamente uma fila compartilhada quando destinada a abrir uma fila local ou vice-versa. As filas compartilhadas e os grupos de filas compartilhadas estão disponíveis apenas no IBM MQ for z/OS.

Para consultar uma fila remota, um programa deve incluir o nome do gerenciador de filas no nome de fila integral ou deve haver uma definição local da fila remota.

Quando um aplicativo usa um nome da fila, esse nome pode ser o nome de uma fila local (ou um alias para um) ou o nome de uma definição local de uma fila remota, mas o aplicativo não precisa saber qual, a menos que precise obter uma mensagem da fila (quando a fila tiver que ser local). Quando o aplicativo abre o objeto da fila, a chamada MQOPEN executa uma função de resolução de nome para determinar em qual fila executar as operações subsequentes. O significado disso é que o aplicativo não possui dependência integrada nas filas específicas sendo definidas em locais específicos em uma rede de gerenciadores de fila. Portanto, se um administrador do sistema realocar as filas na rede e alterar suas definições, os aplicativos que usam essas filas não precisarão ser alteradas.

Nomes do Objeto Reservado

Nomes de objetos que começam com SYSTEM. são reservados para objetos definidos pelo gerenciador de filas. É possível usar os comandos **Alter**, **Define** e **Replace** para alterar essas definições de objeto para adequar sua instalação. Os nomes que são definidos para IBM MQ são listados por completo em [Nomes da Fila](#).

 No IBM MQ for z/OS, o nome da estrutura do aplicativo do recurso de acoplamento CSQSYSAPPL é reservado.

Conceitos relacionados




[Nome de instalação no AIX, Linux, and Windows](#)

Nomes de arquivos do IBM MQ

Cada gerenciador de filas, fila, definição de processo, lista de nomes, canal, canal de conexão do cliente, listener, serviço e objeto de informações sobre autenticação do IBM MQ é representado por um arquivo. Como os nomes de objeto não são necessariamente nomes de arquivo válidos, o gerenciador de filas converte o nome do objeto em um nome de arquivo válido onde necessário.

O caminho padrão para um diretório do gerenciador de filas é o seguinte:

- Um prefixo, que é definido nas informações de configuração do IBM MQ:

-   No AIX and Linux, o prefixo padrão é /var/mqm. Isso é configurado na sub-rotina DefaultPrefix do arquivo de configuração mqs.ini.
-  Em sistemas Windows de 32 bits, o prefixo padrão é C:\Program Files (x86)\IBM\WebSphere MQ. Em sistemas Windows de 64 bits, o prefixo padrão é C:\Program Files\IBM\MQ. Para instalações de 32 e 64 bits, os diretórios de dados são instalados em C:\ProgramData\IBM\MQ. Isso é configurado na sub-rotina DefaultPrefix do arquivo de configuração mqs.ini.

Quando disponível, o prefixo pode ser mudado usando a página de propriedades do IBM MQ no IBM MQ Explorer, caso contrário, edite o arquivo de configuração do mqs.ini manualmente.

- O nome do gerenciador de filas é transformado em um nome de diretório válido. Por exemplo, o gerenciador de filas:

```
queue.manager
```

seria representando como:

```
queue!manager
```

Este processo é conhecido como *transformação de nomes*.

No IBM MQ, é possível fornecer a um gerenciador de filas um nome contendo até 48 caracteres.

Por exemplo, você poderia nomear um gerenciador de filas:

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

No entanto, cada gerenciador de filas é representado por um arquivo e existem limitações no comprimento máximo de um nome de arquivo e nos caracteres que podem ser usados no nome. Como resultado, os nomes de arquivos que representam os objetos são automaticamente transformados para atenderem aos requisitos do sistema de arquivos.


As regras que regem a transformação de um nome do gerenciador de filas são as seguintes:

1. Transformar caracteres individuais:
 - De ... para!
 - De / para &
2. Se o nome ainda não for válido:
 - a. Truncá-lo para oito caracteres
 - b. Anexar um sufixo numérico de três caracteres

Por exemplo, assumindo o prefixo padrão e um gerenciador de filas com o nome `queue.manager`:

-  No Windows com NTFS ou FAT32, o nome do gerenciador de filas torna-se:

```
C:\Program Files\IBM\MQ\mqgrs\queue!manager
```

-  No Windows com FAT, o nome do gerenciador de filas torna-se:

```
C:\Program Files\IBM\MQ\mqgrs\queue!ma
```

-   No AIX and Linux, o nome do gerenciador de filas torna-se:

```
/var/mqm/mqgrs/queue!manager
```

O algoritmo de transformação também se distingue entre os nomes que diferem apenas no caso nos sistemas de arquivos que não fazem distinção entre maiúsculas e minúsculas.

Transformação de Nome de Objeto

Os nomes de objeto não são necessariamente nomes do sistema de arquivos válidos. Talvez você precise transformar seus nomes de objeto. O método usado é diferente daquele para nomes do gerenciador de filas porque, embora existam apenas alguns nomes do gerenciador de filas em cada máquina, existe um grande número de outros objetos para cada gerenciador de filas. Filas, definições de processo, listas de nomes, canais, canais de conexão do cliente, listeners, serviços e objetos de informações de autenticação são representados no sistema de arquivos.

Quando um novo nome for gerado pelo processo de transformação, não haverá nenhum relacionamento simples com o nome do objeto original. É possível usar o comando **dspmqls** para converter entre nomes de objeto reais e transformados.

Referências relacionadas

dspmqls (exibir nomes de arquivos)

Informações relacionadas

Sub-rotina AllQueueManagers do arquivo `mq5.ini`

IBM i Nomes de objetos no IBM i

Um gerenciador de filas tem uma biblioteca de gerenciador de filas associada que tem um nome exclusivo. Os nomes de gerenciadores de filas e os nomes de objetos podem precisar ser transformados para atenderem aos requisitos do Sistema de Arquivos Integrado (IFS) do IBM i.

Quando um gerenciador de filas é criado, o IBM MQ associa uma biblioteca do gerenciador de filas a ele. Esta biblioteca do gerenciador de filas recebe um nome exclusivo, que não ultrapassa 10 caracteres de comprimento, amplamente baseado no nome do gerenciador de filas definido pelo usuário. O gerenciador de filas e a biblioteca do gerenciador de filas são colocados em um diretório que também é baseado no nome do gerenciador de filas com o prefixo `/QIBM/UserData/mqm`. Um exemplo de um gerenciador de filas, biblioteca do gerenciador de filas e diretório é o seguinte:

Nome do gerenciador de filas	LARANJA
Nome da Biblioteca do Gerenciador de Filas	QMORANGE
Diretório	<code>/QIBM/UserData/mqm/ORANGE</code>

Todos os nomes de gerenciadores de filas e nomes de bibliotecas de gerenciador de filas são gravados em sub-rotinas no arquivo `/QIBM/UserData/mqm/mqs.ini`.

Diretórios e arquivos IFS do IBM MQ

O IBM i Integrated File System (IFS) é usado extensivamente pelo IBM MQ para armazenar os dados. Para obter informações adicionais sobre o IFS consulte *Introdução ao Sistema de Arquivos Integrado*.

Cada objeto do IBM MQ, por exemplo, um canal ou um gerenciador de filas, é representado por um arquivo. Como os nomes de objeto não são necessariamente nomes de arquivo válidos, o gerenciador de filas converte o nome do objeto em um nome de arquivo válido onde necessário.

O caminho para um diretório do gerenciador de filas é formado a partir do seguinte:

- Um prefixo, que é definido no arquivo de configuração do gerenciador de filas, `qm.ini`. O prefixo padrão é `/QIBM/UserData/mqm`.
- Um literal, `qmgrs`.
- Um nome do gerenciador de filas codificado, que é o nome do gerenciador de filas transformado em um nome do diretório válido. Por exemplo, o gerenciador de filas `queue/manager` é representado por `queue&manager`.

Este processo é referido como a transformação do nome.

Transformação de Nome do Gerenciador de Filas IFS

No IBM MQ, é possível fornecer a um gerenciador de filas um nome contendo até 48 caracteres.

Por exemplo, é possível nome um gerenciador de filas `QUEUE/MANAGER/ACCOUNTING/SERVICES`. Da mesma maneira que uma biblioteca é criada para cada gerenciador de filas, cada gerenciador de filas também é representado por um arquivo. Por causa dos pontos de código variantes em EBCDIC, existem limitações aos caracteres que podem ser usados no nome. Como resultado, os nomes de arquivos IFS que representam os objetos são automaticamente transformados para atenderem aos requisitos do sistema de arquivos.

Usando o exemplo de um gerenciador de filas com o nome `queue/manager`, transformando o caractere `/` a `&` e assumindo o prefixo padrão, o nome do gerenciador de filas no IBM MQ for IBM i se torna `/QIBM/UserData/mqm/qmgrs/queue&manager`.

Transformação de Nome de Objeto

Os nomes de objeto não são necessariamente nomes do sistema de arquivos válidos; portanto, os nomes do objeto podem precisar ser transformados. O método usado é diferente daquele para os nomes do gerenciador de filas porque, embora existam apenas alguns nomes do gerenciador de filas para cada

máquina, pode haver um grande número de outros objetos para cada gerenciador de filas. Apenas as definições de processo, filas e listas de nomes são representadas no sistema de arquivos; os canais não são afetados por essas considerações.

Quando um novo nome for gerado pelo processo de transformação, não haverá nenhum relacionamento simples com o nome do objeto original. É possível usar o comando DSPMQMOBJN para visualizar os nomes transformados para objetos do IBM MQ.

Enfileiramento distribuído e clusters

Enfileiramento distribuído significa enviar mensagens de um gerenciador de filas para outro. O gerenciador de filas de recebimento pode estar na mesma máquina ou em outra; perto ou do outro lado do mundo. Ele pode estar sendo executado na mesma plataforma que o gerenciador de filas locais ou pode estar em qualquer uma das plataformas suportadas pelo IBM MQ. É possível definir manualmente todas as conexões em um ambiente de enfileiramento distribuído ou você pode criar um cluster e permitir que o IBM MQ defina a maioria dos detalhes da conexão para você.

Enfileiramento distribuído

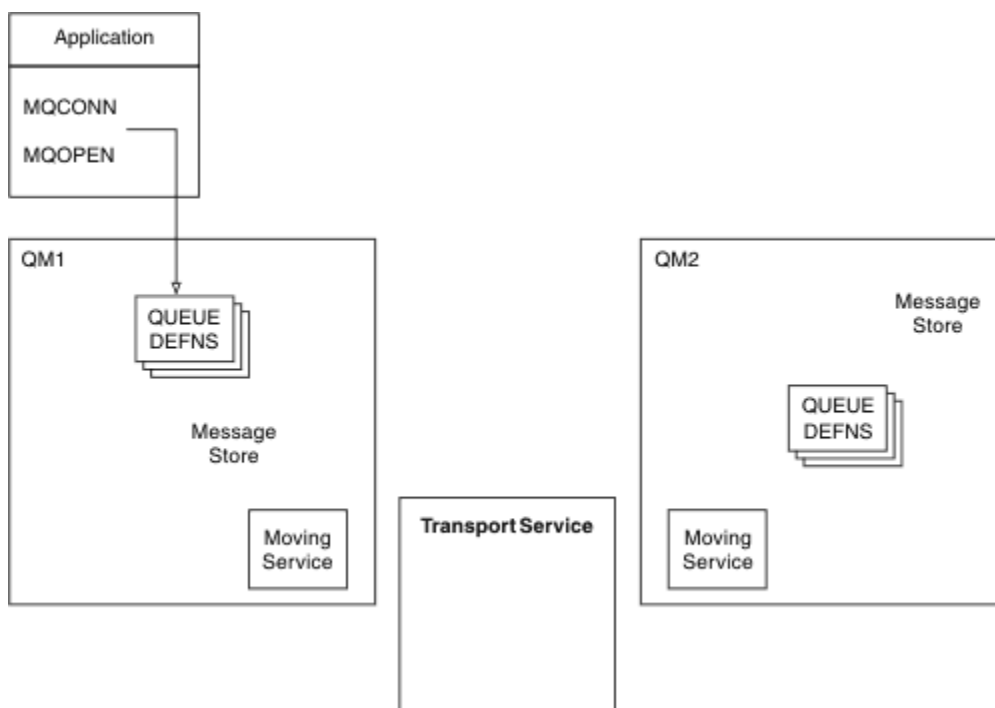


Figura 4. Visão Geral dos Componentes de Enfileiramento Distribuído

Na figura anterior:

- Um aplicativo usa a chamada MQCONN para conectar-se a um gerenciador de filas. O aplicativo usa então a chamada MQOPEN para abrir uma fila para que possa colocar mensagens na fila.
- Cada gerenciador de filas possui uma definição para cada uma de suas filas. Ele pode ter definições de *filas locais* (ou seja, hospedadas por este gerenciador de filas) e definições de *filas remotas* (ou seja, hospedadas por outros gerenciadores de filas).
- Se as mensagens forem destinadas a uma fila remota, o gerenciador de filas locais as manterá em uma *fila de transmissão*, que as persiste em um armazenamento de mensagem, até que elas possam ser encaminhadas para o gerenciador de filas remotas.
- Cada gerenciador de filas contém o software de comunicações, conhecido como o *serviço de movimentação*, que o gerenciador de filas usa para se comunicar com outros gerenciadores de filas.
- O *serviço de transporte* é independente do gerenciador de filas e pode ser qualquer um dos seguintes (dependendo da plataforma):

- Systems Network Architecture Advanced Program-to Program Communication (SNA APPC)
- TCP/IP (Transmission Control Protocol/Internet Protocol)
- Sistema BIOS de Rede (NetBIOS)
- Sequenced Packet Exchange (SPX)

Componentes necessários para enviar uma mensagem

Se uma mensagem tiver que ser enviada para um gerenciador de filas remotas, o gerenciador de filas locais precisará de definições para uma *fila de transmissão* e um *canal*. Um canal é um link de comunicação unidirecional entre dois gerenciadores de fila. Ele pode transportar as mensagens destinadas para qualquer número de filas no gerenciador de filas remotas.

Cada extremidade de canal possui uma definição separada, definindo-a, por exemplo, como a extremidade de envio ou extremidade de recebimento. Um canal simples consiste em uma definição de canal *emissor* no gerenciador de filas locais e definição de canal *receptor* no gerenciador de filas remotas. Essas duas definições devem ter o mesmo nome e juntas elas constituem um canal.

O software que manipula o envio e o recebimento de mensagens é chamado *Message Channel Agent* (MCA). Há um *agente do canal de mensagens* (MCA) em cada extremidade de um canal.

Cada gerenciador de filas deve ter uma *fila de devoluções* (também conhecida como a *fila de mensagens não entregues*). As mensagens são colocadas nesta fila se não puderem ser entregues a seus destinos.

A figura a seguir mostra o relacionamento entre os gerenciadores de filas, as filas de transmissão, os canais e os MCAs:

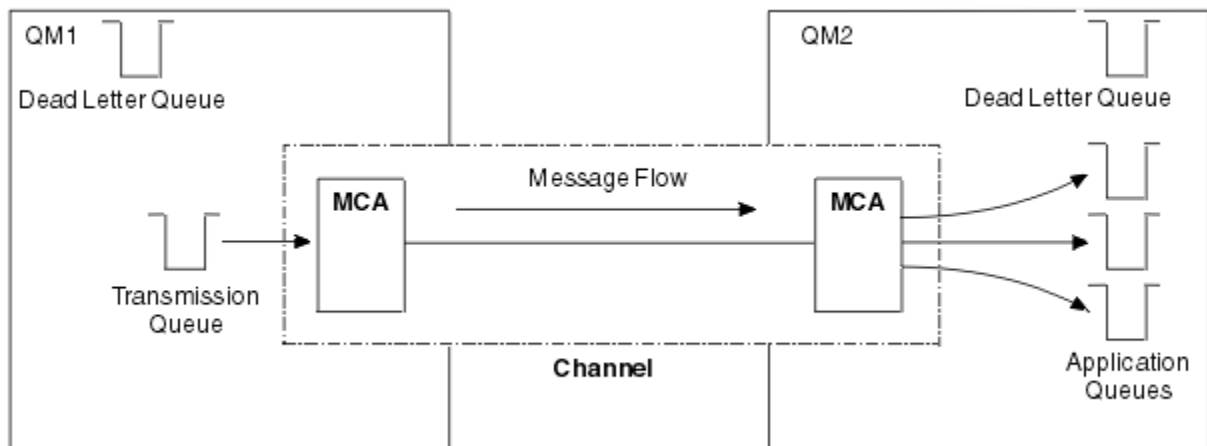


Figura 5. Enviando mensagens

Componentes necessários para retornar uma mensagem

Se o seu aplicativo requer que as mensagens sejam retornadas do gerenciador de filas remotas, é necessário definir outro canal para execução na direção oposta entre os gerenciadores de fila, conforme mostrado na figura a seguir:

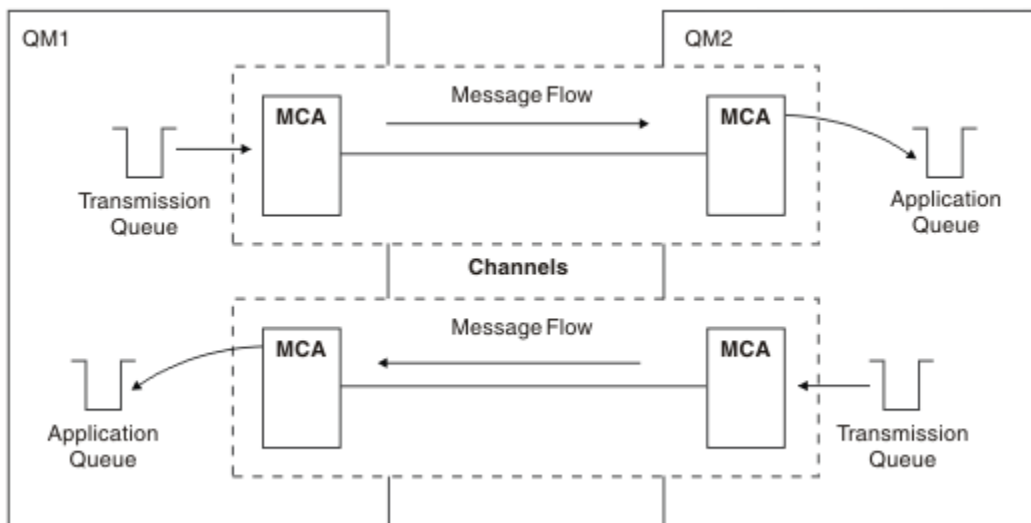


Figura 6. Enviando as Mensagens em Ambas as Direções

Clusters

Em vez de definir manualmente todas as conexões em um ambiente de enfileiramento distribuído, é possível agrupar um conjunto de gerenciadores de filas em um cluster. Ao fazer isso, os gerenciadores de filas podem disponibilizar as filas que eles hospedam para outros gerenciadores de filas no cluster, sem a necessidade de definições de canal explícitas, definições de filas remotas ou filas de transmissão para cada destino. Cada gerenciador de filas em um cluster possui uma única fila de transmissão que transmite as mensagens a qualquer outro gerenciador de filas no cluster. Para cada gerenciador de filas, você só precisa definir um canal do receptor de clusters e um canal do emissor de clusters; quaisquer canais adicionais serão automaticamente gerenciados pelo cluster.

Um cliente IBM MQ pode se conectar a um gerenciador de filas que faz parte de um cluster, assim como ele pode se conectar a qualquer outro gerenciador de filas. Como com o enfileiramento distribuído configurado manualmente, você usa a chamada MQPUT para colocar uma mensagem em uma fila em qualquer gerenciador de filas. Use a chamada MQGET para recuperar as mensagens de uma fila local.

Os gerenciadores de filas nas plataformas que suportam clusters não precisam fazer parte de um cluster. É possível continuar configurando manualmente o enfileiramento distribuído bem como ou em vez de, usar clusters.

Benefícios do uso de clusters

O armazenamento em cluster fornece dois benefícios principais:

- Os clusters simplificam a administração de redes do IBM MQ, que geralmente requerem que muitas definições de objetos para canais, filas de transmissão e filas remotas sejam configuradas. Esta situação é especialmente verdadeira em redes grandes, de mudança em potencial, nas quais muitos gerenciadores de filas precisam estar interconectados. Esta arquitetura é especialmente difícil para configurar e ativamente manter.
- Clusters podem ser usados para distribuir a carga do tráfego de mensagens em filas e gerenciadores de filas no cluster. Esta distribuição permite que a carga de mensagens de uma fila única seja distribuída entre as instâncias equivalentes dessa fila localizadas em vários gerenciadores de filas. Esta distribuição da carga de trabalho pode ser usada para atingir maior resiliência para falhas do sistema e melhorar o desempenho de ajuste de escala de fluxos de mensagens particularmente ativas em um sistema. Em tal ambiente, cada uma das instâncias das filas distribuídas possuem aplicativos consumidores processando as mensagens. Para obter informações adicionais, consulte [Usando clusters para gerenciamento de carga de trabalho](#).

Como as mensagens são roteadas em um cluster

É possível considerar um cluster como uma rede de gerenciadores de filas mantida por um administrador de sistemas consciencioso. Sempre que definir uma fila de clusters, o administrador de sistemas cria automaticamente as definições de fila remota correspondentes conforme necessário nos outros gerenciadores de filas.

Você não precisa criar definições de fila de transmissão porque o IBM MQ fornece uma fila de transmissão em cada gerenciador de filas no cluster. Esta única fila de transmissão pode ser usada para transportar mensagens para qualquer outro gerenciador de filas no cluster. Você não está limitado ao uso de uma fila de transmissão única. Um gerenciador de filas pode usar diversas filas de transmissão para separar as mensagens que vão para cada gerenciador de filas em um cluster. Geralmente, um gerenciador de filas usa uma fila de transmissão do cluster única. É possível alterar o atributo do gerenciador de filas DEFCLXQ, de forma que um gerenciador de filas use uma fila de transmissão de cluster diferente para cada gerenciador de filas em um cluster. Também é possível definir filas de transmissão do cluster manualmente.

Todos os gerenciadores de filas que se unem a um cluster concordam em trabalhar desta maneira. Eles enviam informações sobre eles mesmos e sobre as filas que eles hospedam e recebem informações sobre os outros membros do cluster.

Para assegurar que nenhuma informação seja perdida quando um gerenciador de filas se tornar indisponível, especifique dois gerenciadores de filas no cluster para agirem como *repositórios completos*. Esses gerenciadores de filas armazenam um conjunto completo de informações sobre todos os gerenciadores de filas e filas no cluster. Todos os outros gerenciadores de filas no cluster armazenam informações somente sobre os gerenciadores de filas e filas com os quais eles trocam mensagens. Esses gerenciadores de filas são conhecidos como *repositórios parciais*. Para obter mais informações, consulte [“Repositório de Cluster” na página 58](#).

Para se tornar parte de um cluster, um gerenciador de filas deve ter dois canais; um canal do emissor de clusters e um canal do receptor de clusters:

- Um canal do emissor de clusters é um canal de comunicação semelhante a um canal emissor. Você deve criar manualmente um canal do emissor de clusters em um gerenciador de filas para conectá-lo a um repositório completo que já é um membro do cluster.
- Um canal do receptor de clusters é um canal de comunicação semelhante a um canal receptor. Você deve criar manualmente um canal do receptor de clusters. O canal age como o mecanismo para o gerenciador de filas para receber comunicações do cluster.

Todos os outros canais que são necessários para a comunicação entre este gerenciador de filas e outros membros do cluster são então criados automaticamente.

A figura a seguir mostra os componentes de um cluster chamado CLUSTER:

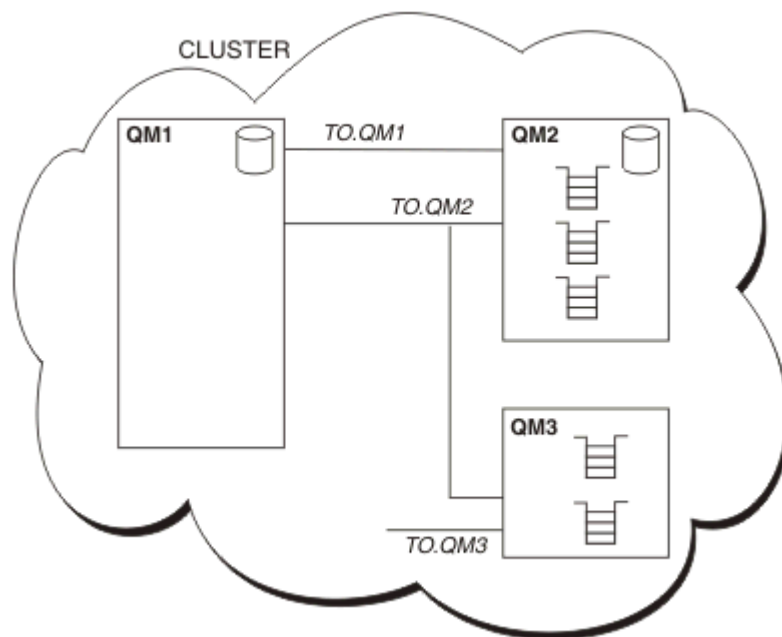


Figura 7. Um Cluster de Gerenciadores de Filas

- CLUSTER contém três gerenciadores de filas, QM1, QM2 e QM3.
- QM1 e QM2 hospedam repositórios das informações sobre os gerenciadores de filas e as filas no cluster.
- QM2 e QM3 hospedam algumas filas de cluster ou seja, filas que estão acessíveis a qualquer outro gerenciador de filas no cluster.
- Cada gerenciador de filas possui um canal do receptor de clusters denominado TO.qmgr no qual pode receber as mensagens.
- Cada gerenciador de filas também possui um canal do emissor de clusters no qual pode enviar as informações a um dos gerenciadores de fila do repositório.
- QM1 e QM3 enviam ao repositório em QM2, e QM2 envia ao repositório em QM1.

Componentes de Enfileiramento Distribuído

Os componentes de enfileiramento distribuído são canais de mensagens, agentes do canal de mensagens, filas de transmissão, inicializadores e listeners de canais e programas de saída do canal. A definição de cada extremidade de um canal de mensagens pode ser um entre vários tipos.

Canais de mensagens são os canais que transportam as mensagens de um gerenciador de filas a outro. Não confunda canais de mensagens com canais MQI. Há dois tipos de canal MQI, conexão do servidor (SVRCONN) e conexão do cliente (CLNTCONN). Para obter mais informações, consulte [Canais](#).

A definição de cada extremidade de um canal de mensagens pode ser um dos seguintes tipos:

- Emissor (SDR)
- Receptor (RCVR)
- Servidor (SVR)
- Solicitante (RQSTR)
- Emissor de cluster (CLUSDR)
- Receptor de cluster (CLUSRCVR)

Um canal de mensagens é definido usando um desses tipos definidos em uma extremidade e um tipo compatível na outra extremidade. As combinações possíveis são:

- Emissor-receptor
- Solicitante-servidor
- Solicitante-emissor (retorno de chamada)
- Servidor-receptor
- Emissor de cluster-receptor de cluster

Instruções detalhadas para criar um canal emissor-receptor estão incluídas em [Definindo os canais](#). Para obter exemplos dos parâmetros necessários para configurar canais do emissor-receptor, consulte [Informações de Configuração de Exemplo](#) aplicável à sua plataforma. Para os parâmetros necessários para definir um canal de qualquer tipo, consulte [DEFINE CHANNEL](#).

Canais Emissor-Receptor

Um emissor em um sistema inicia o canal para que possa enviar as mensagens a outro sistema. O emissor solicita que o receptor na outra extremidade do canal seja iniciado. O emissor envia as mensagens de sua fila de transmissão para o receptor. O receptor coloca as mensagens na fila de destino. [Figura 8 na página 49](#) ilustra isso.

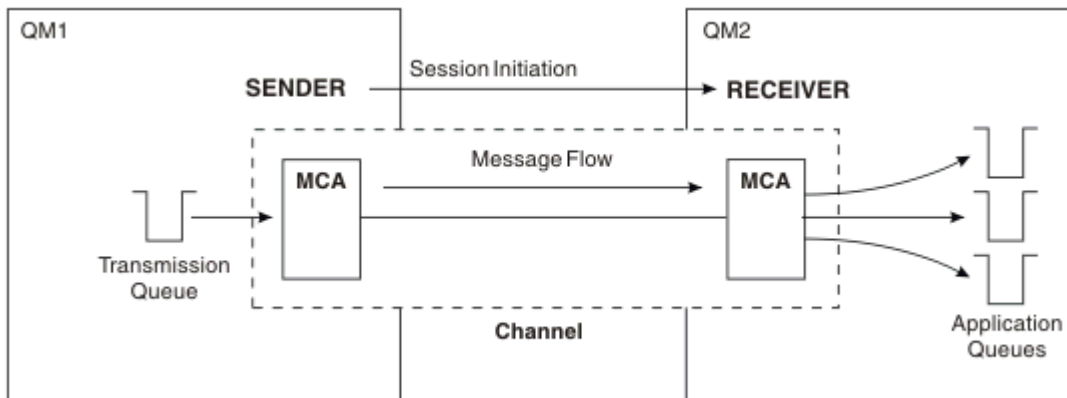


Figura 8. Um Canal Emissor-Receptor

Canais solicitante-servidor

Um solicitante em um sistema inicia o canal para que possa receber as mensagens do outro sistema. O solicitante solicita que o servidor na outra extremidade do canal seja iniciado. O servidor envia as mensagens ao solicitante da fila de transmissão definida em sua definição de canal.

Um canal do servidor também inicia a comunicação e envia as mensagens a um solicitante. Isso se aplica apenas a servidores *completos*, que sejam canais de servidor que tenham o nome de conexão do parceiro especificado na definição de canal. Um servidor completo pode ser iniciado por um solicitante ou pode iniciar uma comunicação com um solicitante.

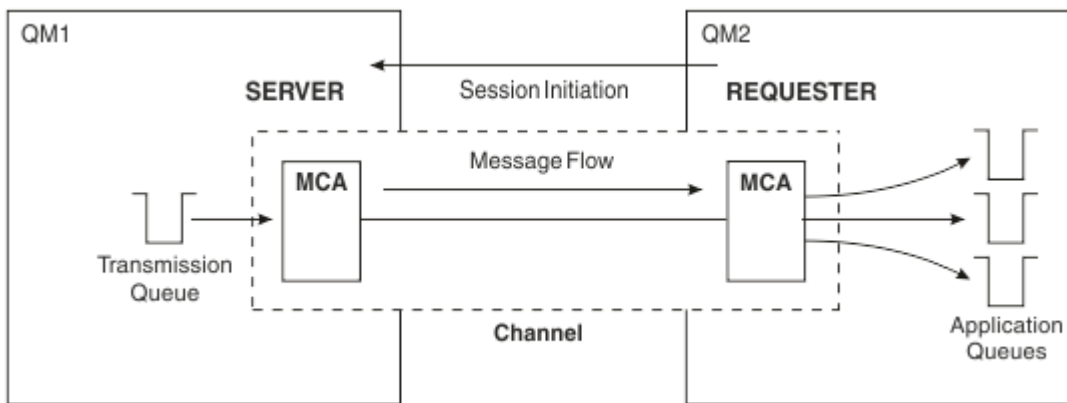


Figura 9. Um Canal Solicitante-Servidor

Canais solicitante-emissor

O solicitante inicia o canal e o emissor termina o canal. O emissor então reinicia a comunicação de acordo com as informações em sua definição de canal (conhecida como *retorno de chamada*). Ele envia as mensagens da fila de transmissão para o solicitante.

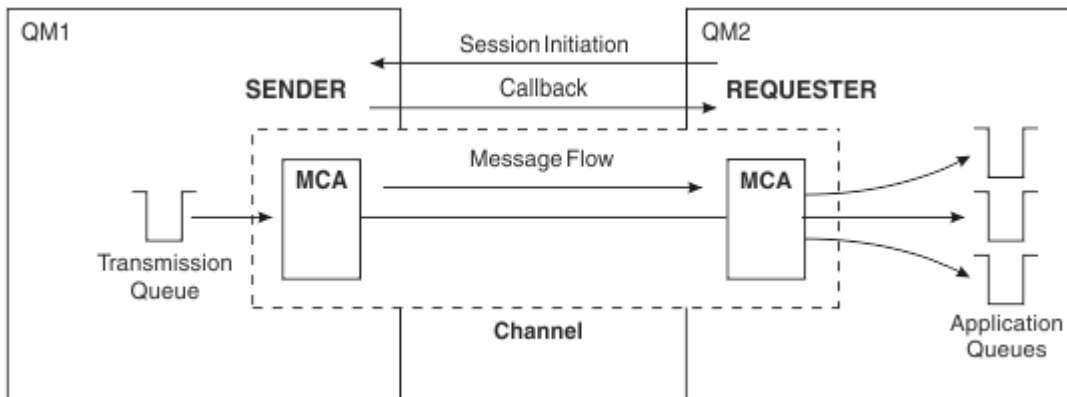


Figura 10. Um Canal Solicitante-Emissor

Canais Servidor-Receptor

Isso é como o emissor-receptor mas se aplica apenas a servidores *completos*, que sejam canais de servidor que tenham o nome de conexão do parceiro especificado na definição de canal. A inicialização do canal deve ser iniciada na extremidade do servidor do link. A ilustração disso é como a ilustração no [Figura 8 na página 49](#).

Canais do Emissor de Clusters

Em um cluster, cada gerenciador de filas tem um canal do emissor de clusters no qual pode enviar as informações de cluster a um dos gerenciadores de filas de repositório completo. Os gerenciadores de filas também podem enviar as mensagens a outros gerenciadores de filas nos canais do emissor de clusters.

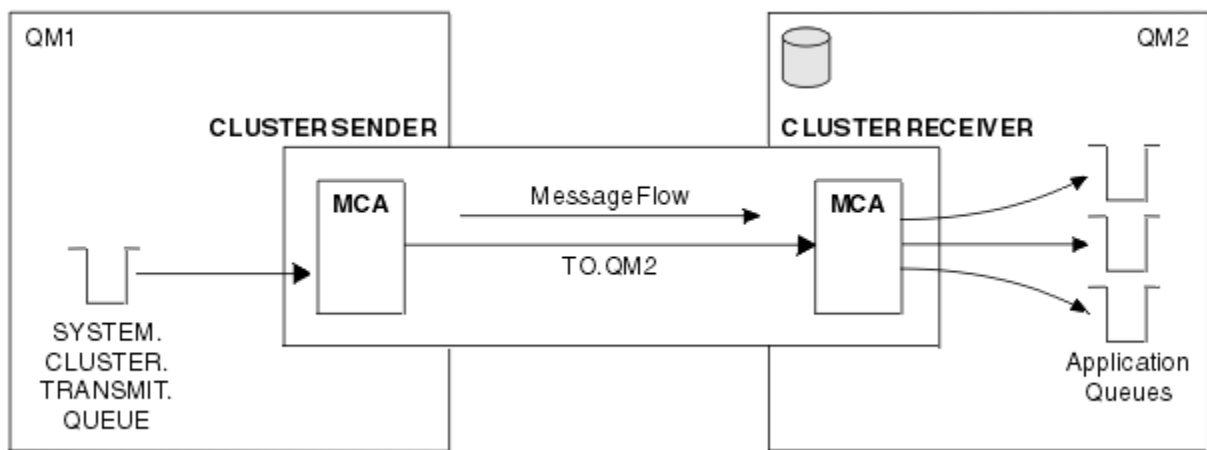


Figura 11. Um Canal do Emissor de Clusters

Canais do Receptor de Clusters

Em um cluster, cada gerenciador de filas tem um canal do receptor de clusters no qual pode receber as mensagens e informações sobre o cluster. A ilustração disso é como a ilustração no [Figura 11 na página 51](#).

Filas de Devoluções

A fila de devoluções (ou fila de mensagens não entregues) é a fila para a qual as mensagens são enviadas se não puderem ser roteadas para seus destinos corretos. Cada gerenciador de filas geralmente possui uma fila de devoluções.

Uma *fila de devoluções* (DLQ), às vezes referida como uma *fila de mensagens não entregues*, é uma fila de contenção para mensagens que não podem ser entregues em suas filas de destino, por exemplo, porque a fila não existe ou está cheia. As filas de devoluções também são usadas na extremidade de envio de um canal, para erros de conversão de dados. Cada gerenciador de filas em uma rede geralmente tem uma fila local para ser usada como uma fila de devoluções de forma que mensagens que não puderem ser entregues ao seu destino correto possam ser armazenadas para recuperação posterior.

As mensagens podem ser colocadas no DLQ 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. O campo *Razão* da estrutura MQDLH contém um código de razão que identifica o motivo pelo qual a mensagem está no DLQ.

Geralmente é necessário definir uma fila de devoluções para cada gerenciador de filas. Se não fizer isso e o MCA não conseguir colocar uma mensagem, ela será deixada na fila de transmissão e o canal será parado. Além disso, se as mensagens rápidas e não persistentes (consulte [mensagens rápidas e não persistentes](#)) não puderem ser entregues e não existir nenhuma fila de devolução no sistema de destino, essas mensagens serão descartadas.

No entanto, usar as filas de devoluções pode afetar a sequência na qual as mensagens são entregues e assim você pode optar por não usá-las.

Tarefas relacionadas

[Trabalhando com filas de mensagens não entregues](#)

[Resolução de problemas de mensagens não entregues](#)

Referências relacionadas

[runmqdlq \(executar manipulador da fila de devoluções\)](#)

Definições de fila remota

As definições de fila remotas são definições para as filas que são de propriedade de outro gerenciador de filas.

Embora os aplicativos possam recuperar as mensagens apenas de filas locais, elas podem colocar as mensagens em filas locais ou filas remotas. Portanto, assim como uma definição para cada uma de suas filas locais, um gerenciador de filas pode ter *definições de fila remota*. A vantagem das definições de fila remota é que elas permitem que um aplicativo coloque uma mensagem em uma fila remota sem ter que especificar o nome da fila remota ou o gerenciador de filas remotas ou o nome da fila de transmissão. As definições de fila remota fornecem independência de local.

Existem outros usos para as definições de fila remota, que serão descritas posteriormente.

Como Chegar ao Gerenciador de Filas Remotas

Talvez nem sempre você tenha um canal entre cada gerenciador de filas de origem e de destino. Existem inúmeras outras maneiras de se vincular entre as duas, incluindo canais de compartilhamento de diversos hops, usando diferentes canais e clusters.

Diversos Hops

Se não houver nenhum link de comunicação direta entre o gerenciador de filas de origem e o gerenciador de filas de destino, é possível passar um ou mais *gerenciadores de fila intermediários* no caminho para o gerenciador de filas de destino. Isso é conhecido como *diversos hops*.

Você precisa definir os canais entre todos os gerenciadores de fila e as filas de transmissão nos gerenciadores de fila intermediários. Isso é mostrado no [Figura 12 na página 52](#).

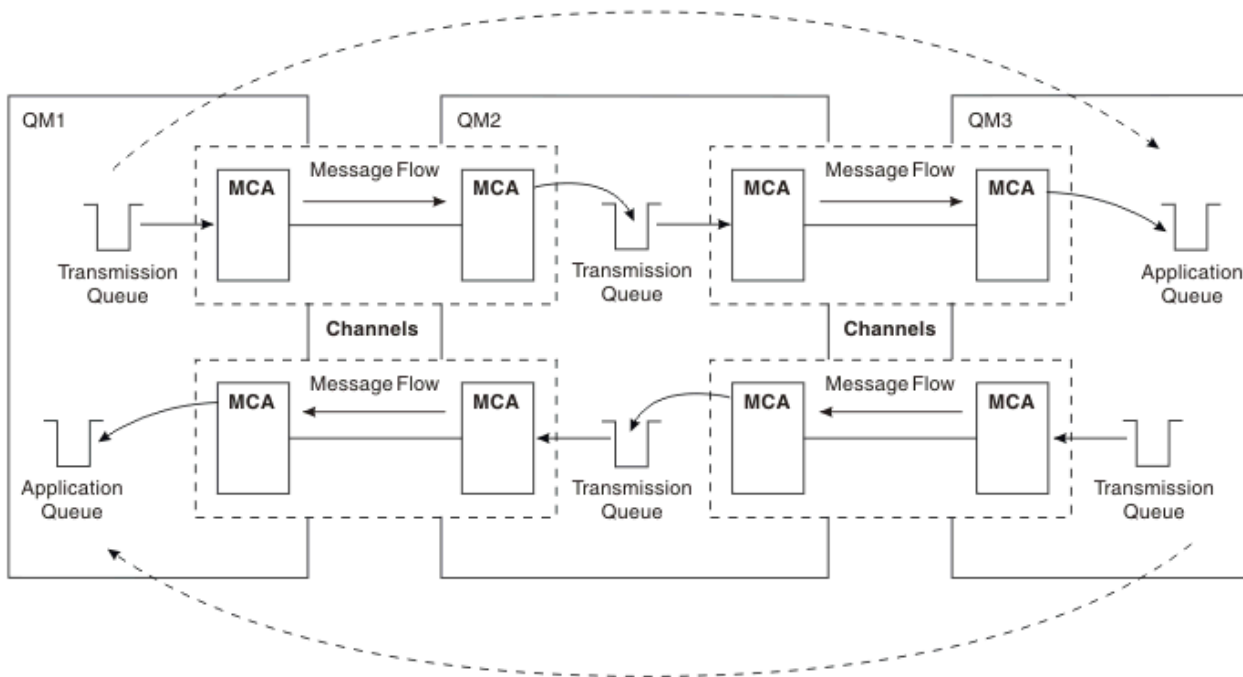


Figura 12. Passando pelos Gerenciadores de Fila Intermediários

Canais de Compartilhamento

Como um editor de telas, você tem a opção de forçar os seus aplicativos a especificar o nome do gerenciador de fila remoto juntamente com o nome da fila ou criar uma *definição de fila remota* para cada fila remota. Esta definição retém o nome do gerenciador de filas remotas, o nome da fila e o nome da fila de transmissão. De qualquer maneira, todas as mensagens de todos os aplicativos que endereçam as

filas no mesmo local remoto possuem suas mensagens enviadas por meio da mesma fila de transmissão. Isso é mostrado no [Figura 13 na página 53](#).

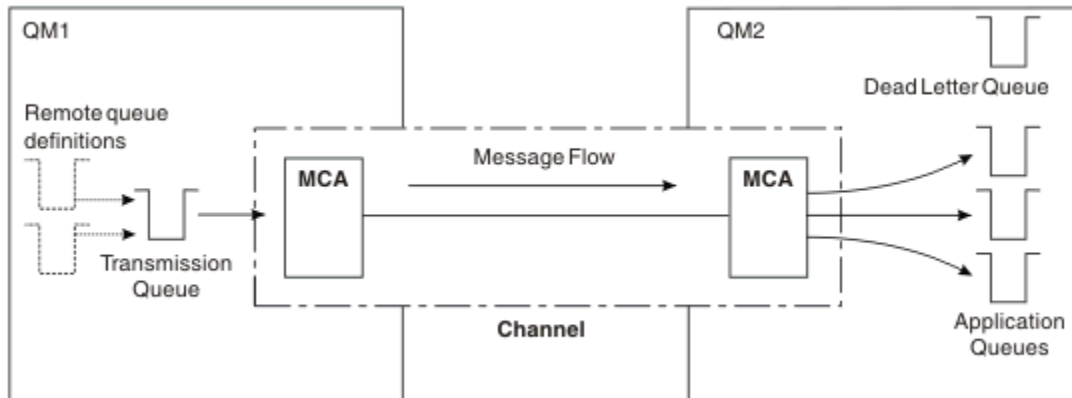


Figura 13. Compartilhando uma Fila de Transmissão

Figura 13 na página 53 ilustra quais mensagens dos diversos aplicativos para diversas filas remotas podem usar o mesmo canais.

Usando Canais Diferentes

Se você tiver mensagens de diferentes tipos a serem enviados entre dois gerenciadores de fila, poderá definir mais de um canal entre os dois. Há momento em que são necessários canais alternativos, talvez por propósitos de segurança ou para negociar a velocidade de entrega no montante absoluto do tráfego da mensagem.

Para configurar um segundo canal, você precisa definir outro canal e outra fila de transmissão e criar uma definição de fila remota que especifica o local e o nome da fila de transmissão. Seus aplicativos podem então usar o canal, mas as mensagens ainda são entregues nas mesmas filas de destino. Isso é mostrado no [Figura 14 na página 53](#).

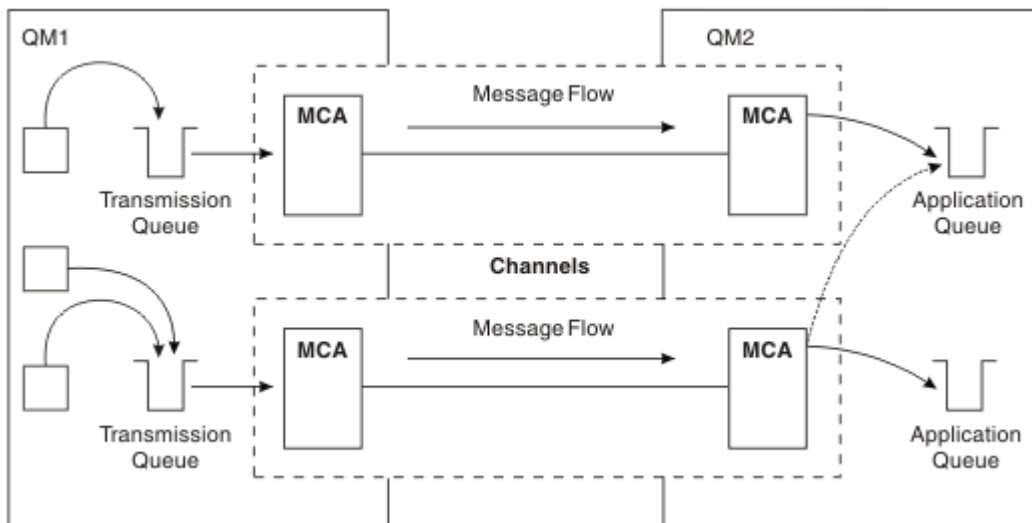


Figura 14. Usando Diversos Canais

Ao usar as definições de fila remota para especificar uma fila de transmissão, seus aplicativos **não** devem especificar o local (ou seja, o gerenciador da fila de destino) sozinhos. Se o fizerem, o gerenciador de filas não usará as definições de fila remota. As definições de fila remota fornecem independência de local. Os aplicativos podem colocar as mensagens em uma fila *lógica* sem saber onde a fila está localizada e você pode alterar a fila *física* sem ter que alterar os seus aplicativos.

Usando o Armazenamento em Cluster

Cada gerenciador de filas em um cluster define um canal do receptor de clusters. Quando outro gerenciador de filas desejar enviar uma mensagem para esse gerenciador de filas, ele definirá o canal do emissor de clusters correspondente automaticamente. Por exemplo, se houver mais de uma instância de uma fila em um cluster, o canal do emissor de clusters poderia ser definido em qualquer um dos gerenciadores de filas que hospede a fila. IBM MQ usa o algoritmo de gerenciamento de carga que usa uma rotina round-robin para selecionar um gerenciador de filas disponível para o qual rotear uma mensagem. Para obter informações adicionais, consulte [Clusters](#).

Informações de Endereçamento

Quando um aplicativo coloca as mensagens que são destinadas para um gerenciador de filas remotas, o gerenciador de filas locais inclui um cabeçalho de transmissão nelas antes de colocá-las na fila de transmissão. Este cabeçalho contém o nome da fila de destino e o gerenciador de filas ou seja, as *informações de endereçamento*.

Em um ambiente de gerenciador de filas únicas, o endereço de uma fila de destino é estabelecido quando um aplicativo abre uma fila na qual colocar as mensagens. Como a fila de destino está no mesmo gerenciador de filas, não há necessidade de qualquer informação de endereçamento.

Em um ambiente de enfileiramento distribuído, o gerenciador de filas precisa conhecer não somente o nome da fila de destino, mas também o local dessa fila (ou seja, o nome do gerenciador de filas), e a rota para esse local remoto (ou seja, a fila de transmissão). As informações de endereçamento estão contidas no cabeçalho de transmissão. O canal de recebimento remove o cabeçalho de transmissão e usa as informações contidas nele para localizar a fila de destino.

É possível evitar a necessidade para que os seus aplicativos especifiquem o nome do gerenciador de fila de destino, se usar uma definição de fila remota. Esta definição especifica o nome da fila remota, o nome do gerenciador de filas remotas para o qual as mensagens estão destinadas e o nome da fila de transmissão usada para transportar as mensagens.

O que São Aliases?

Aliases são usados para fornecer a qualidade de serviço para as mensagens. O alias do gerenciador de filas permite que um administrador do sistema altere o nome de um gerenciador de filas de destino sem que tenha que alterar os seus aplicativos. Ele também permite que o administrador do sistema altere a rota para um gerenciador de filas de destino ou configure uma rota que envolva a passagem por inúmeros outros gerenciadores de fila (diversos hops). O alias da fila de resposta fornece a qualidade de serviço para as respostas.

Os aliases do gerenciador de filas e os aliases de fila de resposta são criados usando uma definição de fila remota que tem um RNAME em branco. Essas definições não definem as filas reais; elas são usadas pelo gerenciador de filas para resolver os nomes de fila física, os nomes do gerenciador de filas e as filas de transmissão.

As definições de alias são caracterizadas tendo um RNAME em branco.

Resolução de Nome da Fila

A resolução de nome da fila ocorre a cada gerenciador de filas sempre que uma fila é aberta. Seu propósito é identificar a fila de destino, o gerenciador de filas de destino (que pode ser local) e a rota para esse gerenciador de filas (que pode ser nulo). O nome resolvido tem três partes: o nome do gerenciador de filas, o nome da fila e, se o gerenciador de filas for remoto, a fila de transmissão.


Quando existir uma definição de fila remota, nenhuma definição de alias será referenciada. O nome da fila fornecido pelo aplicativo é resolvido no nome da fila de destino, gerenciador de filas remotas e fila de transmissão especificados na definição de fila remota. Para obter informações mais detalhadas sobre a resolução do nome da fila, consulte [Resolução do Nome da Fila](#).

Se não houver definição de fila remota e um nome do gerenciador de filas for especificado ou resolvido pelo serviço de nomes, o gerenciador de filas verificará se existe uma definição de alias de gerenciador

de filas que corresponda ao nome do gerenciador de filas fornecido. Se houver, as informações contidas nelas serão usadas para resolver o nome do gerenciador de filas para o nome do gerenciador de filas de destino. A definição de alias do gerenciador de filas também pode ser usada para determinar a fila de transmissão para o gerenciador de filas de destino.

Se o nome da fila resolvido não for uma fila local, o nome do gerenciador de filas e o nome da fila serão incluídos no cabeçalho de transmissão de cada mensagem colocada pelo aplicativo para a fila de transmissão.

A fila de transmissão geralmente usada tem o mesmo nome que o gerenciador de filas resolvido, a menos que mudado por uma definição de fila remota ou uma definição de alias de gerenciador de filas. Se você não definiu essa fila de transmissão, mas definiu uma fila de transmissão padrão, isso será usado.

 Nomes dos gerenciadores de fila sendo executados no z/OS são limitados a quatro caracteres.

Definições de Alias do Gerenciador de Filas

As definições de alias do gerenciador de filas se aplicam quando um aplicativo, que abre uma fila para colocar uma mensagem, especifica o nome da fila e o nome do gerenciador de filas.

As definições de alias do gerenciador de filas possuem três usos:

- Ao enviar as mensagens, remapeando o nome do gerenciador de filas
- Ao enviar as mensagens, alterando ou especificando a fila de transmissão
- Ao receber as mensagens, determinando se o gerenciador de filas locais é o destino desejado para essas mensagens

Mensagens de Saída - Remapeando o Nome do Gerenciador de Filas

As definições de alias do gerenciador de filas pode ser usado para remapear o nome do gerenciador de filas especificado em uma chamada MQOPEN. Por exemplo, uma chamada MQOPEN especifica um nome da fila igual a THISQ e um nome do gerenciador de filas igual a YOURQM. No gerenciador de filas locais, existe uma definição de alias do gerenciador de filas como o seguinte exemplo:

```
DEFINE QREMOTE (YOURQM) RQMNAME(REALQM)
```

Isto mostra que o gerenciador de filas real a ser usado, quando um aplicativo coloca mensagens no gerenciador de filas YOURQM, é REALQM. Se o gerenciador de filas locais é REALQM, ele coloca as mensagens na fila THISQ, a qual é uma fila local. Se o gerenciador de filas locais não for chamado de REALQM, ele roteará a mensagem para uma fila de transmissão chamada REALQM. O gerenciador de filas altera o cabeçalho de transmissão para informar REALQM em vez de YOURQM.

Mensagens de Saída - Alterando ou Especificando a Fila de Transmissão

Figura 15 na página 56 mostra um cenário no qual as mensagens chegam no gerenciador de filas QM1 com cabeçalhos de transmissão mostrando nomes da fila no gerenciador de filas QM3. Neste cenário, QM3 é acessível por saltos múltiplos por meio do QM2.

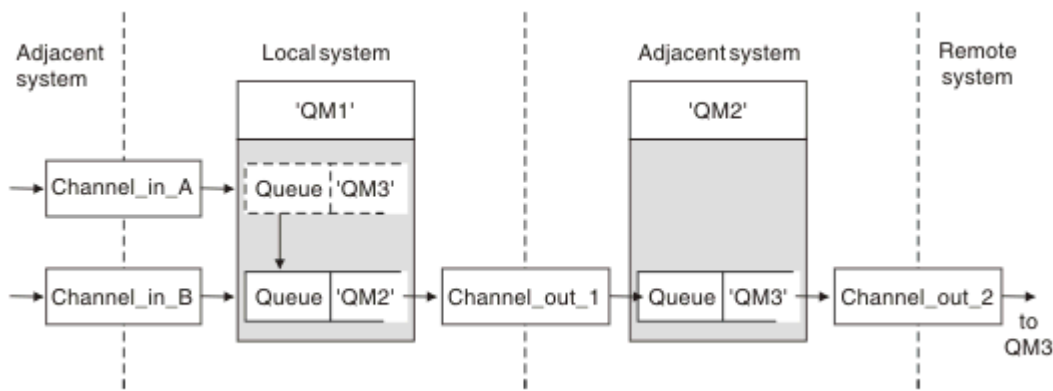


Figura 15. Alias do gerenciador de filas

Todas as mensagens para QM3 são capturadas em QM1 com um alias do gerenciador de filas. O alias do gerenciador de filas é denominado QM3 e contém a definição QM3 por meio da fila de transmissão QM2. A definição se parece com o seguinte exemplo:

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

O gerenciador de filas coloca as mensagens na fila de transmissão QM2 mas não altera o cabeçalho da fila de transmissão porque o nome do gerenciador de filas de destino, QM3, não é mudado.

Todas as mensagens chegam em QM1 e mostrando um cabeçalho de transmissão contendo um nome da fila em QM2 também são colocadas na fila de transmissão QM2. Desta maneira, as mensagens com diferentes destinos são coletadas em uma fila de transmissão comum para um sistema adjacente apropriado, para transmissões adicionais em seus destinos.

Mensagens de Entrada - Determinando o Destino

Um MCA de recebimento abre a fila referenciada no cabeçalho de transmissão. Se uma definição de alias do gerenciador de filas existir com o mesmo nome que o gerenciador de filas referenciado, o nome do gerenciador de filas recebido no cabeçalho de transmissão será substituído pelo RQMNAME a partir dessa definição.

Este processo tem dois usos:

- Direcionar as mensagens para outro gerenciador de filas
- Alterar o nome do gerenciador de filas para ser igual ao gerenciador de filas locais

Definições de Alias da Fila de Resposta

Uma definição de alias da fila de resposta especifica nomes alternativos para informações de resposta no descritor de mensagem. A vantagem disso é que você pode alterar o nome de uma fila ou o gerenciador de filas sem ter que alterar os seus aplicativos.

Resolução de Nome da Fila

Quando um aplicativo responde a uma mensagem, ele usa os dados no *descritor de mensagens* da mensagem recebida para descobrir o nome da fila para a qual responder. O aplicativo de envio indica para onde as respostas são enviadas e anexa essas informações a suas mensagens. Esse conceito deve ser coordenado como parte do design do aplicativo.

A resolução de nome de fila acontece na extremidade de envio do aplicativo antes de a mensagem ser colocada em uma fila. A resolução de nome de fila ocorre antes da interação com o aplicativo remoto para o qual a mensagem está sendo enviada. Esta é a única situação em que ocorre a resolução do nome no momento em que uma fila não estiver sendo aberta.

Resolução de Nome de Fila Usando um Alias de Gerenciador de Filas

Normalmente, um aplicativo especifica uma fila de resposta e deixa o nome do gerenciador de fila de resposta em branco. O gerenciador de filas conclui seu próprio nome no momento da inserção. Este método funciona bem, exceto quando você deseja que um canal alternativo seja usado para respostas, por exemplo, um canal que usa a fila de transmissão QM1_relief em vez do canal de retorno padrão que usa a fila de transmissão QM1. Nessa situação, os nomes de gerenciador de filas especificados em cabeçalhos de fila de transmissão não correspondem aos nomes de gerenciador de filas "reais", mas eles são especificados novamente usando definições de alias do gerenciador de filas. Para retornar as respostas juntamente com as rotas alternativas, é necessário mapear os dados da fila de resposta também, usando as definições de alias de fila de resposta.

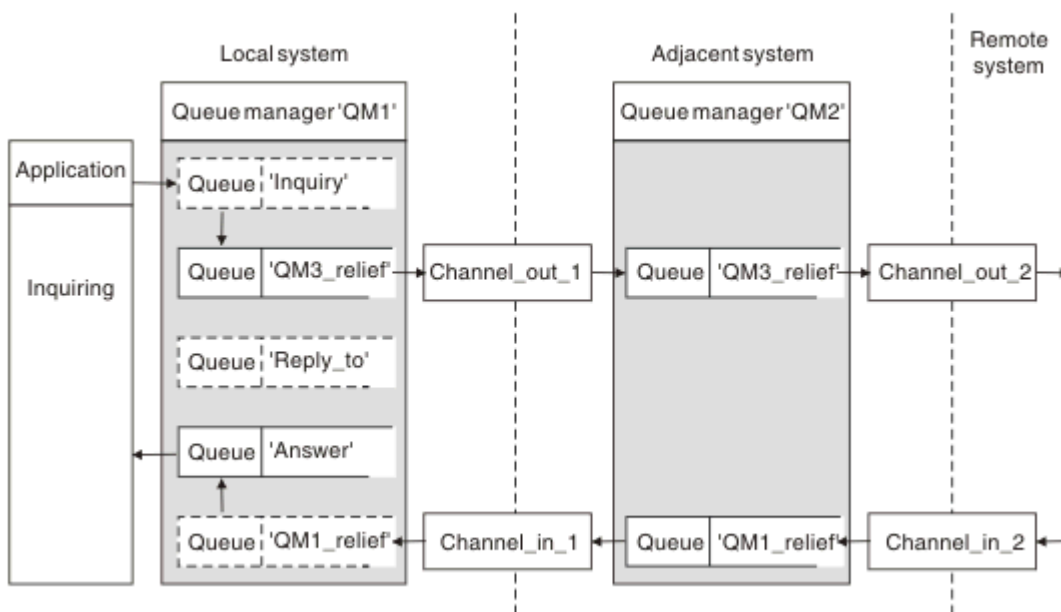


Figura 16. Alias da Fila de Resposta Usados para Alterar o Local de Resposta

No exemplo em [Figura 16](#) na página 57:

1. O aplicativo coloca uma mensagem que usa a chamada MQPUT e que especifica as seguintes informações no descritor de mensagens:

```
ReplyToQ='Reply_to'  
ReplyToQMgr=' '
```

ReplyToQMgr deve estar em branco para que o alias da fila de resposta seja usado.

2. Você cria uma definição de alias de fila de resposta chamada Reply_to, a qual contém o nome Answer, e o nome do gerenciador de filas QM1_relief.

```
DEFINE QREMOTE ('Reply_to') RNAME ('Answer')  
RQMNAME ('QM1_relief')
```

3. As mensagens são enviadas com um descritor de mensagens que mostra ReplyToQ='Answer' e ReplyToQMgr='QM1_relief'.
4. A especificação do aplicativo deve incluir a informação de que as respostas devem ser localizadas na fila Answer em vez de Reply_to.

Para preparar-se para as respostas, você tem que criar o canal de retorno paralelo, definindo:

- Em QM2, a fila de transmissão denominada QM1_relief

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- Em QM1, o alias do gerenciador de filas QM1_relief

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

Este alias do gerenciador de filas termina a cadeia de canais de retorno paralelo e captura as mensagens para QM1.

Se você acha que talvez queira fazer isso em algum momento no futuro, certifique-se que de os aplicativos usam o nome de alias do início. Por enquanto, esse é um alias de fila normal para a fila de resposta, mas posteriormente, poderá ser mudado para um alias do gerenciador de filas.

Nome da Fila de Resposta

É necessário cuidado com a nomeação das filas de resposta. A razão pela qual um aplicativo coloca um nome da fila de resposta na mensagem é que ela pode especificar a fila para a qual suas respostas são enviadas. Ao criar uma definição de alias da fila de resposta com este nome, você não pode ter a fila de resposta real (ou seja, uma definição de fila local) com o mesmo nome. Portanto, a definição de alias da fila de resposta deve conter um novo nome de fila bem como o nome do gerenciador de filas e a especificação do aplicativo deve incluir as informações nas quais suas respostas são encontradas nesta outra fila.

Os aplicativos agora tem que recuperar as mensagens de uma fila diferente daquela que nomearam como fila de resposta quando colocaram a mensagem original.

Componentes do Cluster

Os clusters são compostos de gerenciadores de fila, repositórios do cluster, canais de cluster e filas de cluster.

Consulte os subtópicos a seguir para obter informações sobre cada um dos componentes de cluster:

Conceitos relacionados

[Comparação de Armazenamento em Cluster e Enfileiramento Distribuído](#)



Tarefas relacionadas

[Configurando um cluster do gerenciador de filas](#)

[Configurando um novo cluster](#)

Repositório de Cluster

Um repositório é uma coleção de informações sobre os gerenciadores de filas que são membros de um cluster.

As informações do repositório incluem os nomes dos gerenciadores de filas, seus locais, seus canais, quais filas eles hospedam e outras informações. As informações são armazenadas na forma de mensagens em uma fila chamada SYSTEM.CLUSTER.REPOSITORY.QUEUE. Essa fila é um dos objetos padrão.  No [Multiplataformas](#), ela é definida quando você cria um gerenciador de filas do IBM MQ.  No IBM MQ for z/OS, ela é definida como parte da customização do gerenciador de filas.

Repositório Completo e Repositório Parcial

Geralmente, dois gerenciadores de filas em um cluster contêm um repositório completo. Todos os gerenciadores de filas restantes contêm um repositório parcial.

Um gerenciador de filas que hospeda um conjunto completo de informações sobre cada gerenciador de filas no cluster tem um repositório completo. Outros gerenciadores de filas no cluster têm repositórios parciais que contêm um subconjunto das informações nos repositórios completos.

Um repositório parcial contém informações apenas sobre os gerenciadores de filas com os quais o gerenciador de filas precisa trocar mensagens. Os gerenciadores de filas solicitam atualizações nas informações que eles precisam, de forma que, se elas forem alteradas, o gerenciador de filas do repositório completo enviará as novas informações. Na maior parte do tempo, um repositório parcial contém todas as informações que um gerenciador de filas precisa para executar no cluster. Quando um gerenciador de filas precisa de alguma informação adicional, ele faz consultas do repositório completo e atualiza seu repositório parcial. Os gerenciadores de filas usam a fila `SYSTEM.CLUSTER.COMMAND.QUEUE` para solicitar e receber atualizações para os repositórios.


Ao migrar os gerenciadores de filas que são membros de um cluster, migre os repositórios completos antes dos repositórios parciais. Isso é porque um repositório mais antigo não pode armazenar atributos mais novos introduzidos em uma liberação mais recente. Ele os tolera, mas não os armazena.

Gerenciador de filas do cluster

Um gerenciador de filas do cluster é um gerenciador de filas que é membro de um cluster.

Um gerenciador de filas pode ser membro de mais de um cluster. Cada gerenciador de filas do cluster deve ter um nome que seja exclusivo em todos os clusters nos quais ele é um membro.

Um gerenciador de filas do cluster pode hospedar filas, que ele anuncia aos outros gerenciadores de filas no cluster. No entanto, ele não tem de fazer isso. Em vez disso, ele pode alimentar mensagens em filas hospedadas em outro lugar no cluster e receber somente respostas que são direcionadas explicitamente a ele.

 No IBM MQ for z/OS, um gerenciador de filas do cluster pode ser um membro de um grupo de filas compartilhadas. Nesse caso, ele compartilha suas definições de fila com outros gerenciadores de filas no mesmo grupo de filas compartilhadas.

Os gerenciadores de filas do cluster são autônomos. Eles possuem controle total sobre filas e canais que eles definem. Suas definições não podem ser modificadas por outros gerenciadores de filas (diferentes dos gerenciadores de filas no mesmo grupo de filas compartilhadas). Os gerenciadores de filas do repositório não controlam as definições em outros gerenciadores de filas no cluster. Elas retêm um conjunto completo de todas as definições, para uso quando necessário. Um cluster é uma federação de gerenciadores de filas.

Após criar ou alterar uma definição em um gerenciador de filas do cluster, as informações serão enviadas para o gerenciador de filas do repositório completo. Outros repositórios no cluster são atualizados posteriormente.

Gerenciador de Filas do Repositório Completo

Um gerenciador de filas do repositório completo é um gerenciador de filas do cluster que contém uma representação completa dos recursos do cluster. Para assegurar a disponibilidade, configure dois ou mais gerenciadores de fila do repositório completo em cada cluster. Os gerenciadores de filas do repositório completo recebem informações enviadas por outros gerenciadores de filas no cluster e atualizam seus repositórios. Eles enviam mensagens uns aos outros para assegurarem-se de que ambos serão mantidos atualizados com as novas informações sobre o cluster.

Repositórios e Gerenciadores de Filas

Cada cluster possui pelo menos um (preferivelmente dois) gerenciador de filas contendo repositórios completos de informações sobre os gerenciadores de filas, filas e canais em um cluster. Estes repositórios também contêm solicitações de outros gerenciadores de filas no cluster de atualizações das informações.

Os outros gerenciadores de filas contêm, cada um, um repositório parcial, contendo informações sobre o subconjunto de filas e gerenciadores de filas com os quais eles precisam se comunicar. Os gerenciadores de filas constroem seus repositórios parciais fazendo consultas quando eles precisam acessar pela primeira vez outra fila ou o gerenciador de filas. Eles solicitam ser notificados de qualquer nova informação concernente a essa fila ou ao gerenciador de filas.

Cada gerenciador de filas armazena suas informações de repositório em mensagens em uma fila chamada `SYSTEM.CLUSTER.REPOSITORY.QUEUE`. Os gerenciadores de filas trocam informações do repositório em mensagens em uma fila chamada `SYSTEM.CLUSTER.COMMAND.QUEUE`.

Cada gerenciador de filas que se associa a um cluster define um canal emissor de cluster, `CLUSSDR`, para um dos repositórios. Ele aprende imediatamente quais outros gerenciadores de filas no cluster contêm repositórios completos. Daí em diante, o gerenciador de filas pode solicitar informações a partir de qualquer um dos repositórios. Quando o gerenciador de filas enviar informações para o repositório escolhido, ele também enviará informações para outro repositório (se houver um).


Um repositório completo será atualizado quando o gerenciador de filas que o hospeda receber novas informações de um dos gerenciadores de filas que estão vinculados a ele. As novas informações também são enviadas a outro repositório, para reduzir o risco delas serem atrasadas se um gerenciador de filas de repositório estiver fora de serviço. Como todas as informações são enviadas duas vezes, os repositórios precisam descartar as duplicatas. Cada item de informações transporta um número de sequência, o qual os repositórios usam para identificar as duplicatas. Todos os repositórios são mantidos em etapa para troca de mensagens.

Filas de Clusters

Uma fila de clusters é uma fila que é hospedada por um gerenciador de filas do cluster e disponibilizada para outros gerenciadores de filas no cluster.

Uma definição de fila de clusters é divulgada para outros gerenciadores de filas no cluster. Os outros gerenciadores de filas no cluster podem colocar mensagens em uma fila de clusters sem a necessidade de uma definição de fila remota correspondente. Uma fila de clusters pode ser divulgada em mais de um cluster usando uma lista de nomes de cluster.

Quando uma fila é divulgada, qualquer gerenciador de filas no cluster pode colocar mensagens nela. Para colocar uma mensagem, o gerenciador de filas deve descobrir, a partir dos repositórios completos, onde a fila está hospedada. Em seguida, ele inclui algumas informações de roteamento na mensagem e coloca a mensagem em uma fila de transmissão do cluster.

 Uma fila de clusters pode ser uma fila que é compartilhada por membros de um grupo de filas compartilhadas no IBM MQ for z/OS.

Tarefas relacionadas

[Definindo filas de clusters](#)

Comparação entre Filas Compartilhadas e Filas de Clusters

Estas informações são projetadas para ajudá-lo a comparar filas compartilhadas e filas de clusters e decidir o que pode ser mais adequado para seu sistema.

Custos do inicializador de canais

Em filas de clusters, as mensagens são enviadas por canais, portanto, permita custos do inicializador de canais, além de custos de aplicativos. Há custos na rede porque os canais recebem e enviam mensagens. Esses custos não estão presentes com filas compartilhadas que, portanto, usam menos energia de processamento do que as filas de clusters ao mover mensagens entre os gerenciadores de filas em um grupo de filas compartilhadas.

Disponibilidade das Mensagens

Quando uma mensagem é colocada em uma fila, as filas de clusters enviam a mensagem para um dos gerenciadores de filas com canais ativos conectados a ele. No gerenciador de filas remotas, se os aplicativos usados para processar as mensagens não estiverem funcionando, as mensagens não serão processadas e aguardarão o aplicativo ser iniciado. Da mesma forma, se um gerenciador de filas tiver sido encerrado, nenhuma mensagem no gerenciador de filas será disponibilizada até que o gerenciador de filas seja reiniciado. Essas instâncias mostram menos disponibilidade de mensagens durante o uso de filas compartilhadas.

Ao usar filas compartilhadas, qualquer aplicativo no grupo de filas compartilhadas pode obter as mensagens que são enviadas. Se você encerrar um gerenciador de filas no grupo de filas compartilhadas, as mensagens estarão disponíveis para os outros gerenciadores de filas, fornecendo disponibilidade de mensagens mais alta do que ao usar filas de clusters.

Capacidade

Um recurso de acoplamento é mais caro do que um disco; portanto, o custo de armazenamento de 1.000.000 mensagens em uma fila local é mais baixo do que o de ter um recurso de acoplamento com capacidade suficiente para armazenar o mesmo número de mensagens.

Enviando para outros Gerenciadores de Filas

As mensagens de fila compartilhada estão disponíveis apenas dentro de um grupo de filas compartilhadas. Se desejar usar um gerenciador de filas fora do grupo de filas compartilhadas, você deverá usar canais. É possível usar armazenamento em cluster para fazer o balanceamento de carga de trabalho entre diversos gerenciadores de filas distribuídos remotos.

Balanceamento de Carga de Trabalho

É possível usar armazenamento em cluster para atribuir um peso a quais canais e gerenciadores de filas obtêm uma proporção das mensagens enviadas. Por exemplo, é possível enviar 60% de mensagens para um gerenciador de filas e 40% de mensagens para outro. Essa instância não depende da capacidade do gerenciador de filas remotas de processar trabalho. O sistema com o primeiro gerenciador de filas pode ficar sobrecarregado e o sistema com o segundo gerenciador de filas pode ficar inativo, mas a maioria das mensagens irá para o primeiro.

Com filas compartilhadas, dois sistemas CICS podem obter mensagens. Se um sistema ficar sobrecarregado, o outro assumirá o controle da maior parte da carga de trabalho.

Canais de cluster

Em cada repositório completo, você define manualmente um canal do receptor de clusters, e um conjunto de canais do emissor de clusters para se conectar a qualquer outro repositório completo no cluster. Ao incluir um repositório parcial, você define manualmente um canal do receptor de clusters, e um canal do emissor de clusters único que se conecta a um dos repositórios completos. Canais do emissor de clusters adicionais são definidos automaticamente pelo cluster quando necessário. Canais do emissor de clusters definidos automaticamente obtêm seus atributos a partir da definição de canal do receptor de clusters correspondente no gerenciador de filas de recebimento.

Canal do Receptor de Clusters: CLUSRCVR

Uma definição de canal CLUSRCVR define o término de um canal no qual um gerenciador de filas do cluster pode receber mensagens de outros gerenciadores de filas no cluster.

Deve-se definir pelo menos um canal CLUSRCVR para cada gerenciador de filas do cluster. Definindo o canal CLUSRCVR, o gerenciador de filas mostra aos outros gerenciadores de filas do cluster que está disponível para receber mensagens.

Uma definição de canal CLUSRCVR também permite que outros gerenciadores de filas definam automaticamente definições do canal do emissor de clusters correspondentes. Consulte a seção [“Canais do emissor de clusters definidos automaticamente”](#) na página 62 deste artigo.

Canal do Emissor de Clusters: CLUSSDR

Você define manualmente um canal CLUSSDR a partir de cada gerenciador de filas de repositório completo para qualquer outro gerenciador de filas de repositório completo no cluster. Todas as atualizações trocadas pelos repositórios completos fluem exclusivamente nesses canais. Definindo manualmente esses canais, você controla a rede de repositórios completos explicitamente.

Ao incluir um gerenciador de filas de repositório parcial em um cluster, você define manualmente um único canal CLUSSDR para se conectar a um dos repositórios completos. Faz pouca diferença qual repositório completo é escolhido, porque, após o contato inicial ter sido feito, objetos do gerenciador de filas do cluster adicionais para seu gerenciador de filas, incluindo canais CLUSSDR, são definidos automaticamente conforme necessário. Isso permite que o gerenciador de filas envie informações do cluster para qualquer repositório completo, e envie mensagens para qualquer gerenciador de filas no cluster.

Conforme está explicado na seção deste artigo, canais do emissor definidos automaticamente são baseados na configuração do canal do receptor de clusters. Portanto, qualquer propriedade de canal que você configure em canais de cluster deve ser configurada de forma idêntica em canais do receptor de clusters CLUSSDR correspondentes ou configuradas somente nos canais do receptor de clusters.

Você deve definir canais CLUSSDR manualmente somente pelos motivos descritos anteriormente. Ou seja, para conectar inicialmente um repositório parcial a um repositório completo ou para conectar dois repositórios completos juntos. Configurar manualmente um canal CLUSSDR que se conecta a um repositório parcial ou a um gerenciador de filas não no cluster, causa a emissão de mensagens de erro como AMQ9427 e AMQ9428. Embora isto, às vezes, possa ser inevitável como uma situação temporária, por exemplo ao modificar o local de um repositório completo, a definição manual deve ser excluída o mais rápido possível.

Canais do emissor de clusters definidos automaticamente

Geralmente, quando você inclui um gerenciador de filas de repositório parcial em um cluster, você define manualmente somente dois canais de cluster no gerenciador de filas:

- Um canal do emissor de clusters (CLUSSDR) para um gerenciador de filas de repositório completo para o cluster.
- Um canal do receptor de cluster (CLUSRCVR).

O canal CLUSSDR que você define permite que o gerenciador de filas faça o contato inicial com o cluster. Depois que o contato inicial foi feito, canais CLUSSDR adicionais são definidos automaticamente pelo cluster quando necessário.

Um canal CLUSSDR definido automaticamente obtém seus atributos da definição de canal CLUSRCVR correspondente no gerenciador de filas de recebimento. Mesmo se houver um canal CLUSSDR definido manualmente, os atributos do canal CLUSSDR definido automaticamente são usados. Suponha, por exemplo, que você define um canal CLUSRCVR sem especificar um número de porta no parâmetro **CONNAME**, e define manualmente um canal CLUSSDR que não especifica um número de porta. Quando o canal CLUSSDR definido automaticamente substitui o definido manualmente, o número da porta (obtido do canal CLUSRCVR) se torna em branco. O número da porta padrão é usado e o canal falha.

Onde houver diferenças de configuração entre um canal CLUSSDR definido manualmente e a definição de canal CLUSRCVR correspondente, algumas diferenças entram em vigor imediatamente (por exemplo, os parâmetros de balanceamento de carga de trabalho) e algumas entram em vigor somente na reinicialização do canal (por exemplo, a configuração do TLS).

Para evitar confusão, tanto quanto possível observe as diretrizes a seguir:

- Defina manualmente os canais CLUSSDR somente para apontar para repositórios completos.

- Se você tiver definido manualmente canais CLUSSDR, configure-os para corresponder de forma idêntica à respectiva definição de canal CLUSRCVR no gerenciador de filas de recebimento.

Consulte também [Trabalhando com canais definidos automaticamente](#).

Conceitos relacionados

[Trabalhando com canais autodefinidos](#)

[Trabalhando com filas de transmissão do cluster e canais do emissor de clusters](#)

Tarefas relacionadas

[Configurando um novo cluster](#)

[Incluindo um Gerenciador de Filas em um Cluster](#)

Tópicos em Cluster

Os tópicos de cluster são tópicos administrativos com o atributo de **cluster** definido. Informações sobre tópicos de cluster são enviadas por push para todos os membros de um cluster, e combinadas com tópicos locais para criar partes de um espaço de tópico que se estende por vários gerenciadores de filas. Isso permite que mensagens publicadas em um tópico em um gerenciador de filas sejam entregues às assinaturas de outros gerenciadores de filas no cluster.

Ao definir um tópico de cluster em um gerenciador de filas, a definição de tópico de cluster é enviada para os gerenciadores de filas do repositório completo. Os repositórios completos, em seguida, propagam a definição de tópico de cluster para todos os gerenciadores de filas no cluster, tornando o mesmo tópico de cluster disponível para publicadores e assinantes em qualquer gerenciador de filas no cluster. O gerenciador de filas no qual você cria um tópico de cluster é conhecido como um host de tópico de cluster. O tópico de cluster pode ser usado por qualquer gerenciador de filas no cluster, mas quaisquer modificações em um tópico de cluster deverão ser feitas no gerenciador de filas no qual esse tópico está definido (o host), nesse ponto a modificação é propagada para todos os membros do cluster por meio dos repositórios completos.

Para obter informações sobre como configurar tópicos de cluster para usar *roteamento direto* ou *roteamento de host de tópico*, e sobre herança de tópico em cluster e assinaturas curingas, consulte [Definindo tópicos de cluster](#).

Para obter informações sobre os comandos a serem usados para exibir tópicos de cluster, consulte as informações relacionadas.

Conceitos relacionados

[Trabalhando com tópicos administrativos](#)

[Trabalhando com assinaturas](#)



Referências relacionadas

[DISPLAYTOPIC](#)

[DISPLAYTPSTATUS](#)

[DISPLAYSUB](#)

Objetos de Cluster Padrão

 Em Multiplataformas, os objetos de cluster padrão são incluídos no conjunto de objetos padrão criado automaticamente quando você define um gerenciador de filas.  No z/OS, as definições de objeto do cluster padrão podem ser localizadas nas amostras de customização.

Nota: É possível alterar as definições de canal padrão da mesma maneira que qualquer outra definição de canal, executando comandos MQSC ou PCF. Não altere as definições de fila padrão, exceto para `SYSTEM.CLUSTER.HISTORY.QUEUE`.

SYSTEM.CLUSTER.COMMAND.QUEUE

Cada gerenciador de filas em um cluster possui uma fila local chamada `SYSTEM.CLUSTER.COMMAND.QUEUE` que é usada para transferir mensagens para o repositório completo. A mensagem contém quaisquer informações novas ou alteradas sobre o gerenciador

de filas ou quaisquer solicitações por informações sobre outros gerenciadores de filas. `SYSTEM.CLUSTER.COMMAND.QUEUE` normalmente está vazia.

SYSTEM.CLUSTER.HISTORY.QUEUE

Cada gerenciador de filas em um cluster tem uma fila local chamada `SYSTEM.CLUSTER.HISTORY.QUEUE`. O `SYSTEM.CLUSTER.HISTORY.QUEUE` é usado para armazenar o histórico de informações do estado do cluster para fins de serviço.

Nas configurações do objeto padrão, `SYSTEM.CLUSTER.HISTORY.QUEUE` é configurado como `PUT (ENABLED)`. Para suprimir a coleção histórica, mude a configuração para `PUT (DISABLED)`.

SYSTEM.CLUSTER.REPOSITORY.QUEUE

Cada gerenciador de filas em um cluster tem uma fila local chamada `SYSTEM.CLUSTER.REPOSITORY.QUEUE`. Esta fila é usada para armazenar todas as informações de repositório completo. Esta fila normalmente não está vazia.

SYSTEM.CLUSTER.TRANSMIT.QUEUE

Cada gerenciador de filas tem uma definição para uma fila local chamada `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. `SYSTEM.CLUSTER.TRANSMIT.QUEUE` é a fila de transmissão padrão para todas as mensagens de todas as filas e gerenciadores de filas que estão nos clusters. É possível mudar a fila de transmissão padrão para cada canal do emissor de clusters para `SYSTEM.CLUSTER.TRANSMIT.ChannelName`, mudando o atributo do gerenciador de filas `DEFCLXQ`. Não é possível excluir `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Ele também é usado para definir verificações de autorização se a fila de transmissão padrão usada é `SYSTEM.CLUSTER.TRANSMIT.QUEUE` ou `SYSTEM.CLUSTER.TRANSMIT.ChannelName`.

SYSTEM.DEF.CLUSRCVR

Cada cluster possui uma definição de canal `CLUSRCVR` padrão denominada `SYSTEM.DEF.CLUSRCVR`. O `SYSTEM.DEF.CLUSRCVR` é usado para fornecer valores padrão para quaisquer atributos que não forem especificados ao criar um canal do receptor de clusters em um gerenciador de filas no cluster.

SYSTEM.DEF.CLUSSDR

Cada cluster possui uma definição de canal `CLUSSDR` padrão chamada `SYSTEM.DEF.CLUSSDR`. `SYSTEM.DEF.CLUSSDR` é usado para fornecer valores padrão para quaisquer atributos que não forem especificados ao criar um canal do emissor de clusters em um gerenciador de filas no cluster.

Conceitos relacionados

[Trabalhando com objetos do cluster padrão](#)

Sistema de Mensagens de Publicação/Assinatura

O sistema de mensagens de publicação/assinatura permite separar o provedor de informações dos consumidores dessas informações. O aplicativo de envio e o aplicativo de recebimento não precisam saber nada um do outro para que as informações sejam enviadas e recebidas.

Antes que um aplicativo IBM MQ ponto a ponto possa enviar uma mensagem para outro aplicativo, ele precisa saber algo sobre esse aplicativo. Por exemplo, precisará saber o nome da fila para a qual enviar as informações, e também pode especificar um nome do gerenciador de filas.

O IBM MQ de publicação/assinatura remove a necessidade do aplicativo saber alguma coisa sobre o aplicativo de destino. Todos os aplicativos de envio precisam fazer o seguinte:

- *Coloque* uma mensagem do IBM MQ que contenha as informações que o aplicativo deseja.
- Designe a mensagem em um tópico que denote o assunto das informações.
- Deixe que o IBM MQ manipule a distribuição dessas informações.

Semelhantemente, o aplicativo de destino não precisa saber nada sobre a origem das informações recebidas.

A figura a seguir mostra o sistema publicar/assinar mais simples. Existe um publicador, um gerenciador de filas e um assinante. Uma assinatura é feita pelo assinante em um gerenciador de filas, uma publicação é enviada do publicador para o gerenciador de filas, e a publicação é então encaminhada pelo gerenciador de filas para o assinante.

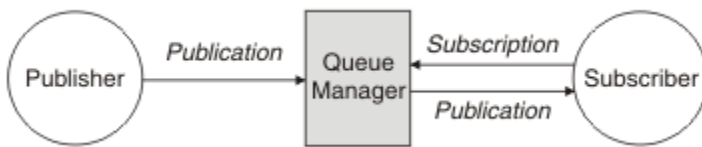


Figura 17. Configuração Simples de Publicação/Assinatura

Um sistema de publicação/assinatura típico tem mais de um publicador e mais de um assinante em muitos tópicos diferentes, e geralmente, mais de um gerenciador de filas. Um aplicativo pode ser tanto um publicador quanto um assinante.

Outra diferença importante entre o sistema de mensagens publicar/assinar e ponto a ponto é que uma mensagem enviada para uma fila ponto a ponto é processada somente por um único aplicativo consumidor. Uma mensagem publicada em um tópico publicar/assinar, no qual mais de um assinante registrou interesse, é processada por cada assinante interessado.

Componentes publicar/assinar

Publicação/Assinatura é o mecanismo pelo qual os assinantes podem receber informações na forma de mensagens, de publicadores. As interações entre os publicadores e assinantes são controladas por gerenciadores de filas, usando os recursos padrão do IBM MQ.

Um sistema de publicação/assinatura típico tem mais de um publicador e mais de um assinante em muitos tópicos diferentes, e geralmente, mais de um gerenciador de filas. Um aplicativo pode ser tanto um publicador quanto um assinante.

O provedor de informações é chamado de *publicador*. Os publicadores fornecem informações sobre um assunto, sem precisar saber nada sobre os aplicativos que estão interessados nessa informação. Os publicadores geram essas informações na forma de mensagens, chamadas *publicações* que desejam publicar e definir o tópico dessas mensagens.

O consumidor das informações é denominado *assinante*. Assinantes criam *assinaturas* que descrevem o tópico em que o assinante está interessado. Assim, a assinatura determina quais publicações são encaminhadas ao assinante. Os assinantes podem fazer várias assinaturas e podem receber informações de vários publicadores diferentes.

Informações publicadas são enviadas em uma mensagem do IBM MQ e o assunto das informações é identificado por seu *tópico*. O publicador especifica o tópico quando ele publica as informações e o assinante especifica os tópicos sobre os quais ele deseja receber publicações. O assinante recebe informações apenas sobre os tópicos que ele assina.

É a existência de tópicos que permite que os provedores e consumidores das informações sejam dissociados no sistema de mensagens de publicação/assinatura, removendo a necessidade de incluir um destino específico em cada mensagem, conforme necessário para o sistema de mensagens ponto a ponto.

As interações entre os publicadores e os assinantes são todas controladas por um gerenciador de filas. O gerenciador de filas recebe mensagens dos publicadores e assinaturas dos assinantes (para um intervalo de tópicos). A tarefa do gerenciador de filas é rotear as mensagens publicadas aos assinantes que registraram interesse no tópico das mensagens.

Os recursos do IBM MQ são usados para distribuir mensagens, portanto, seus aplicativos podem usar todos os recursos que estão disponíveis para aplicativos existentes do IBM MQ. Isso significa que você pode usar mensagens persistentes para obter a entrega única garantida, e que suas mensagens possam ser parte de uma unidade transacional de trabalho para assegurar que as mensagens sejam entregues ao assinante apenas se forem confirmadas pelo publicador.

Publicadores e Publicações

No IBM MQ, um publicador é um aplicativo que disponibiliza informações sobre um tópico especificado disponíveis para um gerenciador de filas na forma de um padrão do IBM MQ mensagem de chamada de publicação. Um publicador pode publicar informações sobre mais de um tópico.

Os publicadores usam o verbo MQPUT para colocar uma mensagem em um tópico aberto anteriormente; esta mensagem é uma publicação. Em seguida, o gerenciador de filas locais roteia a publicação para todos os assinantes que têm assinaturas do tópico da publicação. Uma mensagem publicada pode ser consumida por mais de um assinante.

Além de distribuir publicações para todos os assinantes locais que têm assinaturas apropriadas, um gerenciador de filas também pode distribuir a publicação para quaisquer outros gerenciadores de filas conectados a ele, diretamente ou através de uma rede de gerenciadores de filas que possuem assinantes do tópico.

Em um IBM MQ de rede, um aplicativo de publicação também pode ser um assinante.

Publicações no Ponto de Sincronização

Os publicadores podem emitir chamadas MQPUT ou MQPUT1 no ponto de sincronização para incluir todas as mensagens entregues a assinantes em uma unidade de trabalho. Se a opção MQPMO_RETAIN ou as opções de entrega de tópico NPMMSGDLV ou PMSGDLV com valores ALL ou ALLDUR forem especificadas, o gerenciador de filas usará chamadas MQPUT ou MQPUT1 internas no ponto de sincronização, dentro do escopo da chamada MQPUT ou MQPUT1 do publicador.

Informações de Evento e de Estado

As publicações podem ser categorizadas como publicações de estado, como o preço atual de uma ação ou publicações de evento, como um comércio dessa ação.

Publicações de Estado

Publicações de estado contêm informações sobre o estado atual de algo, como o preço de ações ou a pontuação atual em um jogo de futebol. Quando acontece algo (por exemplo, o preço das ações ou a pontuação da partida de futebol são mudados), as informações de estado anteriores não são mais necessárias porque são substituídas pelas novas informações.

Um assinante vai querer receber a versão atual das informações de estado quando for iniciado, e serão enviadas novas informações sempre que houver mudanças no estado.

Se uma publicação contiver informações sobre estado, ela muitas vezes será publicada como uma publicação retida. Geralmente, um novo assinante deseja as informações de estado atual imediatamente. Ele não quer esperar por um evento que faça com que as informações sejam publicadas novamente. Os assinantes irão receber automaticamente uma publicação retida do tópico quando assinar a menos que o assinante use as opções MQSO_PUBLICATIONS_ON_REQUEST ou MQSO_NEW_PUBLICATIONS_ONLY.

Publicações de Evento

Publicações de evento contêm informações sobre eventos individuais que ocorrem, como o comércio de alguma ação ou a pontuação de um determinado jogo. Cada evento é independente do outro.

Um assinante vai querer receber informações sobre eventos assim que acontecem.

Publicações Retidas

Por padrão, após uma publicação ser enviada a todos os assinantes interessados, ela é descartada. No entanto, um publicador pode especificar que uma cópia de uma publicação seja retida para que ela possa ser enviada aos assinantes futuros que registrarem um interesse no tópico.

A exclusão de publicações depois de terem sido enviadas a todos os assinantes interessados é adequada para informações de evento, mas nem sempre é adequada para informações de estado. Ao reter uma mensagem, novos assinantes não precisam esperar até que as informações sejam publicadas novamente antes que eles recebam as informações de estado iniciais. Por exemplo, um assinante com uma assinatura para um preço de ação receberia o preço atual imediatamente, sem aguardar que o preço da ação fosse mudado (e portanto, republicado).

O gerenciador de filas pode reter apenas uma publicação de cada tópico, portanto, a publicação retida existente de um tópico é excluída quando uma nova publicação retida chega no gerenciador de filas. No entanto, a exclusão da publicação existente não pode ocorrer de maneira síncrona com a chegada da nova publicação retida. Portanto, sempre que possível, não têm mais de um publicador enviando publicações retidas em qualquer tópico.

Os assinantes podem especificar que eles não desejam receber publicações retidas usando a opção de assinatura MQSO_NEW_PUBLICATIONS_ONLY. Os assinantes existentes podem pedir que cópias duplicadas das publicações retidas sejam enviadas a eles.

Há momentos em que você pode não desejar reter publicações, mesmo que para informações de estado:

- Se todas as assinaturas para um tópico forem feitas antes que seja feita qualquer publicação sobre esse tópico, e você não espera ou não permite, novas assinaturas, não haverá necessidade de reter publicações porque elas serão entregues ao conjunto completo de assinantes na primeira vez que forem publicadas.
- Se as publicações ocorrem frequentemente, tal como a cada segundo, um novo assinante (ou um assinante que recupera de uma falha) recebe o estado atual quase imediatamente após a sua primeira assinatura, portanto, não haverá necessidade de reter essas publicações.
- Se as publicações forem grandes, você poderá acabar necessitando de uma quantidade considerável de espaço de armazenamento para armazenar a publicação retida para cada tópico. Em um ambiente do gerenciador de filas múltiplo, as publicações retidas são armazenadas por todos os gerenciadores de filas na rede que possuem uma assinatura correspondente.

Ao decidir se deve usar publicações retidas, considere como os aplicativos de assinatura se recuperam de uma falha. Se o assinante não usar publicações retidas, o aplicativo assinante pode ter que armazenar seu estado atual localmente.

Para assegurar que uma publicação seja retida, use a opção colocar mensagem MQPMO_RETAIN. Se essa opção for usada e a publicação não puder ser retida, a mensagem não será publicada e a chamada falhará com MQRC_PUT_NOT_RETAINED.

Se uma mensagem for uma publicação retida, isso será indicado pela propriedade de mensagem MQIsRetained. A persistência de uma mensagem é conforme ele era quando foi originalmente publicada.

Conceitos relacionados

Considerações de design para publicações retidas em clusters de publicação/assinatura

Publicações no Ponto de Sincronização

Na publicação/assinatura do IBM MQ, o ponto de sincronização pode ser usado por publicadores ou internamente pelo gerenciador de filas.

Publicadores usam sincronização quando emitem chamadas MQPUT/MQPUT1 com a opção MQPMO_SYNCPOINT. Todas as mensagens entregues aos assinantes contam no número máximo de mensagens não confirmadas em uma unidade de trabalho. O atributo do gerenciador de filas MAXUMSGS especifica esse limite. Se o limite for atingido, então o publicador receberá o código de razão 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_REACHED.

Quando um publicador emite chamadas MQPUT/MQPUT1 usando MQPMO_NO_SYNCPOINT com a opção MQPMO_RETAIN ou as opções de entrega de tópico NPMSGDLV/PMSGDLV com valores ALL ou ALLDUR, o gerenciador de filas usa pontos de sincronização internos para garantir que as mensagens sejam entregues conforme solicitado. O publicador poderá receber o código de razão 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_REACHED se o limite for atingido dentro do escopo da chamada do publicador MQPUT/MQPUT1.

Assinantes e assinaturas

Na publicação/assinatura do IBM MQ, um assinante é um aplicativo que solicita informações sobre um tópico específico a partir de um gerenciador de filas em uma rede de publicação/assinatura. Um assinante pode receber mensagens sobre o mesmo tópico ou tópicos diferentes, de mais de um publicador.

As assinaturas podem ser criadas manualmente usando um comando MQSC ou por aplicativos. Essas assinaturas são emitidas para o gerenciador de filas locais e contêm informações sobre as publicações que o assinante deseja receber:

- O tópico em que o assinante está interessado; isto pode ser resolvido para vários tópicos se curingas forem usados.
- Uma cadeia de seleção opcional a ser aplicada a mensagens publicadas.
- Um manipulador para uma fila (conhecida como a *fila de assinante*), na qual as publicações selecionadas devem ser colocada, e o CorrelId opcional.

O gerenciador de filas locais armazena informações de assinatura e quando ele recebe uma publicação, varre as informações para determinar se há uma assinatura que corresponde ao tópico da publicação e sequência de seleção. Para cada assinatura correspondente, o gerenciador de filas direciona a publicação à fila de assinante do assinante. As informações que um gerenciador de filas armazenam sobre assinaturas podem ser visualizadas através do comandos DIS SUB DIS SBSTATUS.

Uma assinatura é excluída apenas quando um dos seguintes eventos ocorre:

- O assinante cancela usando a chamada MQCLOSE (se a assinatura foi feita como não durável).
- A assinatura expira.
- A assinatura é excluída pelo administrador do sistema usando o comando DELETE SUB.
- O aplicativo do assinante termina (se a assinatura foi feita como não durável).
- O gerenciador de filas está interrompido ou reiniciado (se a assinatura foi feita como não durável).

Ao obter mensagens, use opções apropriadas na chamada MQGET. Se o aplicativo processar apenas mensagens para uma assinatura, então, no mínimo, será necessário usar `get-by-correlid`, conforme demonstrado no programa de amostra `C amqssbxa.c` e no assinante MQ não gerenciado. O **CorrelId** a ser usado é retornado de MQSUB no MQSD.campo **SubCorrelId**.

Conceitos relacionados

[Assinaturas clonadas e compartilhadas](#)

Referências relacionadas

[Exemplos de como definir a propriedade sharedSubscription](#)

Filas Gerenciadas e Publicação/Assinatura

Quando criar uma assinatura, você poderá optar por usar o enfileiramento gerenciado. Se você usar o enfileiramento gerenciado, uma fila de assinaturas será criada automaticamente ao criar uma assinatura. As filas gerenciadas são organizadas automaticamente de acordo com a durabilidade da assinatura. Usar filas gerenciadas significa que você não tem que se preocupar com a criação de filas para receber publicações e quaisquer publicações não consumidas são removidas das filas de assinante automaticamente se uma conexão de assinatura não durável for fechada.

Se um aplicativo não tem necessidade de usar uma fila específica como sua fila de assinante, o destino para as publicações que ele recebe, ele pode fazer uso das *assinaturas gerenciadas* usando a opção de assinatura MQSO_MANAGED. Se criar uma assinatura gerenciada, o gerenciador de filas retornará uma manipulação de objetos ao assinante para uma fila de assinantes que o gerenciador de filas cria na qual as publicações serão recebidas. Isso ocorre porque é por meio das *assinaturas gerenciadas* que o IBM MQ manipula a assinatura. A manipulação de objetos da fila será retornada, permitindo procurar, receber ou consultar na fila (não é possível colocar ou configurar atributos de uma fila gerenciada, a menos que você tenha ganho explicitamente o acesso às filas dinâmicas temporárias).

A durabilidade da assinatura determina se a fila gerenciada permanece quando a conexão do aplicativo de assinatura com o gerenciador de filas é interrompida.

As assinaturas gerenciadas são particularmente úteis quando usadas com assinaturas não duráveis, porque quando a conexão do aplicativo é encerrada, as mensagens não consumidas são, em vez disso, permanecer na fila de assinantes, assumindo espaço em seu gerenciador de filas indefinidamente. Se você estiver usando uma assinatura gerenciada, a fila gerenciada será uma fila dinâmica temporária e, como tal, será excluída junto com todas as mensagens não consumidas quando a conexão for interrompida por qualquer um dos seguintes motivos:

- MQCLOSE com MQCO_REMOVE_SUB é usado e o Hobj gerenciado é fechado.
- Uma conexão é perdida para um aplicativo usando uma assinatura não durável (MQSO_NON_DURABLE).
- Uma assinatura é removida porque ela expirou e o Hobj gerenciado é fechado.

As assinaturas gerenciadas também pode ser usadas com assinaturas duráveis, mas é possível que você deseje de deixar as mensagens não consumidas na fila de assinantes, para que eles possam ser recuperados quando a conexão for reaberta. Por essa razão, filas gerenciada para assinaturas duráveis assumem a forma de uma fila dinâmica permanente e permanecem quando a conexão do aplicativo de assinatura com o gerenciador de filas é interrompida.

É possível configurar uma expiração em sua assinatura se desejar usar uma fila gerenciada dinâmica permanente de forma que, embora a fila ainda exista depois que a conexão for interrompida, não continue a existir indefinidamente.

Se excluir a fila gerenciada receberá uma mensagem de erro.

As filas gerenciadas que são criadas são nomeadas com números no final (registros de data e hora), assim, cada uma é exclusiva.

Durabilidade da assinatura

Assinaturas podem ser configuradas para serem duráveis ou não duráveis. A durabilidade da assinatura determina o que acontece com as assinaturas quando os aplicativos de assinatura se desconectam de um gerenciador de filas.

Assinaturas duráveis

As assinaturas duráveis continuarão existindo quando a conexão do aplicativo de assinatura com o gerenciador de filas for encerrada. Se uma assinatura for durável, quando o aplicativo de assinatura for desconectado, a assinatura permanecerá no local e poderá ser usada pelo aplicativo de assinatura quando ele for reconectado solicitando a assinatura novamente usando o **SubName** que foi retornado quando a assinatura foi criada.

Quando assinada como durável, um nome de assinatura (**SubName**) é necessário. Os nomes devem ser exclusivos em um gerenciador de filas para que eles possam ser usados para identificar uma assinatura. Esse meio de identificação é necessário ao especificar uma assinatura que você deseja retomar, se você tiver fechado deliberadamente a conexão com a assinatura (usando a opção MQCO_KEEP_SUB) ou tiver sido desconectado do gerenciador de filas. É possível retomar uma assinatura existente usando a chamada MQSUB com a opção MQSO_RESUME. Os nomes de assinatura também serão exibidos se você usar o comando DISPLAY SBSTATUS com SUBTYPE ALL ou ADMIN

Quando um aplicativo não requer mais uma assinatura durável, ela pode ser removida usando a chamada de função MQCLOSE com a opção MQCO_REMOVE_SUB ou pode ser excluída manualmente usando o comando DELETE SUB MQSC.

É possível usar o atributo de tópico **DURSUB** para especificar se assinaturas duráveis de um tópico podem ou não ser feitas.

No retorno de uma chamada MQSUB usando a opção MQSO_RESUME, a expiração da assinatura é configurada para a expiração original da assinatura e não o tempo de expiração restante.

Um gerenciador de filas continua a enviar publicações para satisfazer uma assinatura durável mesmo se esse aplicativo assinante não está conectado. Isso leva a um acúmulo de mensagens na fila de assinantes. A maneira mais fácil para evitar esse problema é usando uma assinatura não durável sempre que apropriado. Entretanto, em que é necessário usar assinaturas duráveis, um acúmulo de mensagens

pode ser evitado se o assinante assina usando a opção [Publicações retidas](#). Um assinante pode então, controlar quando recebe publicações usando a chamada MQSUBRQ.

Assinaturas Não Duráveis

As assinaturas não duráveis existem apenas enquanto a conexão do aplicativo de assinatura com o gerenciador de filas permanecer aberta. A assinatura é removida quando o aplicativo de assinatura se desconecta do gerenciador de filas deliberadamente ou pela perda de conexão. Quando a conexão é fechada, as informações sobre a assinatura são removidas do gerenciador de filas, e não é mais mostrado se você exibe assinaturas usando o comando DISPLAY SBSTATUS. Mais nenhuma mensagem é colocada na fila de assinantes.

O que acontece com todas as publicações não consumidas na fila de assinantes para assinaturas não duráveis é determinado conforme a seguir.

- Se um aplicativo de assinatura estiver usando um [destino gerenciado](#), quaisquer publicações que não tiverem sido consumidas serão removidas automaticamente.
- Se o aplicativo de assinatura fornece uma manipulação para sua própria fila de assinantes quando o assina, as mensagens não consumidas não são removidas automaticamente. É responsabilidade do aplicativo para limpar a fila, se isso for apropriado. Se a fila é compartilhada por mais de um assinante ou outros aplicativos ponto a ponto, pode não ser apropriado limpar a fila completamente.

Embora não requerido para assinaturas não duráveis, um nome de assinatura, se fornecido, é usado pelo gerenciador de filas. Nomes de assinatura devem ser exclusivos dentro do gerenciador de filas, assim, podem ser usados para identificar uma assinatura.

Conceitos relacionados

[Assinaturas clonadas e compartilhadas](#)

Tarefas relacionadas

[Usando assinaturas compartilhadas do JMS 2.0](#)

Referências relacionadas

[Exemplos de como definir a propriedade sharedSubscription](#)

Sequências de seleção

Uma *sequência de seleção* é uma expressão aplicada a uma publicação para determinar se ela corresponde a uma assinatura. Sequências de seleção podem incluir caracteres curinga.

Ao assinar, além de especificar um tópico, você pode especificar uma sequência de seleção para selecionar as publicações de acordo com suas propriedades de mensagem.

A sequência de seleção é avaliada em relação à mensagem como colocadas pelo publicador antes de ser modificada para entrega para cada assinante. Tome cuidado ao usar campos na sequência de seleção que possam ser modificados como parte da operação de publicação. Por exemplo, os campos MQMD `UserIdentifier`, `MsgId`, e `CorrelId`.

Sequências de seleção não devem fazer referência a campos de propriedade de mensagem incluídos pelo gerenciador de filas como parte da operação de publicação (consulte [Publicar/assinar propriedades de mensagem](#)), exceto para a propriedade de mensagem `MQTopicString`, que contém a sequência de tópicos para a publicação.

Conceitos relacionados

[Regras e restrições de sequência de seleção](#)

tópicos

Um tópico é o assunto das informações que são publicadas em uma mensagem de publicação/assinatura.

Mensagens em sistemas ponto a ponto são enviadas para um endereço de destino específico. Mensagens em sistemas de publicação/assinatura baseados em assunto são enviadas para assinantes com base no assunto que descreve o conteúdo da mensagem. Em sistemas baseados em conteúdo, mensagens são enviadas para assinantes com base no conteúdo da própria mensagem.

O sistema de publicação/assinatura do IBM MQ é um sistema de publicação/assinatura baseado em assunto. Um publicador cria uma mensagem e a publica com a sequência de tópicos que melhor se ajusta ao assunto da publicação. Para receber publicações, um assinante cria uma assinatura com uma sequência de tópicos com correspondência de padrões para selecionar tópicos de publicação. O gerenciador de filas entrega as publicações para os assinantes que têm assinaturas correspondentes ao tópico da publicação e que estão autorizados a receber as publicações. O artigo, [“Sequências de tópicos” na página 71](#), descreve a sintaxe das sequências de tópicos que identificam o assunto de uma publicação. Os assinantes também criam sequências de tópicos para selecionar quais tópicos receber. As sequências de tópicos que os assinantes criam podem conter um dos dois esquemas curinga alternativos para correspondência de padrões com as sequências de tópicos nas publicações. A correspondência de padrão é descrita em [“Esquemas Curinga” na página 72](#).

Na publicação/assinatura baseada em assunto, os publicadores ou administradores, são responsáveis por classificar assuntos nos tópicos. Normalmente os assuntos são organizados hierarquicamente, em árvores de tópicos, usando o caractere ' / ' para criar subtópicos na sequência de tópicos. Consulte [“Árvores de Tópicos” na página 78](#) para obter exemplos de árvores de tópicos. Tópicos são nós na árvore de tópicos. Tópicos podem ser nós folha sem subtópicos ou nós intermediários com subtópicos.

Paralelamente com a organização dos assuntos em uma árvore de tópicos hierárquica, é possível associar tópicos a objetos do tópico administrativo. Você designa atributos a um tópico, como se o tópico é distribuído em um cluster, associando-o a um objeto do tópico administrativo. A associação é feita nomeando o tópico usando o atributo TOPICSTR do objeto do tópico administrativo. Se você não associar explicitamente um objeto do tópico administrativo a um tópico, o tópico herdará os atributos de seu ancestral mais próximo na árvore de tópicos que *foi* associado a um objeto do tópico administrativo. Se você não tiver definido nenhum tópico-pai, ele herdará de SYSTEM.BASE.TOPIC. Objetos de tópico administrativo são descritos em [“Objetos de Tópico Administrativo” na página 79](#).

Nota: Mesmo se você herdar todos os atributos de um tópico de SYSTEM.BASE.TOPIC, defina um tópico raiz para seus tópicos que herde diretamente de SYSTEM.BASE.TOPIC. Por exemplo, no espaço de tópico dos estados americanos, no USA/Alabama USA/Alaska, e assim por diante, o USA é o tópico raiz. O principal propósito do tópico raiz é criar espaços de tópico discretos sem sobreposição para evitar que publicações correspondam às assinaturas erradas. Isso também significa que é possível alterar os atributos do tópico raiz para afetar o espaço de tópico inteiro. Por exemplo, você pode configurar o nome para o atributo **CLUSTER**.

Quando você consulta um tópico como publicador ou assinante, você tem a opção de fornecer uma sequência de tópicos, ou consultar um objeto de tópico. Ou pode-se fazer as duas coisas; nesse caso, a sequência de tópicos fornecida define um subtópico do objeto do tópico. O gerenciador de filas identifica o tópico anexando a sequência de tópicos ao prefixo da sequência de tópicos nomeado no objeto do tópico, inserindo um ' / ' adicional entre as duas sequências de tópicos, por exemplo, *sequência de tópicos/sequência de objetos*. [“Combinando sequências de tópicos” na página 76](#) descreve isso com mais detalhes. A sequência de tópicos resultante é usada para identificar o tópico e associá-lo a um objeto do tópico administrativo. O objeto do tópico administrativo não é necessariamente o mesmo objeto do tópico que o objeto do tópico correspondente ao tópico principal.

Na publicação/assinatura baseada em conteúdo, você define quais mensagens quer receber fornecendo sequências de seleção que procuram o conteúdo de cada mensagem. O IBM MQ fornece um forma intermediária de publicação/assinatura baseada em conteúdo usando seletores de mensagem que varrem propriedades de mensagem em vez de o conteúdo completo da mensagem; consulte [Seletores](#). O uso típico de seletores de mensagem é para se inscrever a um tópico e depois qualificar a seleção com uma propriedade numérica. O seletor permite que você especifique que está interessado nos valores de apenas um certo intervalo, o que não é possível fazer usando curingas baseados em tópicos ou caracteres. Se você precisar filtrar com base no conteúdo integral da mensagem, será necessário usar o IBM Integration Bus.

Sequências de tópicos

Informações de rótulo que você publicar como um tópico usando uma sequência de tópicos. Assine grupos de tópicos usando um caractere curinga ou sequência de tópicos do curinga baseado em tópico.

tópicos

Uma *sequência de tópicos* é uma sequência de caracteres que identifica o tópico de uma mensagem de publicação/assinatura. Você pode usar qualquer caractere que quiser quando você construir uma sequência de tópicos.



Três caracteres têm significado especial em publicar/assinar da IBM WebSphere MQ 7. Eles são permitidos em qualquer lugar em uma sequência de tópicos, mas use-os com cuidado. O uso dos caracteres especiais é explicado em [“Esquema Curinga Baseado em Tópicos”](#) na página 73.

Uma barra (/)

O separador de nível de tópico. Use o caractere ' / ' à estrutura do tópico em uma árvore de tópicos.

Evite níveis de tópico vazio, ' / / ', se você puder. Esses correspondem aos nós na hierarquia de tópicos sem sequência de tópicos. Uma liderança ou trilha ' / ' em uma sequência de tópicos corresponde a um nó vazio de liderança ou de trilha e deve ser evitada.

O sinal de hash (#)

Usado em combinação com ' / ' para construir um curinga de vários níveis em assinaturas. Tome cuidado usando '#' adjacente para ' / ' nas sequências de tópicos usadas para nomear tópicos publicados. [“Exemplos de Sequências de Tópicos”](#) na página 72 mostra um uso sensato de '#'.

As sequências de caracteres '... /#/...', '#/...' e '... /#' possuem um significado especial nas sequências de tópicos de assinatura. As sequências correspondem todos os tópicos em um ou mais níveis na hierarquia de tópicos. Portanto, se você criou um tópico com uma destas sequências, você não poderia assinar a ele, sem também assinar todos os tópicos em diversos níveis na hierarquia de tópicos.

O sinal de mais (+)

Usado em combinação com ' / ' para construir um curinga de único nível em assinaturas. Tome cuidado usando '+' adjacente para ' / ' nas sequências de tópicos usadas para nomear tópicos publicados.

As sequências de caracteres '... /+...', '+/...' e '... /+' possuem um significado especial nas sequências de tópicos de assinatura. As sequências correspondem todos os tópicos em um nível na hierarquia de tópicos. Portanto, se você tiver criado um tópico com uma dessas sequências, não será possível assiná-la, sem também assinar todos os tópicos em um nível na hierarquia de tópicos.

Exemplos de Sequências de Tópicos

```
IBM/Business Area#/Results
IBM/Diversity/%African American
```

Referências relacionadas

[tópico](#)

Esquemas Curinga

Há dois esquemas curingas usados para assinar diversos tópicos. A opção de esquema é uma opção de assinatura.

MQSO_WILDCARD_TOPIC

Selecione os tópicos para assinar usando o esquema curinga baseado em tópico.

Esse será o padrão se nenhum esquema curinga for selecionado explicitamente.

MQSO_WILDCARD_CHAR

Selecione os tópicos para assinar usando o esquema curinga baseado em caracteres.

Configure um esquema especificando o parâmetro **wschema** no comando DEFINE SUB. Para obter mais informações, consulte [DEFINE SUB](#).

Nota: assinaturas que foram criadas antes do IBM WebSphere MQ 7.0 sempre usam o esquema curinga baseado em caracteres.

Exemplos

```
IBM/+/Results
#/Results
IBM/Software/Results
IBM/*ware/Results
```

Esquema Curinga Baseado em Tópicos

Curingas baseados em tópicos permitem que os assinantes assinem mais de um tópico por vez.

Curingas baseados em tópicos são um poderoso recurso do sistema de tópicos na publicação IBM MQ . O curinga de vários níveis e o curinga de nível único podem ser utilizados para assinaturas, mas não podem ser utilizados dentro de um tópico pelo publicador de uma mensagem.

O esquema curinga baseado em tópicos permite que você selecione as publicações agrupadas por nível de tópico. É possível escolher, para cada nível na hierarquia de tópicos, se a sequência na assinatura para esse nível de tópico deve corresponder exatamente à sequência na publicação ou não. Por exemplo, a assinatura IBM/+/Results seleciona todos os tópicos,

```
IBM/Software/Results
IBM/Services/Results
IBM/Hardware/Results
```

Existem dois tipos de curinga.

Curinga de vários níveis

- O curinga de vários níveis é usado em assinaturas. Quando usado em uma publicação, é tratado como um literal.
- O caractere curinga de vários níveis '#' é usado para corresponder a qualquer número de níveis dentro de um tópico. Por exemplo, usando o exemplo da árvore de tópicos, se você se assinar 'USA/Alaska/#', você receberá mensagens sobre os tópicos 'USA/Alaska' e 'USA/Alaska/Juneau'.
- O curinga de vários níveis pode representar zero ou mais níveis. Portanto, 'USA/#' também pode corresponder ao singular 'USA', em que '#' representa níveis zero. O separador de nível de tópico é insignificante neste contexto, pois não há nenhum nível a ser separado.
- O curinga de vários níveis é efetivo apenas quando especificado sozinho ou próximo ao caractere separador de nível de tópico. Portanto, '# ' e 'USA/#' são tópicos válidos onde o caractere '# ' é tratado como um curinga. No entanto, embora 'USA#' também seja uma sequência de tópicos válida, o caractere '# ' não é considerado como um curinga e não tem nenhum significado especial. Consulte a “Quando Curingas Baseados em Tópicos Não São Válidos” na página 75 para obter mais informações.

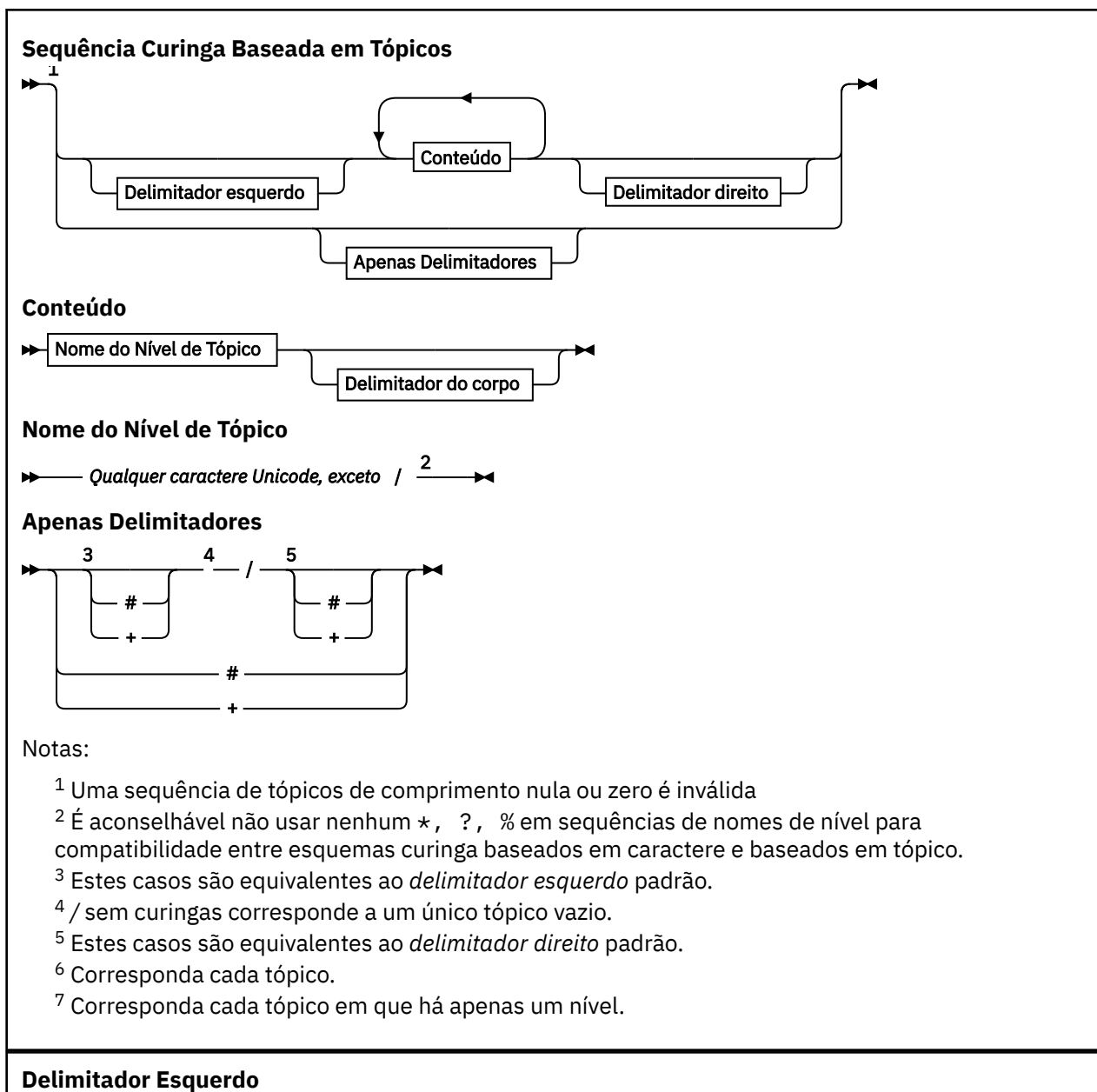
Curinga de nível único

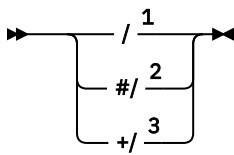
- O curinga único é usado em assinaturas. Quando usado em uma publicação, é tratado como um literal.
- O caractere curinga de nível único '+' corresponde a um, e apenas um nível de tópico. Por exemplo, 'USA/+' corresponde 'USA/Alabama', mas não 'USA/Alabama/Auburn'. Como o curinga de nível único corresponde apenas a um único nível, 'USA/+' não corresponde a 'USA'.

- O curinga de nível único pode ser usado em qualquer nível na árvore de tópicos e em conjunto com o curinga de vários níveis. O curinga de nível único deve ser especificado próximo ao separador de nível de tópico, exceto quando ele for especificado sozinho. Portanto, '+' e 'USA/+' são tópicos válidos onde o caractere '+' é tratado como um curinga. No entanto, embora 'USA+' também seja uma sequência de tópicos válida, o caractere '+' não é considerado como um curinga e não tem nenhum significado especial. Consulte a [“Quando Curingas Baseados em Tópicos Não São Válidos”](#) na página 75 para obter mais informações.

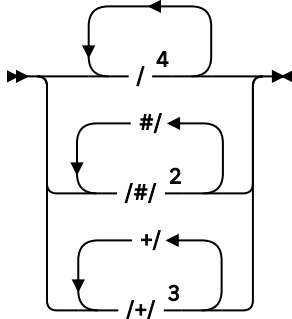
A sintaxe do esquema curinga baseado em tópico não tem caracteres de escape. Se '#' e '+' forem tratados como curingas ou não dependerá de seu contexto. Consulte [“Quando Curingas Baseados em Tópicos Não São Válidos”](#) na página 75 para obter mais informações.

Nota: O início e o fim de uma sequência de tópico são tratados de uma maneira especial. Usando '\$' para denotar o final da cadeia, '\$#/...' é um curinga multinível e '\$/#/...' será um nó vazio na raiz, seguido por um curinga de vários níveis.

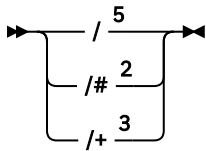




Delimitador de corpo



Delimitador direito



Notas:

- 1 A sequência de tópicos começa com um tópico vazio
- 2 Corresponde a zero ou mais níveis. Diversas sequências de correspondência de vários níveis têm o mesmo efeito como uma sequência de correspondência de vários níveis.
- 3 Corresponde exatamente um nível.
- 4 // é um tópico vazio – um objeto do tópico com nenhuma sequência de tópicos.
- 5 A sequência de tópicos termina com um tópico vazio

Quando Curingas Baseados em Tópicos Não São Válidos

Os caracteres curinga '+' e '#' não possuem significado especial quando são misturados com outros caracteres (incluindo eles próprios) em um nível de tópico.

Isso significa que os tópicos que contêm '+' ou '#' juntos com outros caracteres em um nível de tópico podem ser publicados.

Por exemplo, considere os dois tópicos a seguir:

1. level0/level1/+/level4/#
2. level0/level1/#+/level4/level#

No primeiro exemplo, os caracteres '+' e '#' são tratados como curingas e, portanto, não são válidos em uma sequência de tópicos que deve ser publicada, mas são válidos em uma assinatura.

No segundo exemplo, os caracteres '+' e '#' não são tratados como curingas e, assim, a sequência de tópicos pode ser tanto publicada quanto assinada.

Exemplos

```
IBM/+/Results
#/Results
IBM/Software/Results
```

Esquema Curinga Baseado em Caracteres

O esquema curinga baseado em caracteres permite selecionar os tópicos com base no caractere tradicional correspondente.

É possível selecionar todos os tópicos em diversos níveis em uma hierarquia de tópico usando a sequência '*'. Usar '*' no esquema curinga baseado em caracteres é equivalente a usar a sequência curinga baseada em tópico '#'

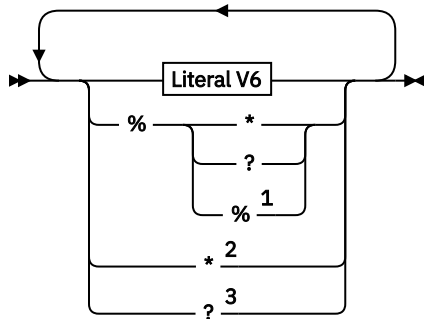
'x*/y' é equivalente a 'x#/y' no esquema baseado em tópico e seleciona todos os tópicos na hierarquia de tópicos entre os níveis 'x' e 'y', em que 'x' e 'y' são nomes de tópico que não estão no conjunto de níveis retornados pelo curinga

O '/+/' no esquema baseado em tópicos não tem um equivalente exato no esquema baseado em caracteres. O 'IBM*/Results' também selecionaria 'IBM/Patents/Software/Results'. Apenas se o conjunto de nomes de tópico em cada nível da hierarquia for exclusivo, será possível construir sempre as consultas com dois esquemas que produzem correspondências idênticas.

Usado de uma maneira geral, '*' e '?' no esquema baseado em caracteres não possuem equivalentes no esquema baseado em tópicos. O esquema baseado em tópico não executa a correspondência parcial usando os curingas. A assinatura 'IBM/*ware/Results' de curinga baseado em caractere não tem nenhum equivalente baseado em tópico.

Nota: As correspondências que usam assinaturas de caractere curinga são mais lentas do que as correspondências que usam assinaturas baseadas em tópico.

Sequência Curinga Baseada em Caractere



Literal V6

► Qualquer caractere Unicode, exceto *,?e% ◄

Notas:

- 1 Significa escapar o caractere seguinte, de forma que ele seja tratado como uma literal. '%' deve ser seguido por '*', '?' ou '%'. Consulte [“Exemplos de Sequências de Tópicos”](#) na página 72.
- 2 Significa corresponder a zero ou mais caracteres em uma assinatura.
- 3 Significa corresponder exatamente a um caractere em uma assinatura.

Exemplos

```
IBM*/Results
IBM/*ware/Results
```

Combinando sequências de tópicos

Ao criar assinaturas, ou abrir tópicos para poder publicar mensagens neles, a sequência de tópicos pode ser formada pela combinação de duas sequências de subtópicos separadas, ou "subtópicos". Um subtópico é fornecido pelo aplicativo ou comando administrativo como uma sequência de tópicos, e o outro é a sequência de tópicos associada a um objeto do tópico. Você pode usar o subtópico como a própria sequência de tópicos ou combiná-los para formar um nome novo de tópico.

Por exemplo, quando você define uma assinatura usando o comando MQSC **DEFINE SUB**, o comando pode ter **TOPICSTR** (sequência de tópicos) ou **TOPICOBJ** (objeto de tópico) como um atributo, ou ambos juntos. Se apenas **TOPICOBJ** for fornecido, a sequência de tópicos associada a esse objeto de tópico será utilizada como a sequência de tópicos. Se apenas **TOPICSTR** for fornecido, ele será utilizado como a sequência de tópicos. Se ambos forem fornecidos, eles serão concatenados para formar uma sequência de tópicos única no formato **TOPICOBJ / TOPICSTR**, em que a sequência de tópicos configurada por **TOPICOBJ** será sempre configurada pela primeira vez e as duas partes da sequência são sempre separadas por um caractere "/".

Da mesma forma, em um programa MQI, o nome integral do tópico será criado por MQOPEN. Ele é composto de dois campos usados em chamadas MQI de publicação/assinatura, na ordem listada:

1. O atributo **TOPICSTR** do objeto de tópico, nomeado no campo **ObjectName**.
2. O parâmetro **ObjectString** define o subtópico fornecido pelo aplicativo.

A sequência de tópicos resultante é retornada no parâmetro **ResObjectString**.

Estes campos são considerados para estarem presentes somente se o primeiro caractere de cada campo não for um caractere em branco ou nulo e o comprimento do campo for maior que zero. Se apenas um dos campos estiver presente, ele será usado sem mudança como o nome do tópico. Se nenhum dos campos tiver um valor, a chamada falhará com o código de razão MQRC_UNKNOWN_OBJECT_NAME, ou MQRC_TOPIC_STRING_ERROR se o nome completo do tópico não for válido.

Se ambos os campos estiverem presentes, um caractere "/" será inserido entre os dois elementos do nome do tópico combinado resultante.

A tabela a seguir mostra exemplos de concatenação de sequência de tópicos:

<i>Tabela 2. Exemplos de Concatenação da Sequência de Tópicos</i>			
TOPICSTR do objeto do tópico	Sequência de tópicos fornecida pelo aplicativo ou comando DEFINE SUB	Nome Completo do Tópico	Comentário
Futebol/Pontuações	' '	Futebol/Pontuações	O TOPICSTR do objeto de tópico é utilizado sozinho.
' '	Futebol/Pontuações	Futebol/Pontuações	O ObjectString/ TOPICSTR é utilizado sozinho.
Futebol	Pontuações	Futebol/Pontuações	Um caractere "/" é incluído no ponto de concatenação.
Futebol	/Pontuações	Futebol//Pontuações	Um 'nó vazio' é gerado entre as duas sequências. Isso é diferente de "Football/ Scores".
/Futebol	Pontuações	/Futebol/Pontuações	O tópico começa com um 'nó vazio'. Isso é diferente de "Football/ Scores".

O caractere "/" é considerado como um caractere especial, fornecendo estrutura para o nome completo do tópico em "Árvores de Tópicos" na página 78. O caractere "/" não deve ser usado por nenhuma outra razão, porque a estrutura da árvore de tópicos será afetada. O tópico "/Football" não é o mesmo que o tópico "Football".

Nota: Se você usar um objeto de tópico ao criar uma assinatura, o valor da sequência de tópicos do objeto do tópico será fixo na assinatura no momento da definição. Qualquer mudança subsequente para o objeto de tópico não afeta a sequência de tópicos para qual a assinatura é definida.

Caracteres curinga em sequência de tópicos

Os seguintes caracteres curinga são caracteres especiais:

- sinal de mais (+)
- sinal de número (#)
- asterisco (*)
- ponto de interrogação (?)

Caracteres curinga possuem significado especial quando utilizados apenas por uma assinatura. Esses caracteres não são considerados inválidos quando utilizados em outro lugar, no entanto, você deve estar seguro de que entende como eles são utilizados e que pode preferir não usar esses caracteres em suas sequências de tópicos ao publicar ou definir objetos do tópico.

Se você publicar em uma sequência de tópicos com # ou + combinados com outros caracteres (incluindo eles mesmos) dentro de um nível de tópico, a sequência de tópicos pode ser assinada com um esquema curinga.

Se você publicar em uma sequência de tópicos com # ou + como o único caractere entre dois caracteres /, a sequência de tópicos não pode ser assinada explicitamente por um aplicativo que usa o esquema de curingas MQSO_WILDCARD_TOPIC. Esta situação resulta no aplicativo obtendo mais publicações do que esperado.

Você não deve utilizar um caractere curinga na sequência de tópicos de um objeto do tópico definido. Se você fizer isso, o caractere será tratado como um caractere literal quando o objeto for utilizado por um publicador e como um caractere curinga quando utilizado por uma assinatura. Isso pode levar a confusões.

Fragmento do Código de Exemplo

Este fragmento de código extraído do exemplo de programa [Exemplo 2: Publicador para um tópico de variável](#), combina um objeto do tópico com uma sequência de tópicos da variável:

```
MQOD td = {MQOD_DEFAULT}; /* Object Descriptor */
td.ObjectType = MQOT_TOPIC; /* Object is a topic */
td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
td.ObjectString.VSPtr = topicString;
td.ObjectString.VSLength = (MQLONG)strlen(topicString);
td.ResObjectString.VSPtr = resTopicStr;
td.ResObjectString.VSBufSize = sizeof(resTopicStr)-1;
MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
```

Árvores de Tópicos

Cada tópico que você definir é um elemento ou nó na árvore de tópicos. A árvore de tópicos pode estar vazia para iniciar ou conter tópicos que foram definidos anteriormente usando comandos MQSC ou PCF. Você pode definir um novo tópico usando os comandos create topic ou especificando o tópico pela primeira vez em uma publicação ou assinatura.

Embora você possa usar qualquer sequência de caracteres para definir uma sequência de tópicos do tópico, é recomendável escolher uma sequência de tópicos que cabe em uma estrutura de árvore hierárquica. Um cuidadoso design das sequências de tópicos e árvores de tópicos pode ajudá-lo com as seguintes operações:

- Assinar vários tópicos.
- Estabelecer políticas de segurança.

Embora você possa construir uma árvore de tópicos em uma estrutura linear, plana, é melhor construir uma árvore de tópicos em uma estrutura hierárquica com um ou mais tópicos de raiz. Para obter mais informações sobre o planejamento de segurança e tópicos, consulte [Publicar/assinar segurança](#).

A [Figura 18 na página 79](#) mostra um exemplo de uma árvore de tópicos com um tópico de raiz.

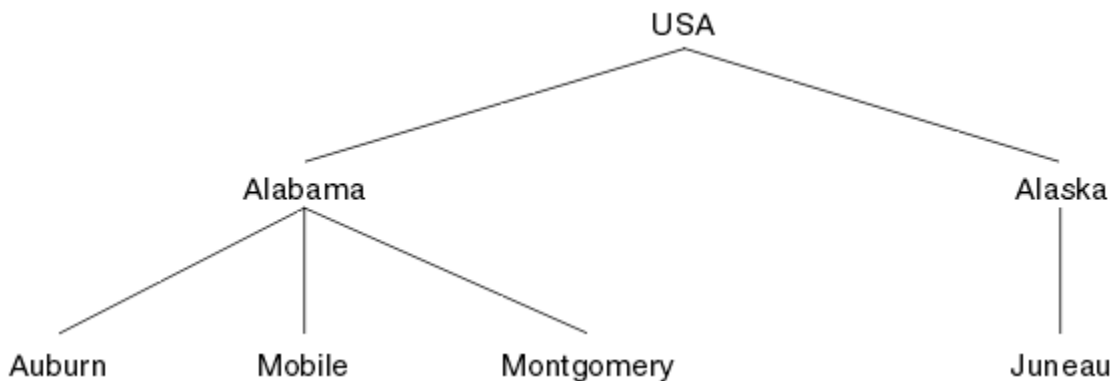


Figura 18. Exemplo de uma Árvore de Tópicos

Cada sequência de caracteres na figura representa um nó na árvore de tópicos. Uma sequência completa de tópicos é criada agregando nós de um ou mais níveis na árvore de tópicos. Os níveis são separados pelo caractere "/". O formato de uma sequência de tópicos totalmente especificada é: "root/level2/level3".

Os tópicos válidos na árvore de tópicos mostrada em [Figura 18 na página 79](#) são:

```
"USA"  
"USA/Alabama"  
"USA/Alaska"  
"USA/Alabama/Auburn"  
"USA/Alabama/Mobile"  
"USA/Alabama/Montgomery"  
"USA/Alaska/Juneau"
```

Ao projetar sequências de tópicos e árvores de tópicos, lembre-se de que o gerenciador de filas não interpreta ou tenta derivar o significado, a própria sequência de tópicos. Ele simplesmente usa a sequência de tópicos para enviar mensagens selecionadas para assinantes desse tópico.

Os seguintes princípios se aplicam à construção e conteúdo de uma árvore de tópicos:

- Não há limite para o número de níveis em uma árvore de tópicos.
- Não há limite para o comprimento do nome de um nível em uma árvore de tópicos.
- Pode haver qualquer quantidade de nós de "raiz"; ou seja, pode haver qualquer quantidade de árvores de tópicos.

Tarefas relacionadas

[Reduzindo o número de tópicos indesejados na árvore de tópicos](#)

Objetos de Tópico Administrativo

Usando um objeto de tópico administrativo, é possível designar atributos específicos não padrão para os tópicos.

[Figura 19 na página 80](#) mostra como um tópico de alto nível de Sport dividido em tópicos separados que abrangem diferentes esportes pode ser visualizado como uma árvore de tópicos:

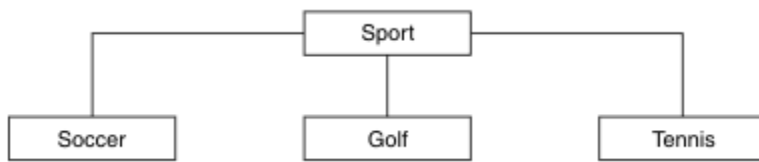


Figura 19. Visualização de uma Árvore de Tópicos

Figura 20 na página 80 mostra como a árvore de tópicos pode ser dividida ainda mais, para separar os diferentes tipos de informações sobre cada esporte:



Figura 20. Árvore de Tópicos Estendida

Para criar a árvore de tópicos ilustrada, nenhum objeto de tópico administrativo precisa ser definido. Cada um dos nós nessa árvore é definido por uma sequência de tópicos criada em uma operação de publicação ou assinatura. Cada tópico na árvore herda seus atributos a partir do respectivo pai. Os atributos são herdados do objeto de tópico-pai, porque, por padrão, todos os atributos são configurados para ASPARENT. Neste exemplo, cada tópico tem os mesmos atributos que o tópico Sport. O tópico Sport não possui nenhum objeto de tópico administrativo e herda seus atributos de `SYSTEM.BASE.TOPIC`.

Observe que não é uma boa prática dar autoridade para usuários que não são mqm no nó de raiz da árvore de tópicos, que é `SYSTEM.BASE.TOPIC`, porque as autoridades são herdadas mas não podem ser restritas. Portanto, ao conceder autoridades neste nível, você estará concedendo autoridades para a árvore inteira. Você deve dar autoridade em um nível de tópico inferior na hierarquia.

Os objetos de tópico administrativo podem ser usados para definir atributos específicos para os nós específicos na árvore de tópicos. No exemplo a seguir, o objeto de tópico administrativo é definido para configurar a propriedade de assinaturas duráveis `DURSUB` do tópico de futebol para o valor `NO`:

```

DEFINE TOPIC(FOOTBALL.EUROPEAN)
TOPICSTR('Sport/Soccer')
DURSUB(NO)
DESCR('Administrative topic object to disallow durable subscriptions')
  
```

A árvore de tópicos agora pode ser visualizada como:

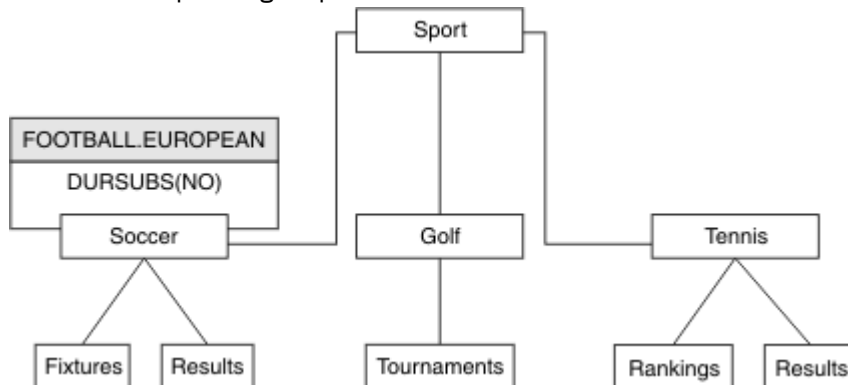


Figura 21. Visualização de um Objeto de Tópico Administrativo Associado ao Tópico Sport/Soccer

Quaisquer aplicativos subscritos para os tópicos sob Soccer na árvore ainda podem usar as sequência de tópicos que usaram antes que o objeto de tópico administrativo fosse incluído. No entanto, um aplicativo agora pode ser gravado para subscrever usando o nome do objeto FOOTBALL . EUROPEAN , em vez da sequência /Sport/Soccer. Por exemplo, para se subscrever em /Sport/Soccer/Results, um aplicativo pode especificar MQSD . ObjectName como FOOTBALL . EUROPEAN e MQSD . ObjectString como Results.

Com este recurso, é possível ocultar parte da árvore de tópicos dos desenvolvedores de aplicativos. Defina um objeto de tópico administrativo em um nó específico na árvore de tópicos e, em seguida, os desenvolvedores de aplicativo podem definir seus próprios tópicos como filhos do nó. Os desenvolvedores devem saber sobre o tópico pai, mas não sobre quaisquer outros nós na árvore pai.

Herdando Atributos

Se uma árvore de tópicos tiver vários objetos de tópico administrativo, cada objeto de tópico administrativo, por padrão, herdará seus atributos de seu tópico administrativo pai mais próximo. O exemplo anterior foi estendido no [Figura 22 na página 81](#):

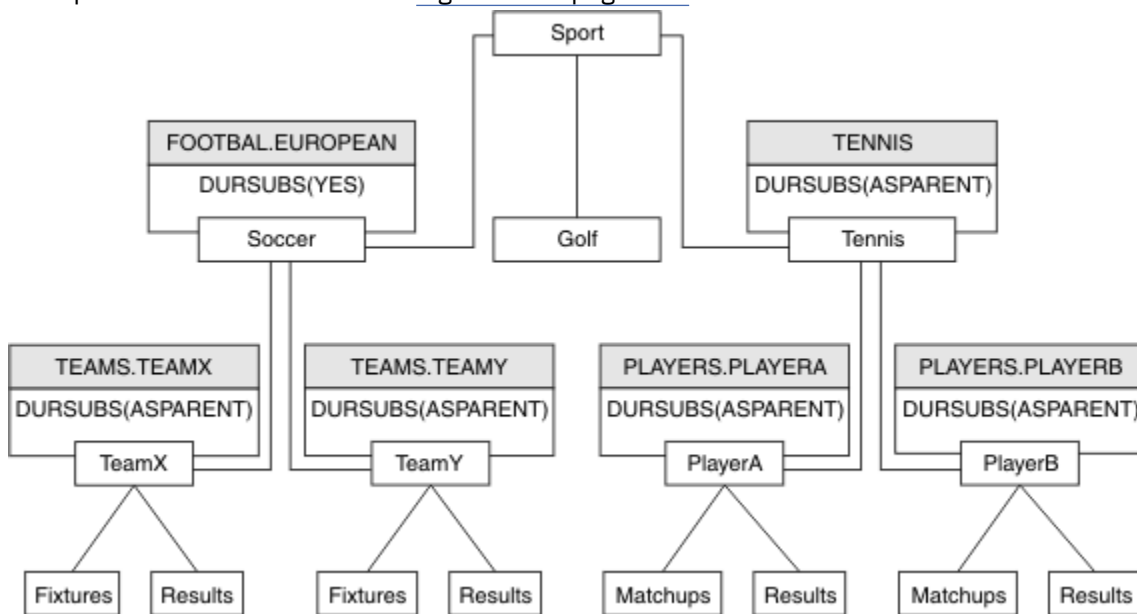


Figura 22. Árvore de Tópicos com Diversos Objetos de Tópico Administrativo

Por exemplo, use a herança para fornecer a todos os tópicos filhos de /Sport/Soccer a propriedade para que as assinaturas sejam não duráveis. Altere o atributo DURSUB de FOOTBALL . EUROPEAN para NO.

Este atributo pode ser configurado usando o seguinte comando:

```
ALTER TOPIC(FOOTBALL.EUROPEAN) DURSUB(NO)
```

Todos os objetos de tópico administrativo de tópicos filhos de Sport/Soccer possuem a propriedade DURSUB configurada para o valor padrão ASPARENT. Após alterar o valor da propriedade DURSUB de FOOTBALL . EUROPEAN para NO, os tópicos filhos de Sport/Soccer herdam o DURSUB valor da propriedade NO. Todos os tópicos filhos de Sport/Tennis herdam o valor de DURSUB do objeto SYSTEM . BASE . TOPIC .. SYSTEM . BASE . TOPIC tem o valor de YES.

Tentar tornar uma assinatura durável para o tópico Sport/Soccer/TeamX/Results falharia agora; no entanto, tentar tornar uma assinatura durável para Sport/Tennis/PlayerB/Results seria bem-sucedido.

Controlando o Uso de Curinga com a Propriedade WILDCARD

Use a propriedade MQSC **Topic WILDCARD** ou a propriedade PCF **Topic WildcardOperation** equivalente para controlar a entrega de publicações para aplicativos assinantes que usam nomes da sequência de tópicos curinga. A propriedade WILDCARD pode ter um dos dois valores possíveis:

CURINGA

O comportamento de assinaturas curingas com relação a este tópico.

PASSTHRU

As assinaturas feitas em um tópico curinga menos específico do que a sequência de tópicos neste objeto do tópico recebem as publicações feitas neste tópico e para sequências de tópicos mais específicas do que este tópico.

BLOCK

As assinaturas feitas em um tópico curinga menos específico do que a sequência de tópicos neste objeto do tópico não recebem as publicações feitas neste tópico ou nas sequências de tópicos mais específicas do que este tópico.

O valor deste atributo é usado quando as assinaturas são definidas. Se você alterar este atributo, o conjunto de tópicos coberto pelas assinaturas existentes não será afetado pela modificação. Este cenário também se aplica se a topologia for alterada quando os objetos do tópico forem criados ou excluídos; o conjunto de tópicos que corresponde às assinaturas criadas seguindo a modificação do atributo WILDCARD é criado usando a topologia modificada. Se você desejar forçar o conjunto de tópicos correspondente para ser reavaliado para assinaturas existentes, deve reiniciar o gerenciador de filas.

No exemplo, “[Exemplo: Criar um Cluster de Publicação/Assinatura Sport](#)” na página 85, é possível seguir as etapas para criar a estrutura da árvore de tópicos mostrada em [Figura 23](#) na página 82.

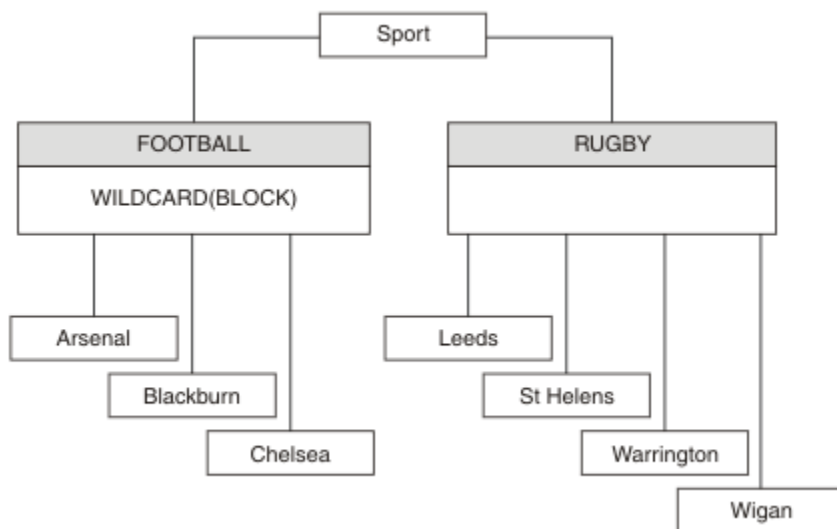


Figura 23. Uma árvore de tópicos que usa a propriedade WILDCARD, BLOCK

Um assinante que usa a sequência de tópicos curinga # recebe todas as publicações para o tópico **Sport** e subárvore **Sport/Rugby**. O assinante não recebe nenhuma publicação para a subárvore **Sport/Football**, porque o valor da propriedade WILDCARD do tópico **Sport/Football** é **BLOCK**.

PASSTHRU é a configuração padrão. É possível configurar o valor da propriedade PASSTHRU de WILDCARD para os nós na árvore **Sport**. Se os nós não tiverem o valor da propriedade **BLOCK** do WILDCARD, configurar PASSTHRU não vai alterar o comportamento observado pelos assinantes para os nós na árvore **Sports**.

No exemplo, crie assinaturas para ver como a configuração curinga afeta as publicações que são entregues; consulte [Figura 27](#) na página 87. Execute o comando de publicação no [Figura 30](#) na página 88 para criar algumas publicações.

Figura 24. Publicar para QMA

Os resultados são mostrados em Tabela 3 na página 83. Observe como a configuração do valor da propriedade BLOCK de WILDCARD, impede que as assinaturas com curingas recebam as publicações para os tópicos no escopo do curinga.

Tabela 3. Publicações Recebidas em QMA

Assinatura	Cadeia do tópico	Publicações Recebidas	Notas
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Todas as publicações para a subárvore Football bloqueadas por WILDCARD (BLOCK) em Sports/ Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) em Sports/ Football impede as assinaturas curingas em Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	WILDCARD padrão em Sports/Rugby não impede a assinatura curinga em Leeds.

Nota:

Imagine que uma assinatura tenha um curinga que corresponda a um objeto do tópico com o valor da propriedade BLOCK de WILDCARD. Se a assinatura também tiver uma sequência de tópicos à direita do curinga correspondente, a assinatura nunca receberá uma publicação. O conjunto de publicações que não estão bloqueadas são publicações para tópicos que são pais do curinga bloqueado. As publicações para os tópicos que são filhos do tópico com o valor da propriedade BLOCK são bloqueadas pelo curinga. Portanto, as sequências de tópicos de assinatura que incluem um tópico à direita do curinga nunca recebem qualquer publicação a ser correspondida.

Configurar o valor da propriedade WILDCARD para BLOCK não significa que você não possa se inscrever usando uma sequência de tópicos que inclua curingas. Essa assinatura é normal. A assinatura tem um tópico explícito que corresponde ao tópico com um objeto do tópico que possui um valor da propriedade BLOCK de WILDCARD. Ela usa os curingas para os tópicos que são pais ou filhos do tópico com o valor da propriedade BLOCK de WILDCARD. No exemplo em [Figura 23 na página 82](#), uma assinatura, como Sports/Football/#, pode receber as publicações.

Curingas e Tópicos do Cluster

As definições do tópico do cluster são propagadas para cada gerenciador de filas em um cluster. Uma assinatura para um tópico de cluster em um gerenciador de filas em um cluster resulta no gerenciador de filas criando assinaturas de proxy. Uma assinatura de proxy é criada em cada outro gerenciador de filas no cluster. As assinaturas que usam sequências de tópicos que contêm curingas, combinados com tópicos de cluster, podem fornecer um comportamento difícil de ser previsto. O comportamento é explicado no seguinte exemplo.

Na configuração do cluster para o exemplo, “Exemplo: Criar um Cluster de Publicação/Assinatura Sport” na página 85, QMB tem o mesmo conjunto de assinaturas que QMA, embora QMB não tenha recebido nenhuma publicação após o publicador publicado para QMA, consulte [Figura 24 na página 83](#). Embora os tópicos Sports/Football e Sports/Rugby sejam tópicos de cluster, as assinaturas definidas em fullsubs.tst não fazem referência a um tópico de cluster. Nenhuma assinatura de proxy é propagada de QMB para QMA. Sem as assinaturas de proxy, nenhuma publicação para QMA é encaminhada para QMB.

Algumas das assinaturas, como Sports/#/Leeds, podem parecer fazer referência a um tópico do cluster, neste caso, Sports/Rugby. A assinatura Sports/#/Leeds resolve, na realidade, para o objeto de tópico SYSTEM.BASE.TOPIC.

A regra para resolver o objeto do tópico referenciado por uma assinatura, como Sports/#/Leeds, é a seguinte. Truncar a sequência de tópicos para o primeiro curinga. Varra para a esquerda da sequência de tópicos procurando pelo primeiro tópico que tem um objeto de tópico administrativo associado. O objeto do tópico pode especificar um nome de cluster ou definir um objeto de tópico local. No exemplo, Sports/#/Leeds, a sequência de tópicos após o truncamento é Sports, que não possui nenhum objeto do tópico e, portanto, Sports/#/Leeds herda de SYSTEM.BASE.TOPIC, que é um objeto do tópico local.

Para ver como a assinatura nos tópicos em cluster pode alterar a maneira como a propagação de curinga funciona, execute o script em lote, upsubs.bat. O script limpa as filas de assinatura e inclui as assinaturas de tópico de cluster em fullsubs.tst. Execute puba.bat novamente para criar um lote de publicações; consulte [Figura 24 na página 83](#).

[Tabela 4 na página 84](#) mostra o resultado de incluir duas novas assinaturas no mesmo gerenciador de filas no qual as publicações foram publicadas. O resultado é o esperado, as novas assinaturas recebem uma publicação cada e os números de publicações recebidas pelas outras assinaturas ficam inalterados. Ocorrem os resultados inesperados no outro gerenciador de filas do cluster; consulte [Tabela 5 na página 85](#).

Assinatura	Cadeia do tópico	Publicações Recebidas	Notas
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Todas as publicações para a subárvore Football bloqueadas por WILDCARD (BLOCK) em Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) em Sports/Football impede as assinaturas curingas em Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	WILDCARD padrão em Sports/Rugby não impede a assinatura curinga em Leeds.
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal recebe uma publicação porque a assinatura não tem um curinga.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds receberia uma publicação em qualquer evento.

[Tabela 5 na página 85](#) mostra os resultados de incluir as duas novas assinaturas no QMB e publicar em QMA. A rechamada desse QMB não recebeu nenhuma publicação sem essas duas novas assinaturas. Conforme esperado, as duas novas assinaturas recebem as publicações, porque Sports/FootBall e Sports/Rugby são tópicos de cluster. O QMB encaminhou assinaturas de proxy para Sports/Football/Arsenal e Sports/Rugby/Leeds para QMA, que então enviou as publicações para QMB.

O resultado inesperado é que duas assinaturas Sports/# e Sports/#/Leeds que não receberam anteriormente nenhuma publicação, agora recebem as publicações. A razão é que as publicações Sports/Football/Arsenal e Sports/Rugby/Leeds encaminhadas para QMB para as outras assinaturas agora estão disponíveis para qualquer assinante conectado ao QMB. Conseqüentemente as assinaturas para os tópicos locais Sports/# e Sports/#/Leeds recebem a publicação Sports/Rugby/Leeds. Sports/#/Arsenal continua não recebendo uma publicação, porque Sports/Football tem seu valor de propriedade WILDCARD configurado como BLOCK.

<i>Tabela 5. Publicações Recebidas em QMB</i>			
Assinatura	Cadeia do tópico	Publicações Recebidas	Notas
SPORTS	<i>Sports/#</i>	<i>Sports/Rugby/Leeds</i>	<i>Todas as publicações para a subárvore Football bloqueadas por WILDCARD (BLOCK) em Sports/Football</i>
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) em Sports/Football impede as assinaturas curingas em Arsenal
SLEEDS	<i>Sports/#/Leeds</i>	<i>Sports/Rugby/Leeds</i>	<i>WILDCARD padrão no Sports/Rugby não evita a assinatura curinga em Leeds.</i>
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal recebe uma publicação porque a assinatura não tem um curinga.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds receberia uma publicação em qualquer evento.

Na maioria dos aplicativos, é desejável para uma assinatura influenciar o comportamento de outra assinatura. Um uso importante da propriedade WILDCARD com valor BLOCK é fazer com que as assinaturas para a mesma sequência de tópicos que contém curingas se comportem uniformemente. Indica se a assinatura está no mesmo gerenciador de filas que o publicador ou um diferente, os resultados da assinatura são os mesmos.

Curingas e Fluxos

Para um novo aplicativo gravado para a API de publicação/assinatura, o efeito é que uma assinatura para * não recebe nenhuma publicação. Para receber todas as publicações de Sports você deve se inscrever a Sports/* ou Sports/#, de maneira semelhante para publicações Business.

O comportamento de um aplicativo de publicação / assinatura enfileirado existente não muda quando o broker de publicação / assinatura é migrado para uma versão posterior do IBM MQ. A propriedade **StreamName** nos comandos **Publish**, **Register Publisher** ou **Subscriber** é mapeada para o nome do tópico para o qual o fluxo foi migrado..

Curingas e Pontos de Assinatura

Para um novo aplicativo gravado para a API de publicação/assinatura, o efeito da migração é que uma assinatura para * não receba nenhuma publicação. Para receber todas as publicações de Sports você deve se inscrever a Sports/* ou Sports/#, de maneira semelhante para publicações Business.

O comportamento de um aplicativo de publicação / assinatura enfileirado existente não muda quando o broker de publicação / assinatura é migrado para uma versão posterior do IBM MQ. A propriedade **SubPoint** nos comandos **Publish**, **Register Publisher** ou **Subscriber** é mapeada para o nome do tópico para o qual a assinatura foi migrada..

Exemplo: Criar um Cluster de Publicação/Assinatura Sport

As etapas que seguem a criação de um cluster, CL1, com quatro gerenciadores de filas: dois repositórios completos, CL1A e CL1B, e dois repositórios parciais, QMA e QMB. Os repositórios completos são usados para reter apenas as definições de cluster. QMA designado ao host de tópico do cluster. As assinaturas duráveis são definidas em QMA e QMB.

Nota: O exemplo é codificado para Windows. Você deve recodificar [Create qmgrs.bat](#) e [create pub.bat](#) para configurar e testar o exemplo em outras plataformas.

1. Crie os arquivos de script.
 - a. [Create topics.tst](#)
 - b. [Create wildsubs.tst](#)
 - c. [Create fullsubs.tst](#)
 - d. [Create qmgrs.bat](#)
 - e. [create pub.bat](#)
2. Execute [Crie qmgrs.bat](#) para criar a configuração.

```
qmgrs
```

Crie os tópicos em [Figura 23 na página 82](#). O script na figura 5 cria os tópicos de cluster Sports/Football e Sports/Rugby

Nota: A opção REPLACE não substitui as propriedades TOPICSTR de um tópico. TOPICSTR é uma propriedade que é proveitosamente variada no exemplo para testar as diferentes árvores de tópicos. Para alterar os tópicos, exclua primeiro o tópico.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Figura 25. Excluir e Criar os Tópicos: topics.tst

Nota: Exclua os tópicos, porque o REPLACE não substitui as sequências de tópicos.

Crie assinaturas com curingas. Os curingas correspondentes dos tópicos com objetos do tópico no [Figura 23 na página 82](#). Crie uma fila para cada assinatura. As filas são limpas e as assinaturas excluídas quando o script não é executado ou reexecutado.

Nota: A opção REPLACE não substitui as propriedades TOPICOBJ ou TOPICSTR de uma assinatura. TOPICOBJ ou TOPICSTR são propriedades proveitosamente variadas no exemplo para testar diferentes assinaturas. Para alterá-las, exclua primeiro a assinatura.

```

DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)

```

Figura 26. Criar Assinaturas Curinga: wildsubs.tst

Crie as assinaturas que fazem referência aos objetos do tópico de cluster.

Nota:

O delimitador, /, é inserido automaticamente entre a sequência de tópicos referenciada por TOPICOBJ e a sequência de tópicos definida por TOPICSTR.

A definição, DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) cria a mesma assinatura. TOPICOBJ é usado como uma maneira rápida de fazer referência à sequência de tópicos que você já definiu. A assinatura, quando criada, não se refere mais ao objeto do tópico.

```

DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)

```

Figura 27. Excluir e Criar Assinaturas: fullsubs.tst

Crie um cluster com dois repositórios. Crie dois repositórios parciais para publicar e subscrever. Execute novamente o script para excluir tudo e inicie novamente. O script também cria a hierarquia de tópico e as assinaturas curingas iniciais.

Nota:

Em outras plataformas, grave um script semelhante ou digite todos os comandos. Usar um script facilita a exclusão de tudo e inicia novamente com uma configuração idêntica.

```

@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof

```

Figura 28. Criar Gerenciadores de Filas: *qmgrs.bat*

Atualize a configuração incluindo as assinaturas nos tópicos do cluster.

```

@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst

```

Figura 29. Atualizar Assinaturas: *upsubs.bat*

Execute *pub.bat*, com um gerenciador de filas como um parâmetro, para publicar as mensagens que contêm a sequência de tópicos de publicação. *Pub.bat* usa o programa de amostra **amqspub**.

```

@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1

```

Figura 30. Publicar: *pub.bat*

Fluxos e Tópicos

A publicação/assinatura enfileirada tem o conceito de um fluxo de publicação que não existe no modelo de publicação/assinatura integrada. Na publicação/assinatura enfileirada, os fluxos fornecem uma maneira de separar o fluxo de informações para os diferentes tópicos. Um fluxo é implementado como um tópico de nível superior que pode ser mapeado para um identificador de tópico diferente administrativamente.

O fluxo padrão `SYSTEM.BROKER.DEFAULT.STREAM` é configurado automaticamente para todos os corretores e gerenciadores de filas em uma rede e nenhuma configuração adicional é necessária para usar o fluxo padrão. Pense no fluxo padrão como um espaço de tópico padrão sem nome. Os tópicos publicados para o fluxo padrão estão imediatamente disponíveis para todos os gerenciadores de filas conectados, com a publicação / assinatura em fila ativada. Os fluxos nomeados são como espaços de tópico, nomeados, separados. O fluxo nomeado deve ser definido em cada broker no qual é usado.

Se os publicadores e assinantes estiverem em diferentes gerenciadores de fila, depois que os corretores forem conectados na mesma hierarquia do corretor, nenhuma configuração adicional será necessária para as publicações e assinaturas fluírem entre eles. A mesma interoperabilidade funciona em reverso, também.

Fluxos Nomeados

Um designer de solução trabalhando com o modelo de programação de publicação/assinatura enfileirada, pode decidir colocar todas as publicações de esportes em um fluxo nomeado denominado Sport. Para que o fluxo esteja disponível para um gerenciador de filas que é executado no IBM MQ com a publicação / assinatura em fila ativada, o fluxo deve ser incluído manualmente.

Aplicativos de publicação/assinatura enfileirada que assinam Soccer/Results no fluxo Sport trabalham sem mudança. Aplicativos de publicação / assinatura integrados que assinam o tópico Sport usando MQSUBe fornecendo a sequência de tópicos Soccer/Results também recebem as mesmas publicações.

A tarefa de incluir um fluxo está descrita no tópico [Incluindo um Fluxo](#). Pode ser necessário incluir fluxos manualmente por duas razões.

1. Continue a desenvolver seus aplicativos de publicação/assinatura enfileirada que estão executando em gerenciadores de filas de versões mais recentes, ao invés de fazer a migração de aplicativos para a interface MQI de publicação/assinatura integrada.
2. O mapeamento padrão de fluxos para os tópicos conduz a uma "colisão" no espaço de tópico e as publicações em um fluxo possuem a mesma sequência de tópicos que as publicações de algum lugar.

Autoridade

Por padrão, na raiz da árvore de tópicos existe diversos objetos do tópico: SYSTEM.BASE.TOPIC, SYSTEM.BROKER.DEFAULT.STREAM e SYSTEM.BROKER.DEFAULT.SUBPOINT. Autoridades (por exemplo, para publicação ou assinatura) são determinadas pelas autoridades no SYSTEM.BASE.TOPIC; Quaisquer autoridades no SYSTEM.BROKER.DEFAULT.STREAM ou SYSTEM.BROKER.DEFAULT.SUBPOINT são ignorados Se um dos SYSTEM.BROKER.DEFAULT.STREAM ou SYSTEM.BROKER.DEFAULT.SUBPOINT for excluído e recriado com uma sequência de tópicos não vazia, as autoridades definidas nesses objetos serão usadas da mesma maneira que um objeto de tópico normal..

Mapeando Entre Fluxos e Tópicos

Um fluxo de publicação / assinatura enfileirado é imitado no IBM MQ criando uma fila e fornecendo o mesmo nome que o fluxo. Às vezes, a fila é chamada de fila de fluxo, porque é como aparece para aplicativos de publicação/assinatura enfileirada. A fila é identificada para o mecanismo publicar/assinar, incluindo-a na lista de nomes especial denominada SYSTEM.QPUBSUB.QUEUE.NAMELIST. É possível incluir quantos fluxos precisar, incluindo as filas adicionais especiais na lista de nomes. Finalmente, você precisa incluir os tópicos, com os mesmos nomes que os fluxos e as mesmas sequências de tópicos que o nome do fluxo, de modo que você possa publicar e subscrever-se nos tópicos.

No entanto, em circunstâncias excepcionais, é possível fornecer aos tópicos correspondentes aos fluxos quaisquer sequências de tópicos que você escolher quando definir os tópicos. O propósito da sequência de tópicos é fornecer ao tópico um nome exclusivo no espaço de tópico. Geralmente, o nome do fluxo serve a esse propósito perfeitamente. Às vezes, um nome do fluxo e um nome de tópico existente se conflitam. Para resolver o problema, escolha outra sequência de tópicos para o tópico associado ao fluxo.. Escolha qualquer sequência de tópico, assegurando que ele seja exclusivo.

A sequência de tópicos definida na definição de tópicos recebe o prefixo da maneira normal para a sequência de tópicos fornecida pelos publicadores e assinantes usando as chamadas MQOPEN ou MQSUB MQI. Os aplicativos que se referem aos tópicos usando os objetos de tópico não são afetados pela opção da sequência de tópicos do prefixo - que é o motivo pelo qual é possível escolher qualquer sequência de tópicos que mantenha as publicações exclusivas no espaço de tópico.

O remapeamento de diferentes fluxos em diferentes tópicos depende dos prefixos usados para as sequências de tópicos serem exclusivos, para separar um conjunto de tópicos completamente do outro. Você deve definir uma convenção de nomenclatura de tópico universal que esteja rigorosamente junta para que o mapeamento funcione.

No IBM MQ, você usa o mecanismo de prefixação para remapear uma sequência de tópicos para outro local no espaço de tópico

Nota: Ao excluir um fluxo, exclua todas as assinaturas no fluxo primeiro. Essa ação é mais importante se alguma dessas assinaturas se originarem de outros corretores na hierarquia do corretor.

Tópicos e Pontos de Assinatura

Pontos de assinatura nomeados são emulados por tópicos e objetos de tópico.

Para incluir pontos de assinatura manualmente, consulte [Incluindo um ponto de assinatura](#).

Pontos de assinatura no IBM MQ

O IBM MQ mapeia pontos de assinatura para diferentes espaços de tópico na árvore de tópicos do IBM MQ. Os tópicos nas mensagens de comando sem um ponto de assinatura são mapeados inalterados para a raiz da árvore de tópicos do IBM MQ e herdam as propriedades do SYSTEM.BASE.TOPIC.

As mensagens de comando com um ponto de assinatura estão em processamento usando a lista de objetos de tópico no SYSTEM.QPUBSUB.SUBPOINT.NAMELIST. O nome do ponto de assinatura na mensagem de comando é correspondida em uma sequência de tópicos para cada um dos objetos de tópico na lista. Se uma correspondência for localizada, o nome do ponto de assinatura será pré-anexado, como um nó de tópico, para a sequência de tópicos. O tópico herda suas propriedades a partir do objeto do tópico associado localizado no SYSTEM.QPUBSUB.SUBPOINT.NAMELIST.

O efeito de usar os pontos de assinatura é criar um espaço de tópico separado para cada ponto de assinatura. O espaço de tópico é enraizado em um tópico que tem o mesmo nome que o ponto de assinatura. Os tópicos em cada espaço de tópico herda suas propriedades a partir do objeto de tópico com o mesmo nome que o ponto de assinatura.

Quaisquer propriedades não definidas no objeto do tópico correspondente são herdadas normalmente do SYSTEM.BASE.TOPIC.

Os aplicativos de publicação/assinatura enfileirados, usando os cabeçalhos da mensagem MQRFH2, continuam trabalhando configurando a propriedade **SubPoint** nas mensagens de comando Publish ou Register subscriber. O ponto de assinatura é combinado com a sequência de tópicos na mensagem de comando e o tópico resultante processado como qualquer outro.

Aplicativos IBM MQ não são afetados por pontos de assinatura. Se um aplicativo usar um tópico que herda informações de um dos objetos do tópico correspondentes, esse aplicativo irá interoperar com um aplicativo enfileirado usando o ponto de assinatura correspondente.

exemplo

Um aplicativo de publicação/assinatura do WebSphere Message Broker existente (agora conhecido como IBM Integration Bus) que foi migrado para IBM MQ criou dois objetos do tópico, GBP e USD, com as sequências de tópicos correspondentes 'GBP' e 'USD'.

Os editores existentes para o tópico NYSE/IBM/SPOT, migrados para execução no IBM MQ e que usam o ponto de assinatura USD, criam publicações sobre o tópico USD/NYSE/IBM/SPOT. Da mesma forma, os assinantes existentes do NYSE/IBM/SPOT, que usam o ponto de assinatura do USD, criam assinaturas para o USD/NYSE/IBM/SPOT.

Assine o preço spot em dólar em um programa de publicação / assinatura IBM MQ chamando MQSUB. Crie uma assinatura usando o objeto tópico do USD e a sequência de tópicos do 'NYSE/IBM/SPOT', conforme ilustrado no fragmento de código 'C'.

```
strncpy(sd.ObjectName, "USD", MQ_TOPIC_NAME_LENGTH);
```

```
sd.ObjectString.VSPtr = "NYSE/IBM/SPOT";  
sd.ObjectString.VSLength = MQVS_NULL_TERMINATED;  
MQSUB(Hconn, &sd, &Hobj, &Hsub, &CompCode, &Reason);
```

1. Configure o atributo CLUSTER dos objetos de tópico USD e GBP no host do tópico do cluster.
2. Exclua todas as cópias dos objetos do tópico USD e GBP em outros gerenciadores de filas no cluster.
3. Certifique-se de que o USD e o GBP estejam definidos no SYSTEM.QPUBSUB.SUBPOINT.NAMELIST em cada gerenciador de filas no cluster.

Exemplo da Configuração de um Único Gerenciador de Filas de Publicação/Assinatura

Figura 31 na página 91 ilustra uma configuração de publicação/assinatura do gerenciador de filas único e básico. O exemplo mostra a configuração para um serviço de notícias, onde as informações estão disponíveis a partir de publicadores sobre vários tópicos:

- Publicador 1 está publicando informações sobre resultados de esportes usando um tópico de Esporte
- Publicador 2 está publicando informações sobre os preços de ações usando um tópico de Estoque
- Publicador 3 está publicando informações sobre revisões de filme usando um tópico de Filmes, e sobre listagens de televisão usando um tópico de TV

Três assinantes registraram interesse em tópicos diferentes, portanto, o gerenciador de filas os envia as informações em que estão interessados:

- Assinante 1 recebe os resultados de esporte e os preços de ação
- Assinante 2 recebe as revisões de filmes
- Assinante 3 recebe os resultados de esporte

Nenhum dos assinantes registraram interesse nas listagens de televisão, portanto elas não são distribuídas.

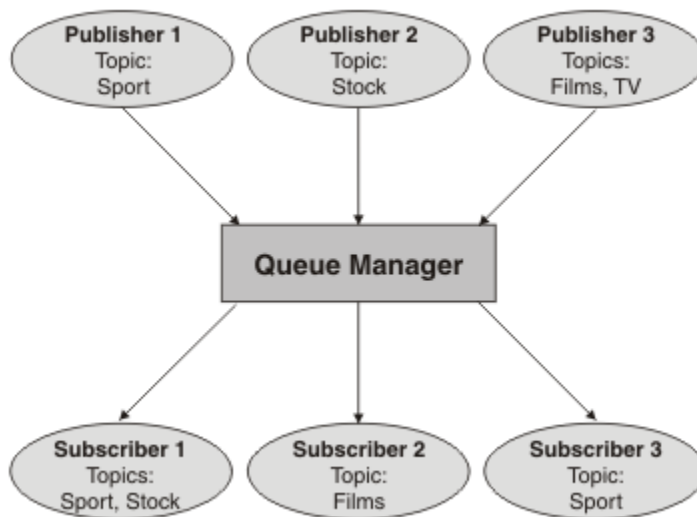


Figura 31. Exemplo de Gerenciador de Filas Único de Publicação/Assinatura

Redes publicar/assinar distribuídas

Cada gerenciador de filas corresponde mensagens publicadas em um tópico com as assinaturas criadas localmente que assinaram esse tópico. É possível configurar uma rede de gerenciadores de filas para que as mensagens publicadas por um aplicativo conectado a um gerenciador de filas sejam entregues a assinaturas correspondentes criadas em outros gerenciadores de filas na rede. Isso requer configuração adicional sobre canais simples entre gerenciadores de filas.

Uma configuração publicar/assinar distribuída é um conjunto de gerenciadores de filas conectados entre si. Todos os gerenciadores de filas podem estar no mesmo sistema físico ou podem ser distribuídos entre vários sistemas físicos. Ao conectar gerenciadores de filas, os assinantes podem assinar um gerenciador de filas e receber mensagens que foram inicialmente publicadas em outro gerenciador de filas. Para ilustrar isso, a figura a seguir inclui um segundo gerenciador de filas na configuração descrita em “Exemplo da Configuração de um Único Gerenciador de Filas de Publicação/Assinatura” na página 91.

- Gerenciador de filas 2 é usado pelo Publicador 4 para publicar as informações da previsão do tempo climático, usando um tópico de Tempo Climático e as informações sobre as condições de tráfego nas principais rodovias, usando um tópico de Tráfego.
- Assinante 4 também usa este gerenciador de filas e subscreve às informações sobre as condições de tráfego usando o tópico Tráfego.
- Assinante 3 também assina às informações sobre as condições climáticas, mesmo se usar um gerenciador de filas diferente do publicador. Isso é possível porque os gerenciadores de filas estão vinculados entre si.

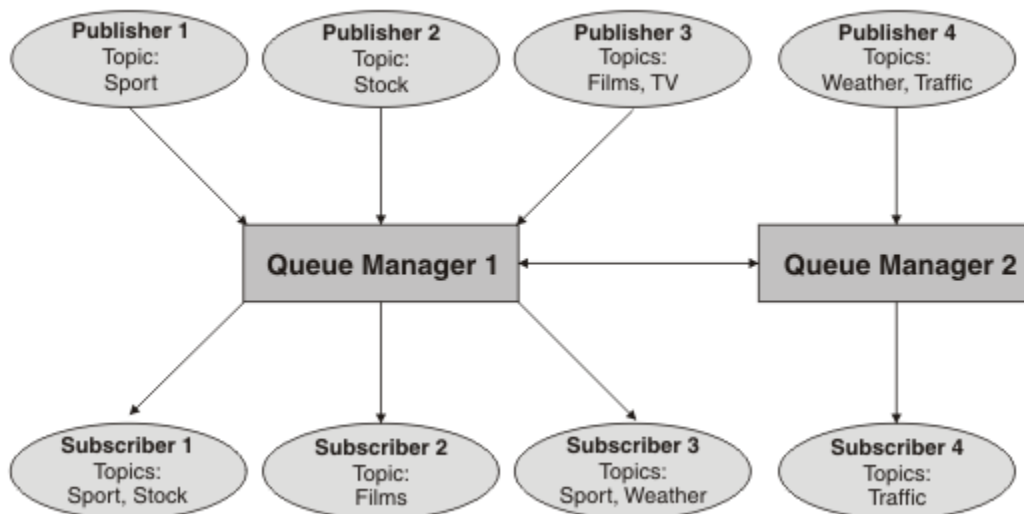


Figura 32. Exemplo de Publicação/Assinatura com Dois Gerenciadores de Filas

É possível conectar gerenciadores de filas manualmente em uma hierarquia de pai e filho ou criar um cluster de publicação/assinatura e deixar que o IBM MQ defina a maioria dos detalhes da conexão para você. Também é possível usar duas topologias em combinação, por exemplo, juntando vários clusters em uma hierarquia.

Visão geral de clusters publicar/assinar

Um cluster publicar/assinar é um cluster padrão com um ou mais objetos de tópico incluídos no cluster. Ao definir um objeto do tópico administrativo em qualquer gerenciador de filas em um cluster e tornar esse objeto do tópico em cluster especificando um nome de cluster, os publicadores e assinantes desse tópico podem conectar-se a qualquer um dos gerenciadores de filas no cluster e as mensagens publicadas são roteadas para os assinantes em canais de cluster entre gerenciadores de filas.

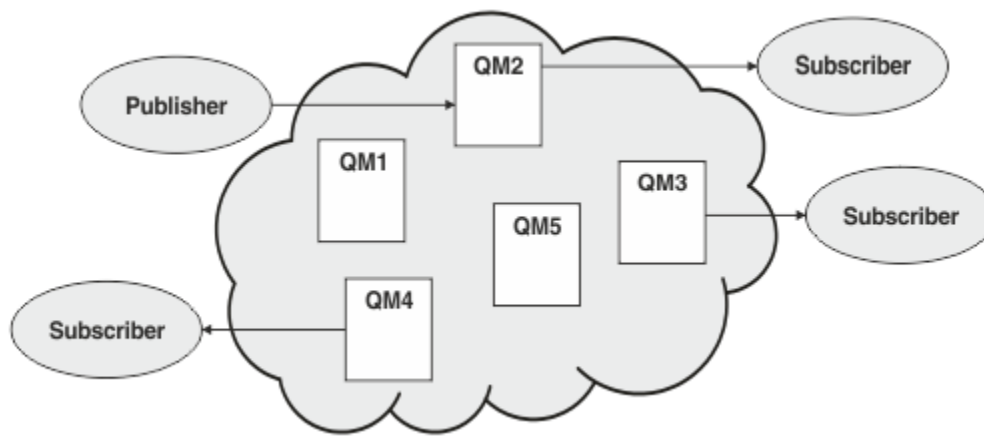


Figura 33. Cluster publicar/assinar

Há duas maneiras de configurar como as mensagens publicar/assinar são roteadas em um cluster:

- roteamento direto
- roteamento de host de tópico

Ao configurar um tópico em cluster roteado diretamente, as mensagens publicadas em um gerenciador de filas são enviadas diretamente desse gerenciador de filas para cada assinatura em qualquer outro gerenciador de filas no cluster. Isso pode fornecer o caminho mais direto para publicações, mas não resulta em todos os gerenciadores de filas em um cluster se tornando cientes de todos os outros gerenciadores de filas, cada um possivelmente tendo canais de cluster estabelecidos entre eles.

Ao usar o roteamento de host de tópico, as mensagens publicadas em um gerenciador de filas são enviadas de lá para um gerenciador de filas que hospeda uma definição do objeto do tópico administrado. O *gerenciador de filas do host de tópico* roteia a mensagem para cada assinatura em qualquer outro gerenciador de filas no cluster. Se os publicadores ou assinantes não forem localizados nos gerenciadores de filas de host de tópico, isso resulta em uma rota mais longa para publicações. No entanto, o benefício é que somente os gerenciadores de filas de host de tópico se tornam cientes de todos os outros gerenciadores de filas no cluster e, possivelmente, têm canais de cluster estabelecidos com eles.

Para obter mais informações, consulte [“Publicar/assinar clusters”](#) na página 95.

Visão Geral de hierarquias publicar/assinar

Uma hierarquia publicar/assinar é um conjunto de gerenciadores de filas conectados por canais em uma estrutura hierárquica. Cada gerenciador de filas identifica seu gerenciador de filas *pai*, conforme descrito em [Conectando um gerenciador de filas a uma hierarquia publicar/assinar](#).

Os publicadores e assinantes de um tópico podem se conectar a qualquer gerenciador de filas na hierarquia, e as mensagens fluem entre eles usando a conectividade do gerenciador de filas hierárquico.

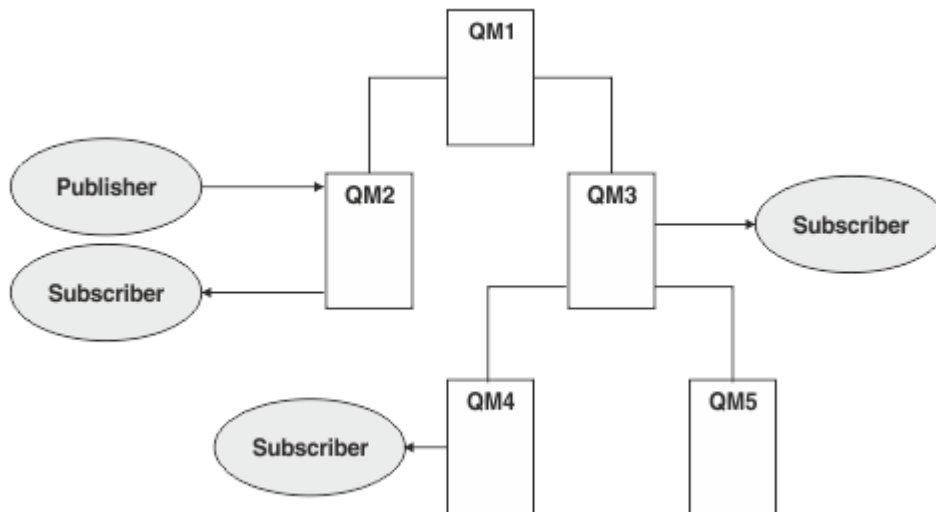


Figura 34. Hierarquia publicar/assinar

Na figura anterior, as publicações entregues aos assinantes em QM3 e QM4 foram roteadas de QM2 para QM1 e, em seguida, para QM3 e, por último, QM4.

As hierarquias fornecem controle direto sobre os relacionamentos entre cada gerenciador de filas na hierarquia. Isso permite um controle de baixa granularidade sobre o roteamento de mensagens de publicadores para assinantes e é útil principalmente ao rotear entre redes do gerenciador de filas conectividade restrita. É necessário considerar especialmente a disponibilidade e capacidade de cada gerenciador de filas por meio do qual uma mensagem é roteada em seu trajeto do publicador para assinantes.

Para obter mais informações, consulte [“Hierarquias de Publicação/Assinatura”](#) na página 97.

Distribuição de publicações entre gerenciadores de filas

Além das opções de roteamento, há duas abordagens para distribuir publicações em uma rede de gerenciadores de filas:

- Enviar publicações somente de um gerenciador de filas para os gerenciadores de filas que hospedam atualmente uma assinatura para essa publicação.
- Enviar cada publicação para todos os gerenciadores de filas e deixar que eles façam a correspondência dela com suas assinaturas.

A opção anterior resulta em mensagens sendo enviadas somente onde necessário, mas requer um nível de conhecimento de assinatura para ser compartilhado entre gerenciadores de filas. A última opção não requer o compartilhamento de conhecimento de assinatura, mas pode resultar no envio desnecessário de mensagens de publicação entre gerenciadores de filas.

Por padrão, o IBM MQ usa o método anterior, no qual as publicações são enviadas somente para gerenciadores de filas que possuam assinaturas. O conhecimento assinatura é propagado entre os gerenciadores de filas na forma de *assinaturas de proxy*. Isso depende da distribuição e do tempo de vida de assinaturas, e da frequência de publicações, com relação a qual é o mais eficiente para uso em uma topologia publicar/assinar distribuída. Consulte [Desempenho de assinatura em redes publicar/assinar](#).

Conceitos relacionados

[“Árvores de Tópicos”](#) na página 78

Cada tópico que você definir é um elemento ou nó na árvore de tópicos. A árvore de tópicos pode estar vazia para iniciar ou conter tópicos que foram definidos anteriormente usando comandos MQSC ou PCF. Você pode definir um novo tópico usando os comandos create topic ou especificando o tópico pela primeira vez em uma publicação ou assinatura.

Tarefas relacionadas

Projetando clusters publicar/assinar

Publicar/assinar clusters

Um cluster publicar/assinar é um cluster padrão de gerenciadores de filas interconectados, nos quais as publicações são automaticamente movidas de aplicativos de publicação para assinaturas que existem em qualquer um dos gerenciadores de filas no cluster. Há duas opções para rotear publicações em um cluster de publicação/assinatura: *roteamento direto* e *roteamento de host de tópico*. O roteamento escolhido depende do tamanho e de padrões de atividade esperados para seu cluster.

Um cluster que é usado para o sistema de mensagens de publicar/assinar não é diferente de um cluster do IBM MQ padrão. Como tal, os gerenciadores de filas no cluster de publicação/assinatura podem existir em computadores fisicamente separados e cada par de gerenciadores de filas é automaticamente conectado junto por canais de cluster quando necessário. Para obter informações adicionais, consulte [Clusters](#).

Para configurar um cluster padrão de gerenciadores de filas para sistemas de mensagens de publicação/assinatura, você define um ou mais objetos de tópico administrado em um gerenciador de filas no cluster. Para tornar o tópico um tópico de cluster, você configura a propriedade **CLUSTER** com o nome do cluster. Ao fazer isso, qualquer tópico usado por um publicador ou assinante nesse ponto ou abaixo na *árvore de tópicos* é compartilhado entre todos os gerenciadores de filas no cluster, e as mensagens publicadas em uma ramificação de cluster da *árvore de tópicos* são automaticamente roteadas para assinaturas em outros gerenciadores de filas no cluster.

Somente uma cópia de cada mensagem é enviada entre o gerenciador de filas do publicador e cada um dos outros gerenciadores de filas, independentemente do número de assinantes para a mensagem no gerenciador de filas de destino. Na chegada a um gerenciador de filas com uma ou mais assinaturas, a mensagem é duplicada em todas as assinaturas.

Qualquer gerenciador de filas que se une ao cluster automaticamente se torna ciente dos tópicos em cluster e publicadores e assinantes nesse gerenciador de filas participam automaticamente do cluster.

Atividades que são em cluster de publicação/assinatura também podem ocorrer em um cluster de publicação/assinatura, trabalhando com as sequências de tópicos que não estão em um objeto de tópico em cluster.

Há duas opções para rotear publicações em um cluster de publicação/assinatura: *roteamento direto* e *roteamento de host de tópico*. Para escolher o roteamento de mensagens para usar dentro do cluster, você configura a propriedade **CLROUTE** no objeto de tópico administrado para um dos valores a seguir:

- **DIRECT**
- **TOPICHOST**

Por padrão, o roteamento de tópico é **DIRECT**. Esta foi a única opção antes do IBM MQ 8.0. Ao configurar um tópico de cluster roteado diretamente em um gerenciador de filas, todos os gerenciadores de filas no cluster ficam cientes de todos os outros gerenciadores de filas no cluster. Ao executar operações de publicação e assinatura, cada gerenciador de filas pode se conectar diretamente a qualquer outro gerenciador de filas no cluster.

A partir de IBM MQ 8.0, é possível configurar o roteamento de tópico como **TOPICHOST**. Quando você usa o roteamento de host de tópico, todos os gerenciadores de filas no cluster ficam cientes dos gerenciadores de filas do cluster que hospedam a definição de tópico roteado (ou seja, os gerenciadores de filas nos quais você definiu o objeto do tópico). Ao executar operações de publicação e assinatura, os gerenciadores de filas no cluster se conectam apenas a estes gerenciadores de filas do host de tópico, e não diretamente uns aos outros. Os gerenciadores de filas do host de tópico são responsáveis pelas publicações de roteamento a partir de gerenciadores de filas em que as publicações são publicadas para gerenciadores de filas com assinaturas correspondentes.

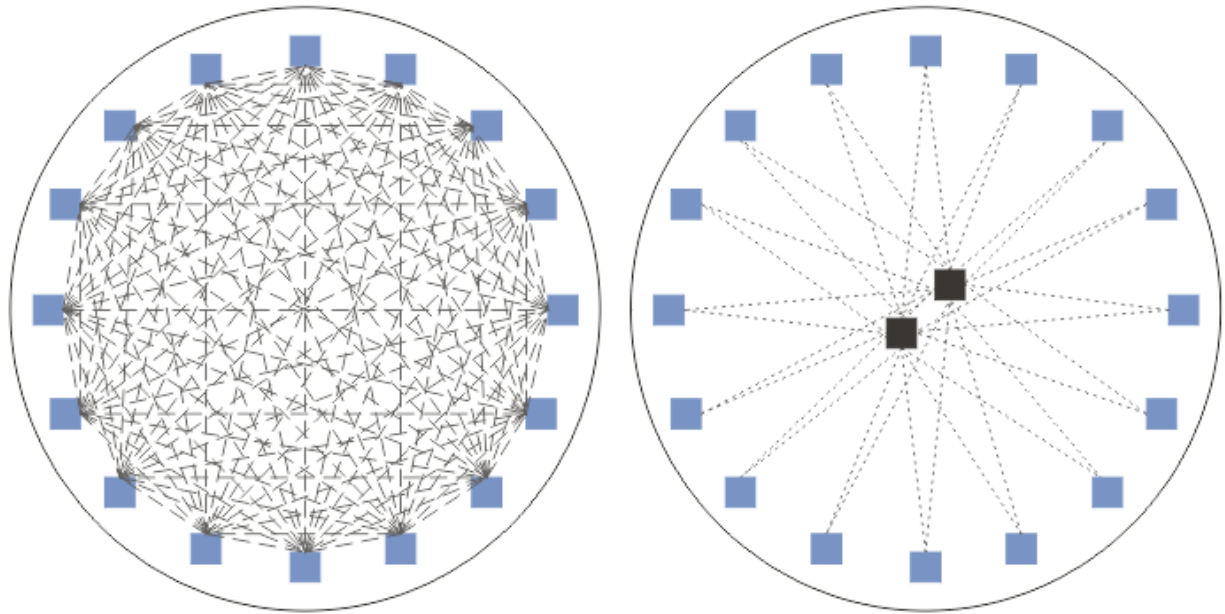


Figura 35. Roteamento Direto e Roteamento de Host de Tópico

Uma visão geral de roteamento direto

Quando um objeto do tópico administrado está configurado para roteamento direto, o objeto do tópico precisa somente ser definido em um dos gerenciadores de filas no cluster para que todos os gerenciadores de filas tomem conhecimento dele. A opção de gerenciador de filas no qual o tópico é definido não afeta o comportamento do sistema de mensagens publicar/assinar para o tópico.

Cada mensagem flui diretamente do gerenciador de filas do publicador para cada assinatura nos outros gerenciadores de filas no cluster, não passando por nenhum gerenciador de filas intermediário.

Por padrão, as mensagens são enviadas somente para outros gerenciadores de filas no cluster que hospedam uma ou mais assinaturas.

- Isso depende de cada gerenciador de filas informar diretamente todos os outros gerenciadores de filas no cluster de todos os tópicos que atualmente possuem uma ou mais assinaturas nele. Isso resulta em todos os gerenciadores de filas no cluster estando cientes de todos os tópicos que estão sendo assinados e de qualquer gerenciador de filas que hospeda uma assinatura estabelecendo um canal para cada um dos outros gerenciadores de filas. Isso independe de cada gerenciador de filas ter um publicador.
- O conhecimento de cada tópico individual assinado em todos os gerenciadores de filas pode ser removido, alterando para um modelo de envio de todas as publicações para todos os gerenciadores de filas no cluster, independentemente de terem assinaturas. Isso reduz o tráfego de conhecimento de assinatura, mas é provável que aumente o tráfego de publicações e o número de canais que cada gerenciador de filas estabelece. Consulte [Desempenho de assinatura em redes publicar/assinar](#).

Os fluxos de mensagens publicar/assinar usando tópicos em cluster roteados diretamente podem abranger diversos clusters publicar/assinar, incluindo um gerenciador de filas de cada cluster em uma hierarquia publicar/assinar. Consulte [Combinando os espaços de tópico de diversos clusters](#).

Para obter uma exploração mais detalhada de roteamento direto, consulte [Roteamento direto de clusters publicar/assinar](#).

Uma visão geral do roteamento de host de tópico

Quando um objeto de tópico administrado é configurado para roteamento de host de tópico, as publicações de um gerenciador de filas no cluster são roteadas por meio de um gerenciador de filas

no qual o objeto de tópico está configurado (um " host de tópico "), e de lá para os gerenciadores de filas nos quais as assinaturas existem.

- Isso depende de cada gerenciador de filas informar todos os hosts de tópico de cada tópico que atualmente possui uma ou mais assinaturas para ele. Qualquer gerenciador de filas que hospeda uma assinatura estabelece um canal para cada host de tópico para o tópico ao qual a assinatura está relacionada.
- Os gerenciadores de filas sem host tópico não ficaram cientes de outros gerenciadores de filas sem host de tópico no cluster para os propósitos de publicar/assinar, e os canais não são estabelecidos entre eles para esse propósito.
- Se o aplicativo de publicação estiver conectado a um gerenciador de filas que hospeda o tópico, as mensagens publicadas serão roteadas diretamente para os gerenciadores de filas nos quais as assinaturas correspondentes foram criadas, sem requerer um 'hop' adicional. De forma semelhante, se as assinaturas correspondentes forem criadas no único gerenciador de filas que hospeda o tópico, as mensagens publicadas nesse tópico serão roteadas diretamente para esse gerenciador de filas, sem requerer um hop adicional.
- As assinaturas no mesmo gerenciador de filas que o publicador são atendidas sem primeiro rotear publicações para os hosts do objeto do tópico.

Assim como para filas em cluster, diversos gerenciadores de filas podem configurar o mesmo objeto do tópico administrativo. Isso fornece maior disponibilidade de roteamento de mensagem e escala horizontal por meio de balanceamento de carga de trabalho. Para objetos do tópico roteados por host de tópico, quando diversos gerenciadores de filas configuram o mesmo tópico nomeado para a mesma ramificação da árvore de tópicos, cada host de tópico se torna ciente dos tópicos assinados por cada gerenciador de filas que hospeda uma assinatura.

- Quando uma mensagem é publicada, ela é enviada para um dos gerenciadores de filas de host de tópico para encaminhar para os gerenciadores de filas que hospedam a assinatura. A opção do gerenciador de filas de host de tópico segue as mesmas regras de balanceamento de carga de trabalho padrão que para filas ponto a ponto em cluster.
- Se um ou mais gerenciadores de filas de host de tópico não puderem ser contatados por um gerenciador de filas de publicação, as mensagens serão roteadas para os gerenciadores de filas de host de tópico restantes disponíveis.

Cada publicação para um tópico em uma ramificação roteada da árvore de tópicos é encaminhada para um dos hosts de tópico, mesmo que não haja nenhuma assinatura para esse tópico em nenhum lugar no cluster. Por padrão, as mensagens são enviadas daqui somente para outros gerenciadores de filas no cluster que hospedam uma ou mais assinaturas.

- Isso depende de cada gerenciador de filas de host de tópico estar sendo informado de todas as sequências de tópicos assinadas em cada gerenciador de filas no cluster.
- O conhecimento de cada tópico individual assinado pode ser removido, alterando para um modelo de envio de todas as publicações roteadas para um host de tópico para todos os gerenciadores de filas no cluster, independentemente de terem assinaturas. Isso reduz o tráfego de conhecimento de assinatura, mas é provável que aumente o tráfego de publicações e, possivelmente, o número de canais estabelecidos com cada gerenciador de filas de host de tópico. Consulte [Desempenho de assinatura em redes publicar/assinar](#).

Os fluxos de mensagens publicar/assinar usando tópicos em cluster roteados por host de tópico **não podem** abranger diversos clusters publicar/assinar usando uma hierarquia publicar/assinar.

Para obter uma exploração mais detalhada do roteamento de host de tópico, consulte [Roteamento de host de tópico em clusters publicar/assinar](#).

Hierarquias de Publicação/Assinatura

Você constrói uma hierarquia publicar/assinar vinculando os gerenciadores de filas usando canais, em seguida, definindo um relacionamento pai-filho entre pares de gerenciadores de filas. Uma mensagem flui de um publicador para as assinaturas por meio das relações diretas em uma hierarquia. Observe que isso pode significar diversos "hops" para chegar lá.

Somente uma cópia da mensagem é enviada entre qualquer par de gerenciadores de filas, independentemente do número de assinantes para a mensagem no gerenciador de filas de destino. Na chegada a um gerenciador de filas com uma ou mais assinaturas, a mensagem é duplicada em todas as assinaturas.

Por padrão, as mensagens são enviadas somente a outros gerenciadores de filas na hierarquia que estão na rota para uma assinatura em outro gerenciador de filas:

- Isso depende de cada gerenciador de filas que informa cada relação direta de todos os tópicos que atualmente possuem uma ou mais assinaturas, neste gerenciador de filas ou em uma de suas outras relações. Isso resulta em todos os gerenciadores de filas na hierarquia ficando cientes de todos os tópicos que estão sendo assinados.
- Esse comportamento pode ser mudado para sempre enviar publicações para todos os gerenciadores de filas na hierarquia, independentemente de quaisquer assinaturas existentes. Isso remove a necessidade de propagar as informações de assinatura na hierarquia, mas pode aumentar o tráfego de publicações.

Ao criar um cluster, é preciso ter cuidado para não criar um loop que faz com que as mensagens entrem em um ciclo permanente na rede. Nenhum loop desse tipo pode ser criado em uma hierarquia.

Cada gerenciador de filas deve ter um nome do gerenciador de filas exclusivo.

Os fluxos de mensagens publicar/assinar podem abranger diversos clusters publicar/assinar. Para isso, inclua um gerenciador de filas de cada cluster em uma hierarquia publicar/assinar.

Para obter uma exploração mais detalhada, consulte [Roteamento em hierarquias publicar/assinar](#).

Assinaturas de proxy em uma rede publicar/assinar

Uma assinatura de proxy é uma assinatura feita por um gerenciador de filas para tópicos publicados em outro gerenciador de filas. Uma assinatura de proxy flui entre gerenciadores de filas para cada sequência de tópico individual que é assinada por uma assinatura. Você não cria assinaturas de proxy explicitamente, o gerenciador de filas faz isso em seu nome.

É possível conectar gerenciadores de filas em um cluster publicar/assinar ou em uma hierarquia publicar/assinar. Fluxo de assinaturas de proxy entre os gerenciadores de filas conectados. As assinaturas de proxy fazem publicações em um tópico criadas por um publicador conectado a um gerenciador de filas serem recebidas por assinantes desse tópico conectados a outros gerenciadores de filas. Consulte o [“Redes publicar/assinar distribuídas” na página 91](#).

Em topologias publicar/assinar com muitas milhares de assinaturas para sequências de tópicos individuais ou onde a existência dessas assinaturas pode ser rapidamente alterada, deve ser considerada a sobrecarga da propagação de assinaturas de proxy. Além da agregação automática descrita no restante deste tópico, é possível fazer mudanças manuais na configuração, que restringem ainda mais o fluxo de assinaturas de proxy e de publicações entre gerenciadores de filas conectados, e reduzem a latência de espera para que uma assinatura de proxy seja propagada para todos os gerenciadores de filas conectados. Consulte [Desempenho de assinatura em redes publicar/assinar](#).

As assinaturas de proxy não contêm seletores usados por assinaturas locais e as sequências de tópicos de assinatura que contêm curingas podem ser simplificadas. Isso pode resultar em publicações correspondendo a assinaturas de proxy, em que as assinaturas reais não correspondem, resultando em um fluxo de publicações adicional entre gerenciadores de filas. O gerenciador de filas que hospeda as assinaturas filtra essas discrepâncias para que publicações adicionais não sejam retornadas às assinaturas.

Agregação de Assinatura de Proxy

As assinaturas de proxy são agregadas usando um sistema de eliminação duplicado. Para uma sequência de tópico resolvida específica, uma assinatura de proxy é enviada na primeira assinatura local ou assinatura de proxy recebida. Assinaturas subsequentes para a mesma sequência de tópicos faz uso desta assinatura de proxy existente.

A assinatura de proxy é cancelada após a última assinatura local ou após a assinatura de proxy recebida ser cancelada.

Agregação da Publicação

Quando houver mais de uma assinatura para a mesma sequência de tópicos em um gerenciador de filas, apenas uma única cópia de cada publicação correspondente a sequência de tópicos é enviada a partir de outros gerenciadores de filas na topologia de publicação/assinatura. Na chegada da mensagem, o gerenciador de filas local fornece uma cópia da mensagem para cada assinatura correspondente.

É possível para mais de uma assinatura de proxy para corresponder à sequência de tópicos de uma única publicação quando as assinaturas de proxy contiverem curingas. Se uma mensagem for publicada em um gerenciador de filas que corresponde a duas ou mais assinaturas de proxy criadas por um único gerenciador de filas conectado, apenas uma cópia da publicação será encaminhada para o gerenciador de filas remoto para satisfazer as assinaturas de proxy múltiplas.

Conceitos relacionados

[Detecção de loop em uma rede de publicação/assinatura distribuída](#)

Curingas em Assinaturas de Proxy

As assinaturas podem usar curingas nas sequências de tópicos para correspondências com sequências de tópicos em publicações.

Há dois esquemas curinga que uma assinatura pode usar: *baseado em tópico* e *baseado em caracteres*. Consulte o [“Esquemas Curinga” na página 72](#).

No IBM MQ, todas as assinaturas de proxy para assinaturas curinga são convertidas para usar curingas baseados em tópico. Se um curinga baseado em caracteres for localizado, ele será substituído por um caractere #, de volta para a / mais próxima. Por exemplo, /aaa/bbb/c*d é convertido em /aaa/bbb/#. A conversão resulta em gerenciadores de filas remotos enviando um pouco mais de publicações a que foram explicitamente assinadas. As publicações adicionais são filtradas pelo gerenciador de filas local, quando ele entrega as publicações para seus assinantes locais.

Controlando o Uso de Curinga com a Propriedade WILDCARD

Use a propriedade MQSC **Topic WILDCARD** ou a propriedade PCF **Topic WildcardOperation** equivalente para controlar a entrega de publicações para aplicativos assinantes que usam nomes da sequência de tópicos curinga. A propriedade WILDCARD pode ter um dos dois valores possíveis:

CURINGA

O comportamento de assinaturas curingas com relação a este tópico.

PASSTHRU

As assinaturas feitas em um tópico curinga menos específico do que a sequência de tópicos neste objeto do tópico recebem as publicações feitas neste tópico e para sequências de tópicos mais específicas do que este tópico.

BLOCK

As assinaturas feitas em um tópico curinga menos específico do que a sequência de tópicos neste objeto do tópico não recebem as publicações feitas neste tópico ou nas sequências de tópicos mais específicas do que este tópico.

O valor deste atributo é usado quando as assinaturas são definidas. Se você alterar este atributo, o conjunto de tópicos coberto pelas assinaturas existentes não será afetado pela modificação. Este cenário também se aplica se a topologia for alterada quando os objetos do tópico forem criados ou excluídos; o conjunto de tópicos que corresponde às assinaturas criadas seguindo a modificação do atributo WILDCARD é criado usando a topologia modificada. Se você desejar forçar o conjunto de tópicos correspondente para ser reavaliado para assinaturas existentes, deve reiniciar o gerenciador de filas.

No exemplo, [“Exemplo: Criar um Cluster de Publicação/Assinatura Sport” na página 85](#), é possível seguir as etapas para criar a estrutura da árvore de tópicos mostrada em [Figura 23 na página 82](#).

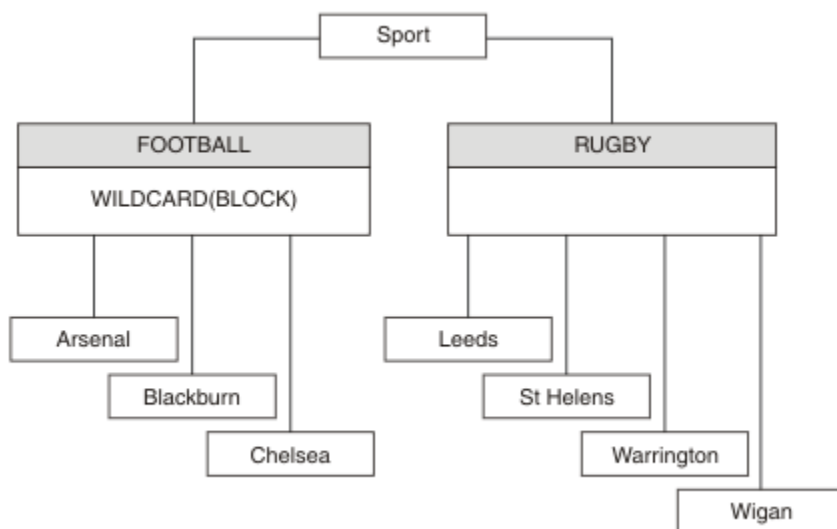


Figura 36. Uma árvore de tópicos que usa a propriedade WILDCARD, BLOCK

Um assinante que usa a sequência de tópicos curinga # recebe todas as publicações para o tópico Sport e subárvore Sport/Rugby. O assinante não recebe nenhuma publicação para a subárvore Sport/Football, porque o valor da propriedade WILDCARD do tópico Sport/Football é BLOCK.

PASSTHRU é a configuração padrão. É possível configurar o valor da propriedade PASSTHRU de WILDCARD para os nós na árvore Sport. Se os nós não tiverem o valor da propriedade BLOCK do WILDCARD, configurar PASSTHRU não vai alterar o comportamento observado pelos assinantes para os nós na árvore Sports.

No exemplo, crie assinaturas para ver como a configuração curinga afeta as publicações que são entregues; consulte Figura 27 na página 87. Execute o comando de publicação no Figura 30 na página 88 para criar algumas publicações.

```
pub QMA
```

Figura 37. Publicar para QMA

Os resultados são mostrados em Tabela 3 na página 83. Observe como a configuração do valor da propriedade BLOCK de WILDCARD, impede que as assinaturas com curingas recebam as publicações para os tópicos no escopo do curinga.

Tabela 6. Publicações Recebidas em QMA			
Assinatura	Cadeia do tópico	Publicações Recebidas	Notas
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Todas as publicações para a subárvore Football bloqueadas por WILDCARD(BLOCK) em Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD(BLOCK) em Sports/Football impede as assinaturas curingas em Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	WILDCARD padrão em Sports/Rugby não impede a assinatura curinga em Leeds.

Nota:

Imagine que uma assinatura tenha um curinga que corresponda a um objeto do tópico com o valor da propriedade BLOCK de WILDCARD. Se a assinatura também tiver uma sequência de tópicos à direita do curinga correspondente, a assinatura nunca receberá uma publicação. O conjunto de publicações que não estão bloqueadas são publicações para tópicos que são pais do curinga bloqueado. As publicações para os tópicos que são filhos do tópico com o valor da propriedade BLOCK são bloqueadas pelo curinga. Portanto, as sequências de tópicos de assinatura que incluem um tópico à direita do curinga nunca recebem qualquer publicação a ser correspondida.

Configurar o valor da propriedade WILDCARD para BLOCK não significa que você não possa se inscrever usando uma sequência de tópicos que inclua curingas. Essa assinatura é normal. A assinatura tem um tópico explícito que corresponde ao tópico com um objeto do tópico que possui um valor da propriedade BLOCK de WILDCARD. Ela usa os curingas para os tópicos que são pais ou filhos do tópico com o valor da propriedade BLOCK de WILDCARD. No exemplo em [Figura 23 na página 82](#), uma assinatura, como `Sports/Football/#`, pode receber as publicações.

Curingas e Tópicos do Cluster

As definições do tópico do cluster são propagadas para cada gerenciador de filas em um cluster. Uma assinatura para um tópico de cluster em um gerenciador de filas em um cluster resulta no gerenciador de filas criando assinaturas de proxy. Uma assinatura de proxy é criada em cada outro gerenciador de filas no cluster. As assinaturas que usam sequências de tópicos que contêm curingas, combinados com tópicos de cluster, podem fornecer um comportamento difícil de ser previsto. O comportamento é explicado no seguinte exemplo.

Na configuração do cluster para o exemplo, [“Exemplo: Criar um Cluster de Publicação/Assinatura Sport” na página 85](#), QMB tem o mesmo conjunto de assinaturas que QMA, embora QMB não tenha recebido nenhuma publicação após o publicador publicado para QMA, consulte [Figura 24 na página 83](#). Embora os tópicos `Sports/Football` e `Sports/Rugby` sejam tópicos de cluster, as assinaturas definidas em `fullsubs.tst` não fazem referência a um tópico de cluster. Nenhuma assinatura de proxy é propagada de QMB para QMA. Sem as assinaturas de proxy, nenhuma publicação para QMA é encaminhada para QMB.

Algumas das assinaturas, como `Sports/#/Leeds`, podem parecer fazer referência a um tópico do cluster, neste caso, `Sports/Rugby`. A assinatura `Sports/#/Leeds` resolve, na realidade, para o objeto de tópico `SYSTEM.BASE.TOPIC`.

A regra para resolver o objeto do tópico referenciado por uma assinatura, como `Sports/#/Leeds`, é a seguinte. Truncar a sequência de tópicos para o primeiro curinga. Varra para a esquerda da sequência de tópicos procurando pelo primeiro tópico que tem um objeto de tópico administrativo associado. O objeto do tópico pode especificar um nome de cluster ou definir um objeto de tópico local. No exemplo, `Sports/#/Leeds`, a sequência de tópicos após o truncamento é `Sports`, que não possui nenhum objeto do tópico e, portanto, `Sports/#/Leeds` herda de `SYSTEM.BASE.TOPIC`, que é um objeto do tópico local.

Para ver como a assinatura nos tópicos em cluster pode alterar a maneira como a propagação de curinga funciona, execute o script em lote, `upsubs.bat`. O script limpa as filas de assinatura e inclui as assinaturas de tópico de cluster em `fullsubs.tst`. Execute `puba.bat` novamente para criar um lote de publicações; consulte [Figura 24 na página 83](#).

[Tabela 4 na página 84](#) mostra o resultado de incluir duas novas assinaturas no mesmo gerenciador de filas no qual as publicações foram publicadas. O resultado é o esperado, as novas assinaturas recebem uma publicação cada e os números de publicações recebidas pelas outras assinaturas ficam inalterados. Ocorrem os resultados inesperados no outro gerenciador de filas do cluster; consulte [Tabela 5 na página 85](#).

Assinatura	Cadeia do tópico	Publicações Recebidas	Notas
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Todas as publicações para a subárvore Football bloqueadas por WILDCARD (BLOCK) em Sports/ Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) em Sports/ Football impede as assinaturas curingas em Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	WILDCARD padrão em Sports/Rugby não impede a assinatura curinga em Leeds.
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal recebe uma publicação porque a assinatura não tem um curinga.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds receberia uma publicação em qualquer evento.

Tabela 5 na página 85 mostra os resultados de incluir as duas novas assinaturas no QMB e publicar em QMA. A rechamada desse QMB não recebeu nenhuma publicação sem essas duas novas assinaturas. Conforme esperado, as duas novas assinaturas recebem as publicações, porque Sports/FootBall e Sports/Rugby são tópicos de cluster. O QMB encaminhou assinaturas de proxy para Sports/ Football/Arsenal e Sports/Rugby/Leeds para QMA, que então enviou as publicações para QMB.

O resultado inesperado é que duas assinaturas Sports/# e Sports/#/Leeds que não receberam anteriormente nenhuma publicação, agora recebem as publicações. A razão é que as publicações Sports/Football/Arsenal e Sports/Rugby/Leeds encaminhadas para QMB para as outras assinaturas agora estão disponíveis para qualquer assinante conectado ao QMB. Consequentemente as assinaturas para os tópicos locais Sports/# E Sports/#/Leeds recebem a publicação Sports/ Rugby/Leeds. Sports/#/Arsenal continua não recebendo uma publicação, porque Sports/Football tem seu valor de propriedade WILDCARD configurado como BLOCK.

Assinatura	Cadeia do tópico	Publicações Recebidas	Notas
SPORTS	Sports/#	Sports/Rugby/ Leeds	Todas as publicações para a subárvore Football bloqueadas por WILDCARD (BLOCK) em Sports/ Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) em Sports/ Football impede as assinaturas curingas em Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/ Leeds	WILDCARD padrão no Sports/Rugby não evita a assinatura curinga em Leeds.
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal recebe uma publicação porque a assinatura não tem um curinga.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds receberia uma publicação em qualquer evento.

Na maioria dos aplicativos, é desejável para uma assinatura influenciar o comportamento de outra assinatura. Um uso importante da propriedade WILDCARD com valor BLOCK é fazer com que as assinaturas para a mesma sequência de tópicos que contém curingas se comportem uniformemente. Indica se a assinatura está no mesmo gerenciador de filas que o publicador ou um diferente, os resultados da assinatura são os mesmos.

Curingas e Fluxos

Para um novo aplicativo gravado para a API de publicação/assinatura, o efeito é que uma assinatura para * não recebe nenhuma publicação. Para receber todas as publicações de Sports você deve se inscrever a Sports/* ou Sports/#, de maneira semelhante para publicações Business.

O comportamento de um aplicativo de publicação / assinatura enfileirado existente não muda quando o broker de publicação / assinatura é migrado para uma versão posterior do IBM MQ. A propriedade **StreamName** nos comandos **Publish**, **Register Publisher** ou **Subscriber** é mapeada para o nome do tópico para o qual o fluxo foi migrado..

Curingas e Pontos de Assinatura

Para um novo aplicativo gravado para a API de publicação/assinatura, o efeito da migração é que uma assinatura para * não receba nenhuma publicação. Para receber todas as publicações de Sports você deve se inscrever a Sports/* ou Sports/#, de maneira semelhante para publicações Business.

O comportamento de um aplicativo de publicação / assinatura enfileirado existente não muda quando o broker de publicação / assinatura é migrado para uma versão posterior do IBM MQ. A propriedade **SubPoint** nos comandos **Publish**, **Register Publisher** ou **Subscriber** é mapeada para o nome do tópico para o qual a assinatura foi migrada..

Exemplo: Criar um Cluster de Publicação/Assinatura Sport

As etapas que seguem a criação de um cluster, CL1, com quatro gerenciadores de filas: dois repositórios completos, CL1A e CL1B, e dois repositórios parciais, QMA e QMB. Os repositórios completos são usados para reter apenas as definições de cluster. QMA designado ao host de tópico do cluster. As assinaturas duráveis são definidas em QMA e QMB.

Nota: O exemplo é codificado para Windows. Você deve recodificar [Create qmgrs.bat](#) e [create pub.bat](#) para configurar e testar o exemplo em outras plataformas.

1. Crie os arquivos de script.
 - a. [Create topics.tst](#)
 - b. [Create wildsubs.tst](#)
 - c. [Create fullsubs.tst](#)
 - d. [Create qmgrs.bat](#)
 - e. [create pub.bat](#)
2. Execute [Crie qmgrs.bat](#) para criar a configuração.

```
qmgrs
```

Crie os tópicos em [Figura 23 na página 82](#). O script na figura 5 cria os tópicos de cluster Sports/Football e Sports/Rugby

Nota: A opção REPLACE não substitui as propriedades TOPICSTR de um tópico. TOPICSTR é uma propriedade que é proveitosamente variada no exemplo para testar as diferentes árvores de tópicos. Para alterar os tópicos, exclua primeiro o tópico.


```

DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')

```

Figura 38. Excluir e Criar os Tópicos: topics.tst

Nota: Exclua os tópicos, porque o REPLACE não substitui as sequências de tópicos.

Crie assinaturas com curingas. Os curingas correspondentes dos tópicos com objetos do tópico no [Figura 23 na página 82](#). Crie uma fila para cada assinatura. As filas são limpas e as assinaturas excluídas quando o script não é executado ou reexecutado.

Nota: A opção REPLACE não substitui as propriedades TOPICOBJ ou TOPICSTR de uma assinatura. TOPICOBJ ou TOPICSTR são propriedades proveitosamente variadas no exemplo para testar diferentes assinaturas. Para alterá-las, exclua primeiro a assinatura.

```

DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)

```

Figura 39. Criar Assinaturas Curinga: wildsubs.tst

Crie as assinaturas que fazem referência aos objetos do tópico de cluster.

Nota:

O delimitador, /, é inserido automaticamente entre a sequência de tópicos referenciada por TOPICOBJ e a sequência de tópicos definida por TOPICSTR.

A definição, DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) cria a mesma assinatura. TOPICOBJ é usado como uma maneira rápida de fazer referência à sequência de tópicos que você já definiu. A assinatura, quando criada, não se refere mais ao objeto do tópico.


```

DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)

```

Figura 40. Excluir e Criar Assinaturas: fullsubs.tst

Crie um cluster com dois repositórios. Crie dois repositórios parciais para publicar e subscrever. Execute novamente o script para excluir tudo e inicie novamente. O script também cria a hierarquia de tópico e as assinaturas curingas iniciais.

Nota:

Em outras plataformas, grave um script semelhante ou digite todos os comandos. Usar um script facilita a exclusão de tudo e inicia novamente com uma configuração idêntica.

```

@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof

```

Figura 41. Criar Gerenciadores de Filas: qmgrs.bat

Atualize a configuração incluindo as assinaturas nos tópicos do cluster.

```

@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst

```

Figura 42. Atualizar Assinaturas: upsubs.bat

Execute pub.bat, com um gerenciador de filas como um parâmetro, para publicar as mensagens que contêm a sequência de tópicos de publicação. Pub.bat usa o programa de amostra **amqspub**.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Figura 43. Publicar: pub.bat

Conceitos relacionados

[Assinaturas curinga e publicações retidas](#)

Escopo da publicação

Ao configurar um cluster ou hierarquia publicar/assinar, o escopo de uma publicação controla ainda mais se os gerenciadores de filas encaminharão uma publicação para gerenciadores de filas remotas. Use o atributo de tópico **PUBSCOPE** para administrar o escopo de publicações.

Se uma publicação não for encaminhada para gerenciadores de filas remotos, apenas assinantes locais recebem a publicação.

Ao usar um cluster publicar/assinar, o escopo de publicações é controlado principalmente pela definição de objetos do tópico em cluster em alguns pontos na árvore de tópicos. O escopo de publicação deve ser configurado para permitir o fluxo de publicações para outros gerenciadores de filas no cluster. É necessário restringir o escopo de publicação para um tópico em cluster somente quando for necessário o controle de baixa granularidade de tópicos específicos em alguns gerenciadores de filas.

Ao usar uma hierarquia publicar/assinar, o escopo de publicações é controlado principalmente por este atributo, em conjunto com o atributo de [escopo de assinatura](#).

O atributo **PUBSCOPE** é usado para determinar o escopo de publicações feitas para um tópico específico. É possível configurar o atributo para um dos seguintes valores:

QMGR

A publicação é entregue apenas para assinantes locais. Estas publicações são chamadas de *publicações locais*. As publicações locais não são encaminhadas para gerenciadores de filas remotos e, portanto, não são recebidas pelos assinantes conectados aos gerenciadores de filas remotas.

ALL

A publicação é entregue para assinantes locais e assinantes conectados aos gerenciadores de filas remotas em um cluster ou hierarquia publicar/assinar. Estas publicações são chamadas de *publicações globais*.

ASPARENT

Use a configuração **PUBSCOPE** do tópico pai na árvore de tópicos.

Os publicadores também podem especificar se uma publicação é local ou global usando a opção de colocar a mensagem MQPMO_SCOPE_QMGR. Se essa opção for usada, ela substitui qualquer comportamento que foi configurado usando o atributo do tópico **PUBSCOPE**.

Conceitos relacionados

[“Objetos de Tópico Administrativo” na página 79](#)

Usando um objeto de tópico administrativo, é possível designar atributos específicos não padrão para os tópicos.

Tarefas relacionadas

[Configurando redes publicar/assinar distribuídas](#)

Escopo da assinatura

O escopo de uma assinatura controla se uma assinatura em um gerenciador de filas recebe publicações que são publicadas em outro gerenciador de filas em um cluster ou uma hierarquia de publicação/assinatura ou apenas as publicações de publicadores locais.

Limitando o escopo de assinatura a um gerenciador de filas para as assinaturas de proxy de serem encaminhadas para outros gerenciadores de filas na topologia de publicação/assinatura. Isso reduz entre o tráfego de mensagens de publicação/assinatura do gerenciador de filas.

Ao usar um cluster publicar/assinar, o escopo de assinaturas é controlado principalmente pela definição de objetos do tópico em cluster em alguns pontos na árvore de tópicos. O escopo de assinatura deve ser configurado para permitir o fluxo de assinaturas de proxy para outros gerenciadores de filas no cluster. É necessário restringir o escopo de assinatura para um tópico em cluster somente quando for necessário o controle de baixa granularidade de tópicos específicos em alguns gerenciadores de filas.

Ao usar uma hierarquia publicar/assinar, o escopo de assinaturas é controlado principalmente por este atributo, em conjunto com o atributo de escopo de publicação.

O atributo de tópico **SUBSCOPE** é usado para determinar o escopo de assinaturas feitas em um tópico específico. É possível configurar o atributo para um dos seguintes valores:

QMGR

Uma assinatura receberá apenas publicações locais e assinaturas de proxy não são propagadas para os gerenciadores de filas remotas.

ALL

Uma assinatura de proxy é propagada para gerenciadores de filas remotas em um cluster ou hierarquia publicar/assinar e o assinante recebe publicações locais e remotas.

ASPARENT

Use a configuração **SUBSCOPE** do tópico pai na árvore de tópicos.

Quando o escopo de assinatura de um tópico é configurado como ALL, diretamente ou resolvido por meio de ASPARENT, as assinaturas individuais para esse tópico podem restringir seu escopo para QMGR especificando MQSO_SCOPE_QMGR ao criar a assinatura. Uma assinatura para um tópico que tem um escopo de QMGR não pode ampliar o escopo para ALL.

Conceitos relacionados

[“Objetos de Tópico Administrativo” na página 79](#)

Usando um objeto de tópico administrativo, é possível designar atributos específicos não padrão para os tópicos.

Tarefas relacionadas

[Configurando redes publicar/assinar distribuídas](#)

Espaços de Tópico

Um espaço de tópico é o conjunto de tópicos no qual é possível assinar e publicar. Um gerenciador de filas em uma topologia publicar/assinar distribuída possui um espaço de tópico que possivelmente inclui tópicos que foram assinados e publicados em gerenciadores de filas conectados nessa topologia.

Nota: Para obter uma visão geral de tópicos em um gerenciador de filas, como objetos do tópico administrativos, sequências de tópicos e árvores de tópicos, consulte [“tópicos” na página 70](#). Referências adicionais a *tópicos* no artigo atual referem-se a *sequências de tópicos*, a menos que seja especificado de outra forma.

Os tópicos são inicialmente criados de uma das seguintes maneiras:

- de forma administrativa, ao definir um objeto do tópico ou assinatura durável.
- dinamicamente, quando um aplicativo cria uma publicação ou assinatura dinamicamente para um novo tópico.

Os tópicos são propagados para outros gerenciadores de filas tanto através de assinaturas de proxy e com a criação de objetos do tópico administrativo. As assinaturas de proxy resultam em publicações que estão sendo encaminhadas do gerenciador de filas para o qual um publicador está conectado, para os gerenciadores de filas de assinantes.

As assinaturas de proxy são propagadas entre todos os gerenciadores de filas que estão conectados juntos por relacionamentos pai-filho em uma hierarquia do gerenciador de filas. O resultado é, você pode assinar em um gerenciador de filas para um tópico definido em qualquer outro gerenciador de filas na

hierarquia. Contanto que haja um caminho conectado entre os gerenciadores de filas, não importa como os gerenciadores de filas estão conectados.

As assinaturas de proxy também são propagadas para assinaturas em tópicos de cluster em um cluster publicar/assinar. Um tópico de cluster é um tópico que está conectado a um objeto do tópico que tenha o atributo **CLUSTER** ou herda o atributo de seu pai. Os tópicos que não sejam tópicos de cluster são conhecidos como tópicos locais e não são replicados para o cluster. Nenhuma assinatura de proxy é propagada para o cluster de assinaturas para tópicos locais.

Para resumir, as assinaturas de proxy são criadas para os assinantes em duas circunstâncias.

1. Um gerenciador de filas é um membro de uma hierarquia e uma assinatura de proxy é redirecionada para o pai e filho do gerenciador de filas.
2. Um gerenciador de filas é um membro de um cluster e a sequência de tópicos de assinatura é resolvida para um tópico que está associado a um objeto do tópico de cluster. Quando o tópico é um tópico de cluster *roteado diretamente*, as assinaturas de proxy são encaminhadas para todos os membros do cluster. Quando o tópico é um tópico de cluster *roteado por host de tópico*, as assinaturas de proxy são encaminhadas somente para os gerenciadores de filas no cluster que definiram o objeto de tópico em cluster. Para obter mais informações, consulte [“Publicar/assinar clusters”](#) na página 95.

Se um gerenciador de filas for um membro de um cluster e uma hierarquia, as assinaturas de proxy são propagadas por ambos os mecanismos sem entregar publicações duplicadas para o assinante.

Os espaços de tópicos de três topologias de publicação/assinatura são descritos na lista a seguir:

- [“Caso 1. Publicar/assinar clusters”](#) na página 108.
- [“Caso 2. Hierarquias de Publicação/Assinatura”](#) na página 109.

Em tópicos separados, as seguintes tarefas de configuração descrevem como combinar espaços de tópico.

- [Criando um único espaço de tópico em um cluster publicar/assinar.](#)
- [Combinando os espaços de tópico de diversos clusters.](#)
- [Combinando e isolando espaços de tópico em diversos clusters.](#)
- [Publicando e assinando espaços de tópico em diversos clusters.](#)

Caso 1. Publicar/assinar clusters

No exemplo, suponha que o gerenciador de filas não está conectado a uma hierarquia de publicação/assinatura.

Se um gerenciador de filas for um membro de um cluster de publicação/assinatura, seu espaço de tópico será composto de tópicos locais e tópicos de cluster. Os tópicos locais são associados aos objetos do tópico sem o atributo **CLUSTER**. Se um gerenciador de filas tiver definições de objeto do tópico local, seu espaço de tópico será diferente de outro gerenciador de filas no cluster que também possui seus próprios objetos de tópico definidos localmente.

Em um cluster de publicação/assinatura, você não pode assinar um tópico definido em outro gerenciador de filas, a menos que o tópico que você assinar seja resolvido para um objeto de tópico de cluster.

Quando as definições com o mesmo nome de um objeto do tópico de cluster forem necessárias em diversos gerenciadores de filas, por exemplo, ao usar *roteamento de host de tópico*, é importante que todas as definições correspondam, onde necessário. Para obter informações adicionais, consulte [Criando um único espaço de tópico em um cluster publicar/assinar](#).

Uma definição local de um objeto do tópico, se a definição for para um tópico de cluster ou um tópico local, tem precedência sobre o mesmo objeto do tópico definido em outro lugar no cluster. O tópico localmente definido é usado, mesmo se o objeto definido em outro lugar for mais recente.

É importante que um objeto do tópico de cluster esteja associado com a mesma sequência de tópicos em todo o cluster. Você não pode modificar a sequência de tópicos com o qual um objeto do tópico está associado. Para associar o mesmo objeto do tópico com uma sequência de tópicos diferente, você deve excluir o objeto do tópico e recriá-lo com a sequência de tópico nova. Se o tópico for armazenado em

cluster, o efeito será excluir as cópias do objeto do tópico armazenadas em outros membros do cluster e, em seguida, para criar cópias do novo objeto do tópico em todos os lugares no cluster. As cópias do objeto do tópico se referem à mesma sequência de tópico.

É possível criar acidentalmente duas definições do mesmo objeto do tópico nomeado em gerenciadores de filas diferentes no cluster, com sequências de tópicos diferentes. Isto pode resultar em comportamento confuso, porque diversas definições do mesmo objeto do tópico com sequências de tópicos diferentes podem produzir diferentes resultados, dependendo de como e onde o tópico é referenciado. Consulte [Diversas definições de tópico de cluster com o mesmo nome](#) para obter informações adicionais sobre este ponto importante.

Caso 2. Hierarquias de Publicação/Assinatura

No exemplo, suponha que o gerenciador de filas não é membro de um cluster de publicação/assinatura.

No IBM MQ, se um gerenciador de filas for um membro de uma hierarquia de publicação / assinatura, seu espaço de tópico consistirá em todos os tópicos definidos localmente e em gerenciadores de filas conectados. O espaço de tópico de todos os gerenciadores de filas em uma hierarquia é o mesmo. Não há divisão de tópicos em tópicos locais e tópicos globais.

Configure uma das opções **PUBSCOPE** e **SUBSCOPE** para QMGR, para evitar que uma publicação em um tópico seja transmitida de um publicador para um assinante conectado aos gerenciadores de filas diferentes na hierarquia.

Suponha que você defina um objeto do tópico Alabama com a sequência de tópicos USA/Alabama no gerenciador de filas QMA. O resultado é o seguinte:

1. O espaço de tópico no QMA agora inclui o objeto do tópico Alabama e a sequência de tópicos USA/Alabama.
2. Um aplicativo ou administrador pode criar uma assinatura no QMA usando o nome do objeto do tópico Alabama.
3. Um aplicativo pode criar uma assinatura para qualquer tópico, incluindo USA/Alabama, em qualquer gerenciador de filas na hierarquia. Se o QMA não tiver sido definido localmente, o tópico USA/Alabama será resolvido para o objeto do tópico SYSTEM.BASE.TOPIC.

Conceitos relacionados

[“Escopo da publicação” na página 106](#)

Ao configurar um cluster ou hierarquia publicar/assinar, o escopo de uma publicação controla ainda mais se os gerenciadores de filas encaminharão uma publicação para gerenciadores de filas remotas. Use o atributo de tópico **PUBSCOPE** para administrar o escopo de publicações.

[“Escopo da assinatura” na página 106](#)

O escopo de uma assinatura controla se uma assinatura em um gerenciador de filas recebe publicações que são publicadas em outro gerenciador de filas em um cluster ou uma hierarquia de publicação/assinatura ou apenas as publicações de publicadores locais.

Tarefas relacionadas

[Configurando redes publicar/assinar distribuídas](#)

IBM MQ Multicast

O IBM MQ Multicast oferece baixa latência, alto fan-out, sistema de mensagens multicast confiável.

O multicast é uma forma eficiente de sistema de mensagens de publicação/assinatura porque pode ser escalada a um alto número de assinantes sem efeitos prejudiciais no desempenho. O IBM MQ ativa um sistema de mensagens Multicast confiável usando confirmações, confirmações negativas e números de sequência para obter um sistema de mensagens de baixa latência com alto fan-out.

A entrega justa do IBM MQ Multicast permite entrega quase simultânea, assegurando que nenhum destinatário tenha uma vantagem. Como o IBM MQ Multicast usa a rede para entregar as mensagens, um mecanismo de publicação/assinatura não é necessário para fan-out de dados. Depois que um tópico é mapeado para um endereço de grupo, não há necessidade para o gerenciador de filas porque

os publicadores e os assinantes podem operar em um modo ponto a ponto. Isso permite que o carregamento seja reduzido nos servidores do gerenciador de filas e o servidor do gerenciador de filas não é mais um ponto de falha em potencial.

Conceitos Iniciais e Multicast

IBM MQ Multicast pode ser facilmente integrado a sistemas e aplicativos existentes usando o objeto Informações de comunicação (COMMINFO). Dois campos do objeto TOPIC permitem a configuração rápida de objetos TOPIC existentes para suportar ou ignorar tráfego multicast.

Objetos Necessários para Multicast

As informações a seguir são uma visão geral em síntese dos dois objetos necessários para o IBM MQ Multicast:

Objeto COMMINFO

O objeto COMMINFO contém os atributos associados à transmissão multicast. Para obter mais informações sobre os parâmetros de objeto COMMINFO, consulte [DEFINE COMMINFO](#).

O único campo COMMINFO que deve ser configurado é o nome do objeto COMMINFO. Esse nome é então usado para identificar o objeto COMMINFO para um tópico. O campo **GRPADDR** do objeto COMMINFO deve ser verificado para assegurar que o valor seja um endereço de grupo multicast válido.

Objeto TOPIC

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. Para obter mais informações sobre os parâmetros do objeto TOPIC, consulte [DEFINE TOPIC](#).

Tópicos existentes podem ser usados com multicast alterando os valores dos seguintes parâmetros do objeto TOPIC: **COMMINFO** e **MCAST**.

- **COMMINFO** Este parâmetro especifica o nome do objeto de informações de comunicação multicast.
- **MCAST** Este parâmetro especifica se multicast é permitido nesta posição na árvore de tópicos. Por padrão, **MCAST** é configurado para **ASPPARENT**, o que significa que o atributo multicast do tópico é herdado do pai. Configurar **MCAST** para **ENABLED** permite tráfego multicast neste nó.

Redes e Tópicos Multicast

As informações a seguir são uma visão geral do que ocorre a assinaturas com diferentes tipos de definição de assinatura e tópico. Todos estes exemplos supõem que o parâmetro **COMMINFO** do objeto TOPIC esteja configurado para o nome de um objeto COMMINFO válido:

Conjunto de tópicos para multicast ativado

Se o parâmetro **MCAST** da sequência de tópicos for configurado para **ENABLED**, assinaturas de clientes com capacidade multicast são permitidas e uma assinatura multicast é feita, a menos que:

- Seja uma assinatura durável de um cliente com capacidade multicast.
- Seja uma assinatura não gerenciada de um cliente com capacidade multicast.
- Seja uma assinatura de um cliente com capacidade não multicast.

Nesses casos, é feita uma assinatura não multicast e essas assinaturas são submetidas a downgrade para publicação/assinatura normal.

Conjunto de tópicos para multicast desativado

Se o parâmetro **MCAST** da sequência de tópicos for configurado para **DISABLED**, uma assinatura não multicast sempre é feita e essas assinaturas são submetidas a downgrade para publicação/assinatura normal.

Conjunto de tópicos somente para multicast

Se o parâmetro **MCAST** da sequência de tópicos for configurado para **ONLY**, assinaturas de clientes com capacidade multicast são permitidas e uma assinatura multicast é feita, a menos que:

- É uma assinatura durável: assinaturas duráveis são rejeitadas com código de razão 2436 (0984) (RC2436): MQRC_DURABILITY_NOT_ALLOWED
- É uma assinatura não gerenciada: assinaturas não gerenciadas são rejeitadas com código de razão 2046 (07FE) (RC2046): MQRC_OPTIONS_ERROR
- É uma assinatura de um cliente não compatível com multicast: essas assinaturas são rejeitadas com código de razão 2560 (0A00) (RC2560): MQRC_MULTICAST_ONLY
- É uma assinatura de um aplicativo conectado localmente: essas assinaturas são rejeitadas com código de razão 2560 (0A00) (RC2560): MQRC_MULTICAST_ONLY

Windows

Linux

AIX

Visão geral do MQ Telemetry

O MQ Telemetry inclui um serviço de telemetria (MQXR) que faz parte de um gerenciador de filas, clientes de telemetria que você mesmo pode gravar ou fazer download gratuitamente, e interfaces da linha de comandos e administrativas do explorer. Telemetria refere-se à coleta de dados e à administração de um amplo intervalo de dispositivos remotos. Com o MQ Telemetry, é possível integrar a coleta de dados e o controle de dispositivos com aplicativos da web.

O MQ Telemetry é um componente do IBM MQ. O upgrade para essas versões é essencialmente a instalação de uma versão mais recente do IBM MQ.

Os aplicativos de amostra continuam disponíveis livremente no Eclipse Paho e MQTT.org. Consulte [Programas de amostra do IBM MQ Telemetry Transport](#).

Como MQ Telemetry é um componente de IBM MQ, MQ Telemetry pode ser instalado com o produto principal ou após o produto principal ter sido instalado. Para obter informações sobre migração, consulte [Migrando o MQ Telemetry no Windows](#) e [Migrando o MQ Telemetry no Linux](#).

Estão incluídos no MQ Telemetry os seguintes componentes:

Canais de Telemetria

Use os canais de telemetria para gerenciar a conexão de clientes MQTT para IBM MQ. Os canais de telemetria usam novos objetos IBM MQ, como SYSTEM.MQTT.TRANSMIT.QUEUE, para interagir com o IBM MQ.

Serviço de telemetria (MQXR)

Os clientes do MQTT usam o serviço de telemetria SYSTEM.MQXR.SERVICE para se conectar a canais de telemetria.

Suporte do IBM MQ Explorer para o MQ Telemetry

O MQ Telemetry pode ser administrado usando o IBM MQ Explorer.

Documentação

A documentação do MQ Telemetry está incluída na documentação do produto padrão do IBM MQ. A documentação do SDK para os clientes Java e C é fornecida na documentação do produto, e como Javadoc e HTML.

Conceitos de Telemetria

Você coleta informações do ambiente ao seu redor para decidir o que fazer. Como consumidor, você verifica o que há em uma loja antes de decidir o que comprar para comer. Você quer saber quanto tempo vai durar a viagem, se sair agora, antes de registrar uma conexão. Você verifica seus sintomas antes de decidir se vai agendar uma consulta médica. Você verifica quando um ônibus chegará antes de decidir se vai esperar por ele. As informações para essas decisões vêm diretamente de medidores e dispositivos, da palavra escrita no papel ou de uma tela ou de você. Em qualquer lugar, e em qualquer hora, você coleta informações, reúne todas elas, analisa uma por uma e toma uma ação.

Se as fontes de informações forem muito dispersas ou inacessíveis, será mais difícil e mais caro coletar informações exatas. Se você quiser fazer muitas mudanças ou se for difícil fazê-las, elas não serão feitas ou serão feitas quando forem menos efetivas.

E se os custos da coleção e do controle de informações de dispositivos totalmente dispersos forem consideravelmente reduzidos por meio da conexão de dispositivos com tecnologia digital à Internet?

Essas informações podem ser analisadas usando os recursos da Internet e da empresa. Você tem mais oportunidades de tomar decisões mais informadas e de agir sobre elas.

Tendências tecnológicas, e pressões econômicas e ambientais, estão fazendo estas mudanças acontecerem:

1. O custo da conexão e do controle de sensores e atuadores está reduzindo devido à normatização e à conexão com processadores digitais de baixo custo.
2. A Internet, e suas tecnologias, estão sendo cada vez mais usadas para a conexão com dispositivos. Em alguns países, telefones celulares excedem computadores pessoais em número de conexões com aplicativos de Internet. E certamente outros dispositivos seguem atrás.
3. A Internet, e suas tecnologias, facilitam a obtenção de dados para um aplicativo. O fácil acesso a dados está conduzindo o uso de análise de dados para transformar dados de sensores em informações que são úteis em muitas outras soluções.
4. O uso inteligente de recursos é muitas vezes a forma mais rápida e barata de reduzir custos e emissões de carbono. As alternativas: localizar novos recursos ou desenvolver novas tecnologias para usar os recursos existentes, pode ser uma solução a longo prazo. A curto prazo, desenvolver novas tecnologias ou localizar novos recursos, muitas vezes é mais arriscado, mais demorado e mais caro do que melhorar soluções existentes.

exemplo

Um exemplo mostra como essas tendências criam novas oportunidades de interação com o ambiente de forma inteligente.

O International Convention for the Safety of Life at Sea (SOLAS) requer que um Automatic Identification System (AIS) seja implementado em vários navios. Ele é requerido em navios mercantes com mais de 300 toneladas e em navios de passageiros. Um AIS é principalmente um sistema de prevenção de colisões para navios costeiros. Ele é usado por autoridades marinhas para monitorar e controlar águas costeiras.

Entusiastas em todo o mundo estão implementando estações de rastreamento AIS de baixo custo e colocando informações sobre navios costeiros na Internet. Outros entusiastas estão compondo aplicativos que combinam informações do AIS com outras informações da Internet. Os resultados são colocados em Web sites e publicados usando Twitter e SMS.

Em um aplicativo, as informações das estações AIS próximas de Southampton são combinadas com informações geográficas e de propriedade dos navios. O aplicativo alimenta informações em tempo real sobre chegadas e partidas de balsas no Twitter. Pessoas que se deslocam para o trabalho regularmente usando as embarcações entre Southampton e a Ilha de Wight assinam o feed de notícias usando Twitter ou SMS. Se o feed mostrar que a balsa está atrasada, os computadores poderão atrasar a sua partida e selecionar a mesma quando ela atracar após o seu horário de chegada planejado.

Para ver mais exemplos, consulte [“Casos de Uso de Telemetria”](#) na página 114.

Tarefas relacionadas

[Instalando MQ Telemetry](#)

[Administrando MQ Telemetry](#)

[Migrando o MQ Telemetry no Windows](#)

[Migrando o MQ Telemetry no Linux](#)

[Desenvolvendo aplicativos para o MQ Telemetry](#)

[Resolução de problemas do MQ Telemetry ..](#)

Referências relacionadas

[Referência do MQ Telemetry](#)

Windows

Linux

AIX

Introdução ao MQ Telemetry

Pessoas, empresas e governos desejam cada vez mais usar o MQ Telemetry para interagir de maneira mais inteligente com o ambiente no qual vivemos e trabalhamos. O MQ Telemetry conecta todos os tipos

de dispositivos à Internet e à empresa e reduz os custos da construção de aplicativos para dispositivos inteligentes.

O que é MQ Telemetry?

- Ele é um recurso do IBM MQ que estende o backbone de sistema de mensagens universal fornecido pelo IBM MQ para um amplo intervalo de sensores, acionadores e dispositivos de telemetria remotos. O MQ Telemetry estende o IBM MQ para que ele possa interconectar aplicativos corporativos inteligentes, serviços e tomadores de decisão com redes de dispositivos instrumentados.
- As partes principais do MQ Telemetry são:

O serviço do MQ Telemetry (MQXR).

Este serviço é executado no servidor IBM MQ e usa o protocolo IBM MQ Telemetry Transport (MQTT) para se comunicar com dispositivos de telemetria.

Aplicativos MQTT que você grava.

Estes aplicativos controlam as informações que são transportadas entre os dispositivos de telemetria e o gerenciador de filas do IBM MQ e as ações que são executadas em resposta a essas informações. Para ajudar a criar esses aplicativos, use as bibliotecas do cliente MQTT.

O Que Ele Pode Fazer por Mim?

- MQTT é um transporte de mensagens aberto que permite que as implementações do MQTT sejam criadas para uma ampla variedade de dispositivos.
- Os clientes MQTT podem ser executados em pequenos dispositivos de área de cobertura que possuem recursos limitados.
- O MQTT funciona de forma eficiente em redes nas quais a largura da banda é baixa, onde o custo de envio de dados é caro ou que pode ser frágil.
- A entrega da mensagem é garantida e separada do aplicativo.
- Os programadores de aplicativos não precisam ter conhecimento de programação de comunicações.
- As mensagens podem ser trocadas com outros aplicativos de sistema de mensagens. Estes podem ser outro aplicativo de telemetria ou um MQI, JMS ou aplicativo do sistema de mensagens corporativo.

Como eu Utilizo?

- Scripts de amostra são fornecidos que funcionam com um aplicativo cliente de amostra do IBM MQ Telemetry Transport v3 (mqttv3app.jar). Consulte [Programas e amostras do IBM MQ Telemetry Transport](#).
- Use o IBM MQ Explorer e suas ferramentas associadas para administrar o recurso de telemetria do IBM MQ.
- Use as bibliotecas de cliente para ajudá-lo a criar os aplicativos MQTT que se conectam a um gerenciador de filas e que usam sistema de mensagens de publicação/assinatura.
- Distribua seu aplicativo e a biblioteca de cliente para o dispositivo no qual seu aplicativo será executado.

Como Funciona?

- MQTT é um protocolo de publicação e assinatura. Um aplicativo cliente MQTT pode publicar mensagens em um servidor MQTT ou assinar para mensagens que são enviadas por aplicativos que se conectam a um servidor MQTT.
- Os aplicativos clientes do MQTT usam bibliotecas de cliente que implementam o transporte de mensagens do MQTT.
- Um aplicativo cliente do MQTT funciona como um cliente MQ padrão, mas pode ser executado em uma variedade muito mais ampla de plataformas e redes.
- O serviço MQ Telemetry (MQXR) transforma um gerenciador de filas do IBM MQ em um servidor MQTT.

- Quando um gerenciador de filas do IBM MQ age como o servidor MQTT, outros aplicativos que se conectam ao gerenciador de filas podem assinar e receber as mensagens do cliente MQTT.
- O gerenciador de filas age como o roteador que distribui mensagens de aplicativos de publicação para aplicativos de assinatura.
- As mensagens podem ser distribuídas entre diferentes tipos de aplicativos clientes. Por exemplo, entre clientes de telemetria e clientes JMS.

Nota: O MQ Telemetry substitui os nós SCADA que foram retirados na versão 7 do WebSphere Message Broker (agora conhecido como IBM Integration Bus) e é executado no Windows, Linux e AIX.

Windows

Linux

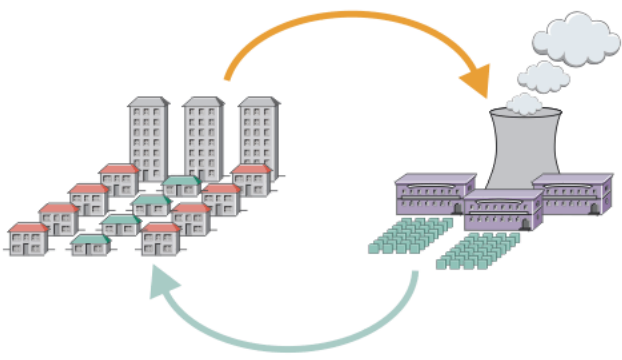
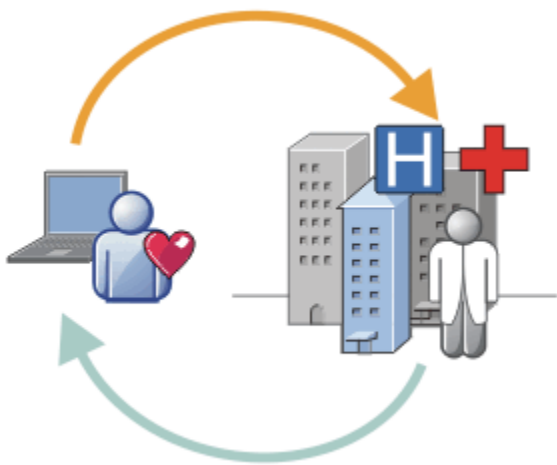
AIX

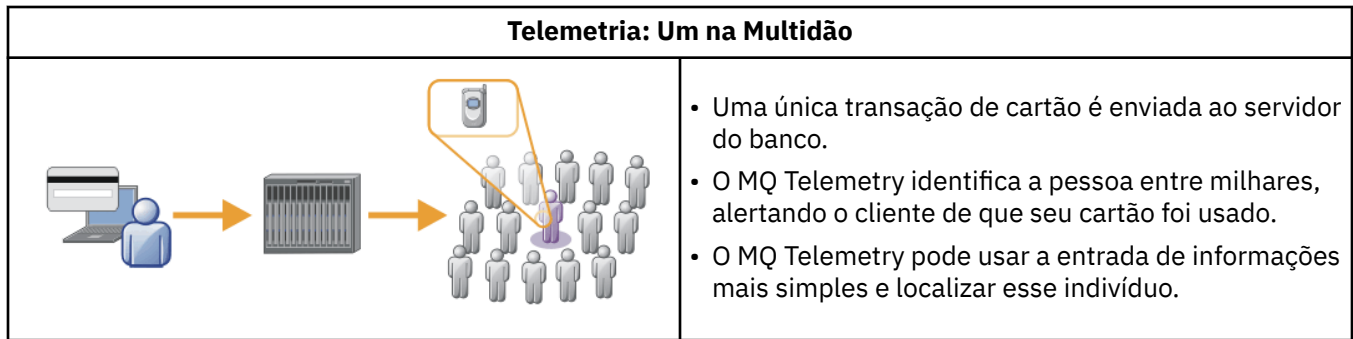
Casos de Uso de Telemetria

A Telemetria é uma tecnologia que permite sensoriamento, medição de dados e controle de dispositivos remotos de forma automatizada. A ênfase está na transmissão de dados a partir de dispositivos para um ponto de controle central. A telemetria também inclui o envio de informações de configuração e controle para dispositivos.

O MQ Telemetry conecta pequenos dispositivos usando o MQTT protocol e conecta os dispositivos com outros aplicativos usando IBM MQ. MQ Telemetry liga os dispositivos e a internet tornando mais fácil construir "soluções inteligentes". Soluções inteligentes desvendam a riqueza de informações disponíveis na Internet, e nos aplicativos corporativos, para aplicativos que monitoram e controlam dispositivos.

Os diagramas a seguir demonstram alguns usos típicos do MQ Telemetry:

Telemetria: Eletricidade Inteligente	
	<ul style="list-style-type: none"> • Mensagem do MQTT contendo dados de uso de energia enviados ao provedor de serviços. • O MQ Telemetry envia CONTROL COMMANDS com base na análise de dados de uso de energia. • Para obter mais informações, consulte o caso de uso a seguir: “Caso de Uso de Telemetria: Controle e Monitoramento de Energia Doméstica” na página 117
Telemetria: Serviços de Saúde Inteligentes	
<ul style="list-style-type: none"> • O MQ Telemetry envia dados de saúde para o seu médico e hospital. • Alertas ou feedback de mensagens do MQTT podem ser enviados com base na análise dos Dados de saúde. • Para obter mais informações, consulte o caso de uso a seguir: “Caso de Uso de Telemetria: Monitoramento de Pacientes no Lar” na página 115 	



Os casos de uso descritos nos subtópicos são obtidos de exemplos reais. Eles ilustram algumas maneiras de usar telemetria, e alguns dos problemas comuns que a tecnologia de telemetria deve resolver.

Windows

Linux

AIX

Caso de Uso de Telemetria: Monitoramento de Pacientes no Lar

Na colaboração entre a IBM e um provedor de assistência médica em um sistema de saúde para pacientes cardíacos, um desfibrilador cardioversor implantável se comunica com um hospital. Dados sobre o paciente e o dispositivo implantado são transferidos com o uso de telemetria RF para o dispositivo MQTT na casa de um paciente.

Normalmente a transferência acontece toda noite para um transmissor localizado ao lado. O transmissor transfere os dados com segurança por meio de sistema de telefonia para o hospital, onde os dados são analisados.

O sistema reduz o número de visitas que um paciente deve fazer ao médico. Ele detecta quando o paciente ou o dispositivo precisa de atenção, e em caso de emergência, alerta o médico de plantão.

A colaboração entre a IBM e o provedor de assistência médica tem características que são comuns para um número de casos de uso de telemetria :

Invisibilidade

O dispositivo não requer intervenção do usuário além de fornecer energia, uma linha telefônica e estar próximo do dispositivo em uma parte do dia. Sua operação é confiável e ele é simples de usar.

Para remover a necessidade de o paciente configurar o dispositivo, o fornecedor do dispositivo faz sua pré-configuração. O paciente só precisa ligá-lo. A eliminação de configuração por parte do paciente simplifica a operação do dispositivo e reduz as chances de ele ser configurado de maneira incorreta.

O cliente MQTT é integrado como parte do dispositivo. O desenvolvedor do dispositivo integra a implementação do cliente MQTT no dispositivo e o desenvolvedor ou fornecedor configura o cliente MQTT como parte da pré-configuração.

O cliente MQTT é enviado como um arquivo JAR do Java SE, que o desenvolvedor inclui em seu aplicativo Java. Para ambientes não-Java, como este, o desenvolvedor do dispositivo pode implementar um cliente em uma linguagem diferente usando os formatos do MQTT e protocolos publicados. Alternativamente, o desenvolvedor pode usar um dos clientes C fornecidos como bibliotecas compartilhadas para as plataformas Windows, Linux e ARM.

Conectividade irregular

A comunicação entre o desfibrilador e o hospital tem características de rede irregulares. Duas redes diferentes são usadas para resolver problemas diferentes de coleção de dados do paciente e de envio de dados para o hospital. Entre a patente e o dispositivo MQTT, uma rede RF de baixa potência e de curto alcance é usada. O transmissor se conecta ao hospital usando uma conexão de VPN TCP/IP por uma linha telefônica de baixa largura de banda.

Muitas vezes não é prático encontrar uma maneira de conectar cada dispositivo diretamente a uma rede de protocolo da Internet. O uso de duas redes, conectadas por um hub, é uma solução comum.

O dispositivo MQTT é um hub simples, que armazena informações do paciente e encaminha para o hospital.

Segurança

O médico deve estar preparado para confiar na autenticidade dos dados do paciente, e o paciente quer que a privacidade de seus dados seja respeitada.

Em algumas situações, é suficiente criptografar a conexão, usando VPN ou TLS. Em outras situações, é desejável manter os dados seguros, mesmo após eles terem sido armazenados.

Às vezes o dispositivo de telemetria não está seguro. Ele pode estar em uma residência compartilhada, por exemplo. O usuário do dispositivo deve ser autenticado para assegurar que os dados sejam do paciente correto. O dispositivo em si pode ser autenticado para o servidor usando TLS e o servidor autenticado para o dispositivo.

O canal de telemetria entre dispositivo e o gerenciador de filas suporta JAAS para autenticação do usuário e TLS para criptografia de comunicação e autenticação de dispositivo. O acesso a uma publicação é controlado pelo gerenciador de autoridade de objeto no IBM MQ.

O identificador usado para autenticação do usuário pode ser mapeado para um identificador diferente, como a identidade de um paciente comum. Um identificador comum simplifica a configuração da autorização para tópicos de publicação no IBM MQ.

Conectividade

A conexão entre o dispositivo MQTT e o hospital usa discagem, e trabalha com largura de banda baixa, como 300 bauds.

Para operar efetivamente a 300 baud, o MQTT protocol inclui apenas alguns bytes extras em uma mensagem além dos cabeçalhos TCP/IP.

O MQTT protocol fornece o sistema de mensagens *fire-forget* de transmissão única, que mantém as latências baixas. Ele também poderá usar múltiplas transmissões para garantir *pelo menos uma vez e exatamente uma* entrega, se a entrega garantida for mais importante do que o tempo de resposta. Para garantia de entrega, as mensagens são armazenadas no dispositivo até terem sido entregues com sucesso. Se um dispositivo for conectado sem fio, a garantia de entrega será especialmente útil.

Escalabilidade

Os dispositivos de telemetria normalmente são implementados em grandes números, de dezenas de milhares a milhões.

A conexão de vários dispositivos a um sistema representa grandes demandas em uma solução. Existem demandas de negócios, como o custo dos dispositivos e seu software, e demandas de administração, como gerenciamento de licenças, dispositivos e usuários. Demandas técnicas incluem carregamentos na rede e nos servidores.

Abrir uma conexão exige mais recursos do servidor do que manter as conexões abertas. Mas em um caso de uso como esse, que usa linhas telefônicas, as despesas de conexão significam que as conexões são deixadas abertas não mais do que o necessário. As transferências de dados são feitas basicamente em lote. As conexões podem ser planejadas para a noite toda para evitar picos repentinos de conexões na hora de dormir.

No cliente, a escalabilidade dos clientes é beneficiada pelo mínimo de configuração de cliente requerida. O cliente de MQTT é integrado ao dispositivo. Não há requisitos de configuração ou etapa de aceitação da licença do cliente MQTT para integração com a implementação de dispositivos para pacientes.

No servidor, o MQ Telemetry tem uma meta inicial de 50.000 conexões abertas por gerenciador de filas.

As conexões são gerenciadas usando o IBM MQ Explorer. O IBM MQ Explorer filtra as conexões a serem exibidas para um número gerenciável. Com um esquema, adequadamente escolhido, para alocação de identificadores para clientes, você pode filtrar conexões com base na geografia ou em ordem alfabética por nome de paciente.

Monitoramento de Energia Doméstica

Medidores inteligentes coletam mais detalhes sobre o consumo de energia do que os medidores tradicionais.

Os medidores inteligentes muitas vezes são acoplados a uma rede de telemetria local para monitorar e controlar dispositivos individuais em uma casa. Alguns também são conectados remotamente para monitoramento e controle a uma certa distância.

A conexão remota poderia ser configurada por um indivíduo, por um utilitário de energia ou por um ponto de controle central. O ponto de controle remoto pode ler o uso de energia e fornecer dados de uso. Ele pode fornecer dados para influenciar o uso, como informações contínuas sobre o tempo e preço. Ele pode limitar a carga para melhorar a eficiência de geração de energia geral.

Medidores inteligentes estão começando a ser amplamente implementados. O governo do Reino Unido, por exemplo, organizou um conselho para discutir sobre a implementação de medidores inteligentes em cada domicílio até 2020.

Os casos de uso de medidor do lar possuem um número de características comuns:

Invisibilidade

A menos que o usuário queira se envolver na economia de energia usando o medidor, o medidor não requer intervenção do usuário. Isso não deve reduzir a confiabilidade do fornecimento de energia para dispositivos individuais.

Um cliente MQTT pode ser integrado no software implementado com o medidor e não requer instalação nem configuração separadas.

Conectividade irregular

A comunicação entre dispositivos e medidores inteligentes demanda normas de conectividade diferentes das que existem entre o medidor e o ponto de conexão remota.

A conexão do medidor inteligente com dispositivos deve ser altamente disponível e compatível com as normas de rede para uma rede doméstica.

É provável que a rede remota use várias conexões físicas. Algumas delas, como um celular, têm um alto custo de transmissão e podem ser intermitentes. A especificação MQTT v3 tem como objetivo as conexões remotas e as conexões entre adaptadores locais e o medidor inteligente.

As conexões entre tomadas de energia e dispositivos e o medidor usam uma rede doméstica, como Zigbee. MQTT para redes de sensores (MQTT-S) foi projetado para trabalhar com Zigbee e outros protocolos de rede de baixa largura de banda. MQ Telemetry não suporta MQTT-S diretamente. Ele requer um gateway para conectar o MQTT-S ao MQTT v3.

Assim como o monitoramento de pacientes em casa, as soluções para monitoramento e controle de energia doméstica requerem diversas redes, conectadas usando um medidor inteligente como um hub.

Segurança

Há inúmeros problemas de segurança associados aos medidores inteligentes. Esses problemas incluem não repúdio de transações, autorização de todas as ações de controle que são iniciadas e privacidade dos dados de consumo de energia.

Para assegurar a privacidade, os dados transferidos entre o medidor e o ponto de controle remoto pelo MQTT podem ser criptografados usando TLS. Para assegurar a autorização de ações de controle, a conexão do MQTT entre o medidor e o ponto de controle remoto pode ser mutuamente autenticada usando TLS.

Conectividade

A natureza física da rede remota pode variar consideravelmente. Ela pode usar uma conexão de banda larga existente, ou uma rede remota com alto custo de chamadas e disponibilidade

intermitente. O MQTT é um protocolo eficiente e confiável para conexões intermitentes e de alto custo. Consulte [“Caso de Uso de Telemetria: Monitoramento de Pacientes no Lar”](#) na página 115.

Escalabilidade

Eventualmente as empresas de energia, ou pontos de controle central, planejam a implementação de dezenas de milhares de medidores inteligentes. Inicialmente, os números de medidores por implementação estão entre dezenas e centenas de milhares. Esse número é comparável com o destino inicial do MQTT de 50.000 conexões de cliente abertas por gerenciador de filas.

Um aspecto crítico da arquitetura para monitoramento e controle de energia doméstica é o uso do medidor inteligente como um concentrador de rede. Cada adaptador de dispositivo é um sensor separado. Por eles são conectados a um hub local usando o MQTT, o hub pode concentrar os fluxos de dados em uma única sessão TCP/IP com o ponto de controle central e também armazenar mensagens por um curto período para superar indisponibilidades na sessão.

As conexões remotas devem ser deixadas abertas em casos de uso de energia doméstica por duas razões. Primeiro porque a abertura de conexões leva muito tempo em relação ao envio de solicitações. O tempo que se leva para abrir várias conexões para enviar solicitações de "limitação de carga" em um curto intervalo é muito longo. Segundo, para receber solicitações de limitação de carga da empresa de energia, primeiro a conexão deve ser aberta pelo cliente. Com o MQTT, as conexões são sempre iniciadas pelo cliente, e para receber solicitações de limitação de carga da empresa de energia, a conexão deve ser deixada aberta.

Se a taxa de abertura de conexões for crítica ou se o servidor iniciar solicitações com tempo crítico, normalmente a solução será manter várias conexões abertas.

Windows Linux AIX **Casos de Uso de Telemetria: Radio Frequency Identification (RFID)**

RFID é o uso de uma identificação RFID integrada para identificar e controlar um objeto por meio de modo wireless. Identificações RFID podem ser lidas a vários metros de distância e fora da linha de visão do leitor de RFID. Identificações passivas são ativadas por um leitor de RFID. Identificações ativas fazem transmissões sem ativação externa. As identificações ativas devem ter uma fonte de alimentação. As identificações passivas podem incluir uma fonte de alimentação para aumentar sua distância.

O RFID é usado em vários aplicativos e os tipos de casos de uso variam bastante. Os casos de uso do RFID e os casos de monitoramento de pacientes no lar e monitoramento e controle de energia do lar têm algumas semelhanças e diferenças.

Invisibilidade

Em muitos casos de uso, o leitor de RFID é implementado em grandes números e deve trabalhar sem intervenção do usuário. O leitor inclui um cliente integrado do MQTT para se comunicar com um ponto de controle central.

Por exemplo, em um armazém de distribuição, um leitor usa um sensor de movimentos para detectar uma paleta. Ele ativa identificações RFID de itens na paleta e envia dados e solicitações para aplicativos centrais. Os dados são usados para atualização do local do estoque. As solicitações controlam o que acontece com a paleta em seguida, como sua mudança para um compartimento específico. Sistemas de bagagens de aeroportos e linhas aéreas estão usando RFID dessa forma.

Em alguns casos de uso do RFID, o leitor possui um ambiente de computação padrão, como Java Platform, Micro Edition (Java ME). Nesses casos, o cliente do MQTT pode ser implementado em uma etapa de configuração distinta, após a fabricação.

Conectividade irregular

Os leitores de RFID podem ser separados do dispositivo de controle local que contém um cliente MQTT, ou cada leitor pode integrar um cliente MQTT. Normalmente, fatores geográficos ou de comunicações indicam a escolha da topologia.

Segurança

Privacidade e autenticidade são questões de segurança para a anexação de identificações RFID. As identificações RFID são reservadas e podem ser monitoradas, falsificadas e corrompidas secretamente.

A solução dos problemas de segurança de RFID aumenta a oportunidade de implementação de novas soluções de RFID. Embora haja exposição da segurança na identificação RFID, e no leitor local, o uso de processamento de informações central sugere abordagens para reagir a diferentes ameaças. Por exemplo, a violação de uma identificação pode ser detectada correlacionando dinamicamente os níveis de estoque com as entregas e os envios.

Conectividade

Normalmente os aplicativos RFID envolviam armazenamento e encaminhamento em lote de informações reunidas em consultas imediatas e leitores de RFID. No caso de uso de armazém de distribuição, o leitor de RFID fica conectado o tempo todo. Quando uma identificação é lida, ela é publicada junto com as informações sobre o leitor. O aplicativo warehousing publica a resposta de volta para o leitor.

No aplicativo warehousing, a rede é geralmente confiável e as solicitações imediatas podem usar mensagens *fire-forget* para desempenho de baixa latência. Os dados de armazenamento e encaminhamento em lote podem usar o sistema de mensagens *exatamente uma vez* para minimizar os custos de administração associados à perda de dados.

Escalabilidade

Se o aplicativo RFID requerer respostas imediatas, em um ou dois segundos, os leitores de RFID deverão ficar conectados.

Casos de Uso de Telemetria: Sensoriamento de Ambiente

O sensoriamento de ambiente usa telemetria para coletar informações sobre qualidade e níveis de água dos rios, poluentes atmosféricos e outros dados ambientes.

Os sensores localizam-se frequentemente em locais remotos, sem acesso à comunicação com fio. A largura de banda wireless é cara e a confiabilidade pode ser baixa. Normalmente inúmeros sensores de ambiente em uma pequena área geográfica são conectados a um dispositivo de monitoramento local em um local seguro. As conexões locais podem ser com e sem fio.

Invisibilidade

É provável que dispositivos sensores sejam menos acessíveis, tenham potência mais baixa e sejam implementados em maiores números do que um dispositivo de monitoramento central. Às vezes os sensores não conseguem se comunicar e o dispositivo de monitoramento local inclui adaptadores para transformar e armazenar dados do sensor. O dispositivo de monitoramento provavelmente irá incorporar um computador de propósito geral que suporta o Java Platform, Standard Edition (Java SE) ou o Java Platform, Micro Edition (Java ME). A invisibilidade dificilmente será um grande requisito ao configurar o cliente do MQTT.

Conectividade irregular

A capacidade dos sensores e o custo e a largura de banda da conexão remota normalmente resultam em um hub de monitoramento local conectado a um servidor central.

Segurança

A menos que a solução esteja sendo usada em um caso de uso militar ou de defesa, a segurança não é o principal requisito.

Conectividade

Muitos usos não requerem monitoramento contínuo ou disponibilidade imediata dos dados. Dados de exceção, como um alerta de nível de enchente, não precisam ser encaminhados diretamente. Dados do sensor são agregados no monitor local para reduzir custos de conexão e comunicação e depois

transferidos com o uso de conexões planejadas. Dados de exceção são encaminhados assim que detectados no monitor.

Escalabilidade

Os sensores estão concentrados ao redor de hubs locais e seus dados estão agregados em pacotes que são transmitidos de acordo com um planejamento. Ambos esses fatores reduzem a carga no servidor central que seria imposta pelo uso direto de sensores conectados.

Windows Linux AIX Casos de Uso de Telemetria: Aplicativos

Remotos

Aplicativos remotos são aqueles executados em dispositivos wireless. Os dispositivos são plataformas de aplicativo genéricas ou dispositivos customizados.

Plataformas gerais incluem dispositivos portáteis, como telefones, Personal Data Assistants e notebooks. Dispositivos customizados usam hardware de propósito especial padronizado para aplicativos específicos. Um dispositivo para registrar a entrega de pacotes "assinada" é um exemplo de dispositivo móvel customizado. Aplicativos em dispositivos móveis customizados muitas vezes são baseados em plataformas de software genéricas.

Invisibilidade

A implementação de aplicativos móveis customizados é gerenciada e pode incluir configuração do aplicativo cliente MQTT. Invisibilidade não deve ser o principal requisito durante a configuração do cliente de MQTT.

Conectividade irregular

Ao contrário da topologia de hub local dos casos de uso anteriores, os clientes móveis se conectam remotamente. A camada do aplicativo cliente se conecta diretamente a um aplicativo no hub central.

Segurança

Com pouca segurança física, o dispositivo móvel e o usuário remoto devem ser autenticados. O TLS é usado para confirmar a identidade do dispositivo e o JAAS para autenticar o usuário.

Conectividade

Se o aplicativo remoto depender de cobertura wireless, ele deverá ter capacidade para operar no modo off-line e para lidar efetivamente com uma conexão interrompida. Nesse ambiente, o objetivo é ficar conectado, mas o aplicativo deve ter capacidade de armazenar e encaminhar mensagens. Muitas vezes as mensagens são pedidos ou confirmações de entrega, e têm valor comercial importante. Elas precisam ser armazenadas e encaminhadas de forma confiável.

Escalabilidade

A escalabilidade não é um grande problema. O número de aplicativos clientes não deve exceder milhares ou dezenas de milhares, em casos de uso de aplicativo remoto customizado.

Windows Linux AIX Conectando Dispositivos de Telemetria a um Gerenciador de Filas

Dispositivos de telemetria se conectam a um gerenciador de filas usando um cliente MQTT v3. O cliente MQTT v3 usa TCP/IP para se conectar a um listener TCP/IP chamado de serviço de telemetria (MQXR).

Quando você conectar um dispositivo de telemetria a um gerenciador de filas, o cliente do MQTT iniciará uma conexão TCP/IP usando o método `MqttClient.connect`. Como os clientes IBM MQ, um cliente MQTT deve ser conectado ao gerenciador de filas para enviar e receber mensagens. A conexão é estabelecida no servidor usando um listener TCP/IP, instalado com o MQ Telemetry, chamado de serviço de telemetria (MQXR). Cada gerenciador de filas executa no máximo um serviço de telemetria (MQXR).

O serviço de telemetria (MQXR) usa o endereço de soquete remoto configurado por cada cliente no método `MqttClient.connect` para alocar a conexão com um canal de telemetria. Um endereço de

soquete é a combinação do número da porta e do nome do host TCP/IP. Diversos clientes que usam o mesmo endereço de soquete remoto são conectados ao mesmo canal de telemetria pelo serviço de telemetria (MQXR).

Se houver diversos gerenciadores de filas em um servidor, divida os canais de telemetria entre os gerenciadores de filas. Aloque os endereços de soquete remotos entre os gerenciadores de filas. Defina cada canal de telemetria com um endereço de soquete remoto exclusivo. Dois canais de telemetria não devem usar o mesmo endereço de soquete.

Se o mesmo endereço de soquete remoto for configurado para canais de telemetria em diversos gerenciadores de filas, o primeiro canal de telemetria para conexão tem precedência. Os canais subsequentes que se conectam no mesmo endereço falham.

Se houver diversos adaptadores de rede no servidor, divida os endereços de soquete remotos entre canais de telemetria. A alocação de endereços de soquete é totalmente arbitrária, contanto que todo endereço de soquete específico seja configurado em apenas um canal de telemetria.

Configure o IBM MQ para conectar clientes MQTT usando os assistentes fornecidos no suplemento do MQ Telemetry para o IBM MQ Explorer. Alternativamente, siga as instruções em [Configurando um Gerenciador de Filas para Telemetria em Linux e AIX](#) e [Configurando um Gerenciador de Filas para Telemetria em Windows](#) para configurar a telemetria manualmente.

Referências relacionadas

[Propriedades MQXR](#)

Windows

Linux

AIX

Protocolos de Conexão do Telemetry

O MQ Telemetry suporta TCP/IP IPv4 e IPv6 e TLS.

Windows

Linux

AIX

Serviço de telemetria (MQXR)

O serviço de telemetria (MQXR) é um listener TCP/IP que é gerenciado como um serviço do IBM MQ. Crie o serviço usando um assistente do IBM MQ Explorer ou com um comando **runmqsc**.

O serviço MQ Telemetry (MQXR) é chamado SYSTEM.MQXR.SERVICE ..

O assistente de Configuração de amostra de telemetria, fornecido na função do MQ Telemetry para o IBM MQ Explorer, cria o serviço de telemetria e um canal de telemetria de amostra; veja [Verificando a instalação do MQ Telemetry usando o IBM MQ Explorer](#) .

Crie a configuração de amostra a partir da linha de comandos; consulte [Verificando a instalação do MQ Telemetry usando a linha de comandos](#).

O serviço de telemetria (MQXR) começa a para automaticamente com o gerenciador de filas. Controle o serviço usando a pasta de serviços no IBM MQ Explorer. Para ver o serviço, deve-se clicar no ícone para parar IBM MQ Explorer a filtragem de objetos SYSTEM da exibição.

Para obter um exemplo de como criar o serviço manualmente, consulte

- [Linux](#) [AIX](#) [Criando o SYSTEM.MQXR.SERVICE no Linux](#)
- [Windows](#) [Criando o SYSTEM.MQXR.SERVICE no Windows](#)

V 9.3.0 Do IBM MQ 9.3.0 em diante, [Criando o SYSTEM.MQXR.SERVICE no Linux](#) e [Criando o SYSTEM.MQXR.SERVICE no Windows](#) são atualizados para especificar a chave padrão para requerer que passphrases para os canais TLS do MQTT sejam criptografados. Para obter mais informações, consulte [Criptografia de senha para canais TLS do MQTT](#).

Windows

Linux

AIX

Canais de Telemetria

Crie canais de telemetria para criar conexões com propriedades diferentes, como Java Authentication and Authorization Service (JAAS) ou autenticação TLS, ou para gerenciar grupos de clientes.

Crie canais de telemetria usando o assistente **New Telemetry Channel**, fornecido na função MQ Telemetry para IBM MQ Explorer. Configure um canal, usando o assistente, para aceitar conexões de clientes MQTT em uma porta TCP/IP específica. Desde a IBM WebSphere MQ 7.1, é possível configurar o MQ Telemetry usando o programa da linha de comandos, **runmqsc**.

Crie diversos canais de telemetria, em diferentes portas, para tornar grandes números de conexões do cliente fáceis de gerenciar, dividindo os clientes em grupos. Cada canal de telemetria tem um nome diferente.

É possível configurar canais de telemetria com diferentes atributos de segurança para criar diferentes tipos de conexão. Crie diversos canais para aceitar conexões do cliente em diferentes endereços TCP/IP. Use o TLS para criptografar mensagens e autenticar o canal de telemetria e o cliente. Consulte [Configuração do TLS de clientes e canais de telemetria do MQTT](#). Especifique o ID do usuário para simplificar a autorização do acesso aos objetos do IBM MQ. Especifique uma configuração do JAAS para autenticar o usuário do MQTT com o JAAS. Consulte [Identificação, autorização e autenticação do cliente MQTT](#).

Windows

Linux

AIX

Protocolo IBM MQ Telemetry Transport

O protocolo IBM MQ Telemetry Transport (MQTT) v3 foi projetado para trocar mensagens entre pequenos dispositivos em baixa largura de banda, ou conexões caras, e para enviar mensagens no modo confiável. Ele usa TCP/IP.

O MQTT protocol está publicado; consulte [Formato e protocolo do IBM MQ Telemetry Transport](#). A versão 3 do protocolo usa publicação/assinatura e suporta três qualidades de serviço: *fire-forget*, *pelo menos uma vez* e *exatamente uma vez*.

O tamanho pequeno dos cabeçalhos do protocolo e a carga útil da mensagem da matriz de bytes mantêm as mensagens pequenas. Os cabeçalhos abrangem um cabeçalho fixo de 2 bytes e até 12 bytes de cabeçalhos variáveis adicionais. O protocolo usa cabeçalhos variáveis de 12 bytes para se inscrever e se conectar e cabeçalhos variáveis de apenas 2 bytes para a maioria das publicações.

Com três qualidades de serviço, é possível fazer trocas entre baixa latência e confiabilidade; consulte [Qualidades de serviço fornecidas por um cliente MQTT](#). *Fire-forget* não usa nenhum armazenamento de dispositivo persistente e apenas uma transmissão para enviar ou receber uma publicação. *Pelo menos uma vez* e *exatamente uma vez* requerem armazenamento persistente no dispositivo para manter o estado do protocolo e salvar uma mensagem até que seja reconhecida.

Windows

Linux

AIX

Clientes MQTT

Um aplicativo do cliente MQTT é responsável por coletar informações do dispositivo de telemetria, conectando-se ao servidor e publicando as informações para o servidor. Ele também pode se inscrever a tópicos, receber publicações e controlar o dispositivo de telemetria.

Diferente dos aplicativos do cliente IBM MQ, os aplicativos do cliente MQTT não são aplicativos IBM MQ. Eles não especificam um gerenciador de filas para conexão. Eles não se limitam a usar interfaces de programação específicas do IBM MQ. Em vez disso, clientes do MQTT implementam o protocolo MQTT 3. É possível escrever sua própria biblioteca do cliente para interface com o MQTT protocol na linguagem de programação e na plataforma de sua escolha. Consulte [IBM MQ Telemetry Transport formato e protocolo](#).

Para simplificar a gravação de aplicativos clientes do MQTT, use as bibliotecas do cliente C, Java, e JavaScript que encapsulam o MQTT protocol para diversas plataformas. Se você incorporar essas bibliotecas em seus apps MQTT, um cliente do MQTT totalmente funcional poderá ser tão curto quanto 15 linhas de código. As bibliotecas do cliente do MQTT estão disponíveis livremente por meio do Eclipse Paho e do MQTT.org. Consulte [Programas de amostra do IBM MQ Telemetry Transport](#).

O aplicativo do cliente MQTT é sempre responsável por iniciar uma conexão com um canal de telemetria. Após a conexão, o aplicativo cliente do MQTT ou um aplicativo IBM MQ pode iniciar uma troca de mensagens.

Os aplicativos do cliente MQTT e os aplicativos do IBM MQ publicam e assinam o mesmo conjunto de tópicos. Um aplicativo IBM MQ também pode enviar uma mensagem diretamente para um

aplicativo cliente MQTT sem que o aplicativo cliente crie primeiro uma assinatura. Consulte [Configurar enfileiramento distribuído para enviar mensagens para clientes MQTT](#).

Aplicativos do cliente MQTT são conectados ao IBM MQ usando um canal de telemetria. O canal de telemetria age como uma ponte entre os diferentes tipos de mensagem usados pelo MQTT e IBM MQ. Ele cria publicações e assinaturas no gerenciador de filas em nome do cliente do aplicativo MQTT. O canal de telemetria envia publicações que correspondem às assinaturas de um aplicativo do cliente MQTT do gerenciador de filas para o aplicativo do cliente MQTT.

Windows

Linux

AIX

Envie uma mensagem para um cliente MQTT

Os aplicativos IBM MQ podem enviar mensagens de clientes do MQTT v3 publicando em assinaturas criadas por clientes ou enviando mensagens diretamente. Os clientes MQTT os podem enviar mensagens para outros publicando em tópicos assinados por outros clientes.

Um cliente MQTT assina uma publicação, que ele recebe do IBM MQ

Execute a tarefa, [“Publicando uma mensagem para o utilitário do cliente MQTT a partir do IBM MQ Explorer”](#) na página 125 para enviar uma publicação do IBM MQ para um cliente de MQTT.

A maneira padrão de um cliente MQTT v3 receber mensagens é criando uma assinatura para um tópico ou conjunto de tópicos. No fragmento de código de exemplo, [Figura 44 na página 124](#), o cliente MQTT assina usando a sequência de tópicos "MQTT Examples". Um IBM MQ aplicativo C [Figura 45 na página 124](#) publica no tópico usando a sequência de tópicos "MQTT Examples". No fragmento de código [Figura 46 na página 124](#), o cliente MQTT recebe a publicação no método de retorno de chamada, `messageArrived`.

Para obter informações adicionais sobre como configurar o IBM MQ para enviar publicações em resposta a assinaturas de clientes do MQTT, consulte [Publicando uma mensagem em resposta a uma assinatura do cliente do MQTT](#).

Um aplicativo IBM MQ envia uma mensagem diretamente para um cliente MQTT

Execute a tarefa, [“Enviando uma mensagem para um cliente MQTT usando o IBM MQ Explorer”](#) na página 129, para enviar uma mensagem diretamente do IBM MQ para um cliente MQTT.

Uma mensagem enviada dessa forma para um cliente MQTT é chamada de mensagem não solicitada. Os clientes MQTT v3 recebem mensagens não solicitadas como publicações com um nome de tópico configurado. O serviço de telemetria (MQXR) configura o nome do tópico para o nome da fila remota.

Para obter mais informações sobre como configurar o IBM MQ para enviar mensagens diretamente para clientes do MQTT, consulte [Enviando uma mensagem para um cliente diretamente](#).

Um cliente MQTT publica uma mensagem

Um cliente MQTT v3 pode publicar uma mensagem que é recebida por outro cliente de MQTT v3, mas não pode enviar uma mensagem não solicitada. O fragmento de código [Figura 47 na página 125](#) mostra como um cliente MQTT v3, gravado em Java, publica uma mensagem.

O padrão típico para enviar uma mensagem para um cliente específico do MQTT v3, é cada cliente criar uma assinatura para seu próprio `ClientIdentifier`. Execute a tarefa [“Publicando uma mensagem para um cliente MQTT v3”](#) na página 130 para publicar uma mensagem de um cliente MQTT para outro cliente MQTT usando `ClientIdentifier` como uma sequência de tópicos.

Fragmentos de Código de Exemplo

O fragmento de código em [Figura 44 na página 124](#) mostra como um cliente MQTT gravado em Java cria uma assinatura. Ele também precisa de um método de retorno de chamada, `messageArrived`, para receber publicações para a assinatura.

```

String    clientId = String.format("%-23.23s",
                                   System.getProperty("user.name") + "_" +
                                   (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int       QoS = 1;
client.subscribe(topicString, QoS);

```

Figura 44. Assinante do cliente MQTT v3

O fragmento de código em [Figura 45 na página 124](#) mostra como um aplicativo IBM MQ gravado em C envia uma publicação. O fragmento de código é extraído da tarefa [Criar um publicador para um tópico de variável](#)

```

/* Define and set variables to defaults */
/* Omitted lines declaring variables */
char * topicName = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
    MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    td.ObjectType = MQOT_TOPIC; /* Object is a topic */
    td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
    strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
    td.ObjectString.VSPtr = topicString;
    td.ObjectString.VSLength = (MQLONG)strlen(topicString);
    MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    pmo.Options = MQPMO_FAIL_IF QUIESCING | MQPMO_RETAIN;
    MQPUT(Hconn, Hobj, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQDISC(&Hconn, &CompCode, &Reason);
} while (0);

```

Figura 45. Publicador do IBM MQ

Quando a publicação chega, o cliente MQTT chama o método `messageArrived` da classe do MQTT aplicativo cliente `MqttCallback`

```

public class Callback implements MqttCallback {
    public void messageArrived(MqttTopic topic, MqttMessage message) {
        try {
            System.out.println("Message arrived: \" + message.toString()
                               + "\" on topic \" + topic.toString() + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
// ... Other callback methods
}

```

Figura 46. método `messageArrived`

Figura 47 na [página 125](#) mostra um MQTT v3 publicando uma mensagem para a assinatura criada em [Figura 44 na página 124](#).

```

String      address = "localhost";
String      clientId = String.format("%-23.23s",
    System.getProperty("user.name") + " " +
    (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient  client = new MqttClient(address, clientId);
String      topicString = "MQTT Examples";
MqttTopic   topic = client.getTopic(Example.topicString);
String      publication = "Hello world";
MqttMessage message = new MqttMessage(publication.getBytes());
MqttDeliveryToken token = topic.publish(message);

```

Figura 47. Publicador do cliente MQTT v3

Windows Linux AIX **Publicando uma mensagem para o utilitário do cliente MQTT a partir do IBM MQ Explorer**

Siga as etapas nesta tarefa para publicar uma mensagem usando o IBM MQ Explorer e assine-a com o utilitário do cliente MQTT. Uma tarefa adicional mostra como configurar um alias de gerenciador de filas em vez de configurar a fila de transmissão padrão para SYSTEM.MQTT.TRANSMIT.QUEUE.

Antes de começar

A tarefa assume que você esteja familiarizado com o IBM MQ e o IBM MQ Explorer e que o IBM MQ e o recurso MQ Telemetry estejam instalados.

O usuário criando os recursos do gerenciador de filas para esta tarefa deve ter autoridade suficiente para isso. Para fins de demonstração, o ID do usuário IBM MQ Explorer é assumido como um membro do grupo mqm.

Sobre esta tarefa

Na tarefa, você cria um tópico no IBM MQ e subscreve-se para o tópico usando o utilitário do cliente MQTT. Quando você publica para o tópico usando o IBM MQ Explorer, o cliente MQTT recebe a publicação.

Procedimento

Execute uma das seguintes tarefas:

- Você instalou o MQ Telemetry, mas ainda não o iniciou. Execute a tarefa: [“Iniciando Tarefa sem Serviço de Telemetria \(MQXR\) Definido”](#) na página 126.
- Você executou o IBM MQ Telemetry antes, mas quer usar um novo gerenciador de filas para fazer a demonstração. Execute a tarefa: [“Iniciando Tarefa sem Serviço de Telemetria \(MQXR\) Definido”](#) na página 126.
- Você quer executar a tarefa usando um gerenciador de filas existente que não tem recursos de telemetria definidos. Você não quer executar o assistente **Definir Configuração de Amostra**.
 - a. Execute uma das seguintes tarefa para configurar a telemetria:
 - [Configurando um gerenciador de filas para telemetria no Linux e no AIX](#)
 - [Configurando um gerenciador de filas para telemetria no Windows](#)
 - b. Execute a tarefa: [“Iniciando Tarefa com Serviço de Telemetria \(MQXR\) em Execução”](#) na página 127
- Se quiser executar a tarefa usando um gerenciador de filas existente que já tem recursos de telemetria definidos, execute a tarefa: [“Iniciando Tarefa com Serviço de Telemetria \(MQXR\) em Execução”](#) na página 127.

Como proceder a seguir

Execute “Enviando uma mensagem para um cliente MQTT usando o IBM MQ Explorer” na página 129 para enviar uma mensagem diretamente para o utilitário do cliente.

Iniciando Tarefa sem Serviço de Telemetria (MQXR) Definido

Crie um gerenciador de filas e execute **Definir Configuração de Amostra** para definir recursos de telemetria de amostra para o gerenciador de filas. Publique uma mensagem usando o IBM MQ Explorer e assine-a com o utilitário do cliente MQTT.

Sobre esta tarefa

Quando você configura recursos de telemetria de amostra usando **Definir Configuração de Amostra**, o assistente configura as permissões de ID do usuário guest. Considere atentamente se você quer que o ID do usuário guest seja autorizado dessa forma. guest on Windowse nobody on Linuxrecebem permissão para publicar e assinar a raiz da árvore de tópicos e para colocar mensagens em SYSTEM.MQTT.TRANSMIT.QUEUE.

O assistente também configura a fila de transmissão padrão como SYSTEM.MQTT.TRANSMIT.QUEUE, o que pode interferir nos aplicativos em execução em um gerenciador de filas existente. É possível, mas trabalhoso, configurar a telemetria e não usar a fila de transmissão padrão; execute a seguinte tarefa: “Usando um Alias de Gerenciador de Filas” na página 128. Nessa tarefa, você cria um gerenciador de filas para evitar a possibilidade de interferência em qualquer fila de transmissão padrão existente.

Procedimento

1. Usando o IBM MQ Explorer, crie e inicie um novo gerenciador de filas.
 - a) Clique com o botão direito na Queue Managers Pasta do > **Novo** > **Gerenciador de filas...** Digite um nome de gerenciador de filas > **Concluir**.
Crie um nome de gerenciador de filas; por exemplo, MQTTQMR.
2. Crie e inicie o serviço de telemetria (MQXR) e crie um canal de telemetria de amostra.
 - a) Abra a pasta Queue Managers\QmgrName\Telemetry.
 - b) Clique em **Definir configuração de amostra...** > **Concluir**
Deixe a caixa de seleção **Ativar utilitário do cliente MQTT** marcada.
3. Crie uma assinatura para MQTT Example usando o utilitário do cliente MQTT.
 - a) Clique em **Conectar**.
O **Histórico do Cliente** registra um evento do Connected
 - b) Digite MQTT Example no campo **Assinatura\Tópico** > **Assinar**.
O **Histórico do Cliente** registra um evento do Subscribed
4. Crie MQTTExampleTopic no IBM MQ.
 - a) Clique com o botão direito na pasta Queue Managers\QmgrName\Topics no **MQ Explorer** > **Novo** > **Tópico**.
 - b) Digite MQTTExampleTopic como o **Nome** > **Avançar**.
 - c) Digite MQTT Example como a **Sequência de tópicos** > **Concluir**.
 - d) Clique em **OK** para fechar a janela de reconhecimento.
5. Publicar Hello World! no tópico MQTT Example usando IBM MQ Explorer.
 - a) Clique na pasta Queue Managers\QmgrName\Topics no IBM MQ Explorer.
 - b) Clique com o botão direito em MQTTExampleTopic > **Publicação de teste...**
 - c) Digite Hello World! no campo **Dados da mensagem** > **Publicar mensagem** > Alternar para a janela Utilitário do cliente do MQTT.
O **Histórico do Cliente** registra um evento do Received

Iniciando Tarefa com Serviço de Telemetria (MQXR) em Execução

Crie um canal de telemetria e um tópico. Autorize o usuário a usar o tópico e a fila de transmissão de telemetria. Publique uma mensagem usando o IBM MQ Explorer e assine-a com o utilitário do cliente MQTT.

Antes de começar

Nesta versão da tarefa, um gerenciador de filas, *QmgrName*, está definido e em execução. Um serviço de telemetria (MQXR) está definido e em execução. O serviço de telemetria (MQXR) pode ter sido criado manualmente ou pela execução do assistente **Definir Configuração de Amostra**.

Sobre esta tarefa

Nesta tarefa você configura um gerenciador de filas existente para enviar uma publicação para o utilitário do cliente MQTT.

A etapa “1” na página 127 da tarefa configura a fila de transmissão padrão como SYSTEM.MQTT.TRANSMIT.QUEUE, o que pode interferir nos aplicativos em execução em um gerenciador de filas existente. É possível, mas trabalhoso, configurar a telemetria e não usar a fila de transmissão padrão; execute a seguinte tarefa: [“Usando um Alias de Gerenciador de Filas” na página 128](#).

Procedimento

1. Configure SYSTEM.MQTT.TRANSMIT.QUEUE como a fila de transmissão padrão.
 - a) Clique com o botão direito em Queue Managers*QmgrName* folder > **Propriedades...**
 - b) Clique em **Comunicação** no navegador.
 - c) Clique em **Selecionar ...** > Selecionar ... SYSTEM.MQTT.TRANSMIT.QUEUE > **OK** > **OK**
2. Crie um canal de telemetria MQTTEExampleChannel para conectar o utilitário do cliente MQTT ao IBM MQ e iniciar o utilitário do cliente MQTT.
 - a) Clique com o botão direito na pasta Queue Managers*QmgrName* \Telemetry\Channels no **MQ Explorer** > **Novo** > **Canal de telemetria...**
 - b) Digite MQTTEExampleChannel no campo **Nome do canal** > **Avançar** > **Avançar**.
 - c) Mude o **ID do usuário fixo** no painel de autorização do cliente para o ID do usuário que publicará e assinará MQTTEExample > **Próximo**.
 - d) Deixe **Ativar utilitário do cliente** marcado > **Concluir**.
3. Crie uma assinatura para MQTT Example usando o utilitário do cliente MQTT.
 - a) Clique em **Conectar**.

O **Histórico do Cliente** registra um evento do Connected
 - b) Digite MQTT Example no campo **Assinatura\Tópico** > **Assinar**.

O **Histórico do Cliente** registra um evento do Subscribed
4. Crie MQTTEExampleTopic no IBM MQ.
 - a) Clique com o botão direito na pasta Queue Managers*QmgrName*\Topics no **MQ Explorer** > **Novo** > **Tópico**.
 - b) Digite MQTTEExampleTopic como o **Nome** > **Avançar**.
 - c) Digite MQTT Example como a **Sequência de tópicos** > **Concluir**.
 - d) Clique em **OK** para fechar a janela de reconhecimento.
5. Se você deseja que um usuário que não esteja no grupo mqm publique e assine o tópico MQTTEExample, faça o seguinte:
 - a) Autorize o usuário a publicar e assinar o tópico MQTTEExampleTopic:


```
setmqaut -m qMgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
```

b) Autorize o usuário a colocar uma mensagem na SYSTEM.MQTT.TRANSMIT.QUEUE:

```
setmqaut -m qMgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
```

6. Publicar Hello World! no tópico MQTT Example usando IBM MQ Explorer.

a) Clique na pasta Queue Managers\QmgrName\Topics no IBM MQ Explorer.

b) Clique com o botão direito em MQTTExampleTopic > **Publicação de teste...**

c) Digite Hello World! no campo **Dados da mensagem** > **Publicar mensagem** > Alternar para a janela Utilitário do cliente do MQTT.

O **Histórico do Cliente** registra um evento do Received

Usando um Alias de Gerenciador de Filas

Publique uma mensagem no utilitário do cliente MQTT usando IBM MQ Explorer sem configurar a fila de transmissão padrão como SYSTEM.MQTT.TRANSMIT.QUEUE.

A tarefa é uma continuação da tarefa anterior e usa um alias de gerenciador de filas para evitar a configuração da fila de transmissão padrão como SYSTEM.MQTT.TRANSMIT.QUEUE.

Antes de começar

Conclua qualquer uma das tarefas, [“Iniciando Tarefa sem Serviço de Telemetria \(MQXR\) Definido”](#) na página 126 ou [“Iniciando Tarefa com Serviço de Telemetria \(MQXR\) em Execução”](#) na página 127.

Sobre esta tarefa

Quando um cliente MQTT cria uma assinatura, o IBM MQ envia sua resposta usando ClientIdentifier como o nome do gerenciador de filas remotas. Nessa tarefa, ele usa o ClientIdentifier, MyClient.

Se não houver fila de transmissão ou alias do gerenciador de filas chamado MyClient, a resposta será colocada na fila de transmissão padrão. Ao configurar a fila de transmissão padrão para SYSTEM.MQTT.TRANSMIT.QUEUE, o cliente MQTT obtém a resposta

É possível evitar a configuração da fila de transmissão padrão como SYSTEM.MQTT.TRANSMIT.QUEUE usando um alias de gerenciador de filas. Você deve configurar um alias de gerenciador de filas para cada ClientIdentifier. Normalmente há vários clientes para tornar prático o uso de alias de gerenciador de filas. Muitas vezes o ClientIdentifier é imprevisível, tornando impossível a configuração de telemetria dessa maneira.

Contudo, em algumas circunstâncias, você pode ter que configurar a fila de transmissão padrão para algo além de SYSTEM.MQTT.TRANSMIT.QUEUE. As etapas em [Procedimento](#) configuram um alias do gerenciador de filas em vez de configurar a fila de transmissão padrão para SYSTEM.MQTT.TRANSMIT.QUEUE

Procedimento

1. Remova SYSTEM.MQTT.TRANSMIT.QUEUE como a fila de transmissão padrão.

a) Clique com o botão direito em Queue Managers\QmgrName folder > **Propriedades...**

b) Clique em **Comunicação** no navegador.

c) Remova SYSTEM.MQTT.TRANSMIT.QUEUE do campo **Fila de transmissão padrão** > **OK**.

2. Verifique se não é mais possível criar uma assinatura com o utilitário do cliente MQTT:

a) Clique em **Conectar**.

O **Histórico do Cliente** registra um evento do Connected

b) Digite MQTT Example no campo **Assinatura\Tópico** > **Assinar**.

- O **Histórico do Cliente** registra um evento `Subscribe failed` e um evento `Connection lost`
3. Crie um alias de gerenciador de filas para `ClientIdentifier`, `MyClient`.
 - a) Clique com o botão direito na pasta `Queue Managers\QmgrName\Queues` > **Novo** > **Definição de fila remota**.
 - b) Dê à definição o nome `MyClient` > **Avançar**.
 - c) Digite `MyClient` no campo **Gerenciador de Filas Remotas**.
 - d) Digite `SYSTEM.MQTT.TRANSMIT.QUEUE` no campo **Fila de transmissão** > **Concluir**.
 4. Conecte o utilitário do cliente MQTT novamente.
 - a) Verifique se **Identificador de Cliente** está configurado como `MyClient`.
 - b) **Conectar**

O **Histórico do Cliente** registra um evento do `Connected`
 5. Crie uma assinatura para MQTT `Example` usando o utilitário do cliente MQTT.
 - a) Clique em **Conectar**.

O **Histórico do Cliente** registra um evento do `Connected`

 - b) Digite MQTT `Example` no campo **Assinatura\Tópico** > **Assinar**.

O **Histórico do Cliente** registra um evento do `Subscribed`
 6. Publicar `Hello World!` no tópico MQTT `Example` usando IBM MQ Explorer.
 - a) Clique na pasta `Queue Managers\QmgrName\Topics` no IBM MQ Explorer.
 - b) Clique com o botão direito em `MQTTExampleTopic` > **Publicação de teste...**
 - c) Digite `Hello World!` no campo **Dados da mensagem** > **Publicar mensagem** > Alternar para a janela Utilitário do cliente do MQTT.

O **Histórico do Cliente** registra um evento do `Received`

Windows Linux AIX Enviando uma mensagem para um cliente MQTT usando o IBM MQ Explorer

Envie uma mensagem para o utilitário do cliente MQTT colocando uma mensagem em uma fila do IBM MQ usando o IBM MQ Explorer. A tarefa mostra como configurar uma definição de fila remota para enviar uma mensagem diretamente para um cliente MQTT.

Antes de começar

Execute a tarefa [“Publicando uma mensagem para o utilitário do cliente MQTT a partir do IBM MQ Explorer”](#) na página 125. Deixe o utilitário do cliente de MQTT conectado.

Sobre esta tarefa

A tarefa demonstra o envio de uma mensagem para um cliente MQTT usando uma fila em vez de publicá-la para um tópico. Você não cria uma assinatura no cliente. A etapa [“2”](#) na página 130 da tarefa demonstra que a assinatura anterior foi excluída.

Procedimento

1. Descarte quaisquer assinaturas existentes desconectando e reconectando o utilitário do cliente MQTT.

A assinatura é descartada porque, a menos que você altere os padrões, o utilitário do cliente MQTT se conecta a uma sessão limpa; consulte [Sessões limpas](#).

Para tornar mais fácil a execução da tarefa, digite seu próprio `ClientIdentifier`, em vez de usar o `ClientIdentifier` gerado criado pelo utilitário cliente de MQTT.

 - a) Clique em **Desconectar** para desconectar o utilitário do cliente MQTT do canal de telemetria.

O **Histórico do Cliente** registra um evento Disconnected

- b) Altere **Identificador de Cliente** para MyClient.
- c) Clique em **Conectar**.

O **Histórico do Cliente** registra um evento Connected

2. Verifique se o utilitário do cliente do MQTT não recebe mais publicação para o MQTTExampleTopic.
 - a) Clique na pasta Queue Managers\QmgrName\Topics no IBM MQ Explorer.
 - b) Clique com o botão direito em MQTTExampleTopic > **Publicação de teste...**
 - c) Digite Hello World! no campo **Dados da mensagem** > **Publicar mensagem** > Alternar para a janela Utilitário do cliente do MQTT.

Nenhum evento é registrado no **Histórico do Cliente**.

3. Crie uma definição de fila remota para o cliente.

Configure ClientIdentifier, MyClient, como o nome do gerenciador de filas remotas na definição de fila remota. Use qualquer nome como nome da fila remota. O nome da fila remota é passado para um cliente MQTT como o nome do tópico.

- a) Clique com o botão direito na pasta Queue Managers\QmgrName\Queues > **Novo** > **Definição de fila remota**.
 - b) Dê à definição o nome MyClientRemoteQueue > **Avançar**.
 - c) Digite MQTTExampleQueue no campo **Fila Remota**.
 - d) Digite MyClient no campo **Gerenciador de Filas Remotas**.
 - e) Digite SYSTEM.MQTT.TRANSMIT.QUEUE no campo **Fila de transmissão** > **Concluir**.
4. Coloque uma mensagem de teste no MyClientRemoteQueue.
 - a) Clique com o botão direito em **MyClientRemoteQueue** > **Colocar mensagem de teste...**
 - b) Digite Hello queue! no campo de dados Mensagem > **Colocar mensagem** > **Fechar**

O **Histórico do Cliente** registra um evento do Received

5. Remova SYSTEM.MQTT.TRANSMIT.QUEUE como a fila de transmissão padrão.
 - a) Clique com o botão direito em Queue Managers\QmgrName folder > **Propriedades...**
 - b) Clique em **Comunicação** no navegador.
 - c) Remova SYSTEM.MQTT.TRANSMIT.QUEUE do campo **Fila de transmissão padrão** > **OK**.
6. Reexecute a etapa “4” na [página 130](#).

MyClientRemoteQueue é uma definição de fila remota que nomeia explicitamente a fila de transmissão. Não é necessário definir uma fila de transmissão padrão para enviar uma mensagem para o MyClient.

Como proceder a seguir

Com a fila de transmissão padrão não mais configurada como SYSTEM.MQTT.TRANSMIT.QUEUE, o MQTT Client Utility não consegue criar uma nova assinatura, a menos que um alias do gerenciador de filas seja definido para ClientIdentifier, MyClient. Restaure a fila de transmissão padrão para SYSTEM.MQTT.TRANSMIT.QUEUE.

Publicando uma mensagem para um cliente MQTT v3

Publique uma mensagem de um cliente MQTT v3 para outro usando ClientIdentifier como o nome do tópico e o IBM MQ como o broker de publicação/assinatura.

Antes de começar

Execute a tarefa “Publicando uma mensagem para o utilitário do cliente MQTT a partir do IBM MQ Explorer” na página 125. Deixe o utilitário do cliente de MQTT conectado.

Sobre esta tarefa

A tarefa demonstra duas coisas:

1. Assinar um tópico em um cliente MQTT e receber uma publicação de outro cliente MQTT .
2. Configurar assinaturas "ponto a ponto" usando `ClientIdentifier` como a sequência de tópicos.

Procedimento

1. Descarte quaisquer assinaturas existentes desconectando e reconectando o utilitário do cliente MQTT.

A assinatura é descartada porque, a menos que você altere os padrões, o utilitário do cliente MQTT se conecta a uma sessão limpa; consulte [Sessões limpas](#).

Para tornar mais fácil a execução da tarefa, digite seu próprio `ClientIdentifier`, em vez de usar o `ClientIdentifier` gerado criado pelo utilitário cliente de MQTT.

- a) Clique em **Desconectar** para desconectar o utilitário do cliente MQTT do canal de telemetria.

O **Histórico do Cliente** registra um evento `Disconnected`

- b) Altere **Identificador de Cliente** para `MyClient`.

- c) Clique em **Conectar**.

O **Histórico do Cliente** registra um evento `Connected`

2. Crie uma assinatura para o tópico, `MyClient`

`MyClient` é `ClientIdentifier` desse cliente.

- a) Digite `MyClient` no campo **Assinatura\Tópico** > **Assinar**.

O **Histórico do Cliente** registra um evento do `Subscribed`

3. Inicie outro utilitário do cliente de MQTT.

- a) Abra a pasta `Queue Managers\QmgrName\Telemetry\channels`.

- b) Clique com o botão direito no canal **Texto sem formatação** > **Executar utilitário do cliente MQTT...**

- c) Clique em **Conectar**.

O **Histórico do Cliente** registra um evento `Connected`

4. Publique `Hello MyClient!` para o tópico `MyClient`

- a) Copie o tópico da assinatura, `MyClient`, a partir do utilitário do cliente do MQTT em execução com o `ClientIdentifier`, `MyClient`.

- b) Cole `MyClient` no campo **Publicação\Tópico** de cada uma das instâncias do utilitário do cliente do MQTT.

- c) Digite `Hello MyClient!` no campo **Publicação\mensagem**.

- d) Clique em **Publicar** em ambas as instâncias.

Resultados

O **Histórico do cliente** no utilitário do cliente do MQTT com o `ClientIdentifier`, `MyClient`, registra dois eventos em **Recebidos** e um evento em **Publicados**. A outra instância do utilitário do cliente MQTT registra um evento **Publicado**.

Se você vir apenas um evento **Recebido**, verifique as causas possíveis a seguir:

1. A fila de transmissão padrão para o gerenciador de filas está configurada como `SYSTEM.MQTT.TRANSMIT.QUEUE?`

2. Você criou aliases de gerenciador de filas ou de definições de filas remotas referenciando MyClient ao fazer os outros exercícios? Caso você tenha um problema de configuração, exclua quaisquer recursos que referenciam MyClient, como aliases de gerenciador de filas ou filas de transmissão. Desconecte os utilitários do cliente, pare e reinicie o serviço de telemetria (MQXR).

Windows

Linux

AIX

Enviando uma mensagem para um aplicativo

IBM MQ por meio de um cliente do MQTT

Um aplicativo IBM MQ pode receber uma mensagem de um cliente do MQTT v3 assinando um tópico. O cliente do MQTT se conecta ao IBM MQ usando um canal de telemetria e envia uma mensagem para o aplicativo IBM MQ publicando no mesmo tópico.

Execute a tarefa, “[Publicando uma mensagem para IBM MQ a partir de um cliente MQTT](#)” na página 132, para saber como enviar uma publicação de um cliente MQTT para uma assinatura definida no IBM MQ.

Se o tópico for armazenado em cluster ou distribuído usando uma hierarquia de publicação/assinatura, a assinatura poderá estar em um gerenciador de filas diferente para o gerenciador de filas ao qual o cliente MQTT está conectado.

Windows

Linux

AIX

Publicando uma mensagem para IBM MQ a

partir de um cliente MQTT

Crie uma assinatura para um tópico usando o IBM MQ Explorer e publique no tópico usando o utilitário do cliente MQTT.

Antes de começar

Execute a tarefa “[Publicando uma mensagem para o utilitário do cliente MQTT a partir do IBM MQ Explorer](#)” na página 125. Deixe o utilitário do cliente de MQTT conectado.

Sobre esta tarefa

A tarefa demonstra a publicação de uma mensagem com um cliente MQTT e o recebimento da publicação usando uma assinatura durável não gerenciada criada usando o IBM MQ Explorer.

Procedimento

1. Crie uma assinatura durável para a sequência de tópicos MQTT Example.
Execute as etapas a seguir para criar a fila e a assinatura usando o IBM MQ Explorer.
 - a) Clique com o botão direito na pasta Queue Managers*QmgrName*\Queues em IBM MQ Explorer > **Novo > Fila local...**
 - b) Digite MQTTExampleQueue como o nome da fila > **Concluir**.
 - c) Clique com o botão direito na pasta Queue Managers*QmgrName*\Subscriptions em IBM MQ Explorer > **Novo > Assinatura...**
 - d) Digite MQTTExampleSubscription como o nome da fila > **Avançar**.
 - e) Clique em **Selecionar...** > MQTTExampleTopic > **OK**.
Você já criou o tópico, MQTTExampleTopic na etapa “4” na página 126 de “[Publicando uma mensagem para o utilitário do cliente MQTT a partir do IBM MQ Explorer](#)” na página 125.
 - f) Digite MQTTExampleQueue como o nome do destino > **Concluir**.
2. Como uma etapa opcional, configure a fila para uso por um usuário diferente, sem autoridade do mqm.
Se você estiver definindo a configuração para usuários com menos autoridade do que mqm, será necessário fornecer ao put e get autoridade para o MQTTExampleQueue. O acesso ao tópico e à fila de transmissão foi configurado em “[Publicando uma mensagem para o utilitário do cliente MQTT a partir do IBM MQ Explorer](#)” na página 125.
 - a) Autorize um usuário a colocar e chegar à fila MQTTExampleQueue:

```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```

3. Publique Hello IBM MQ! no tópico MQTT Example usando o utilitário cliente MQTT .

Se você não deixou o utilitário do cliente do MQTT conectado, clique com o botão direito no canal **PlainText** > **Executar o utilitário do cliente do MQTT...** > **Conectar**.

a) Digite MQTT Example no campo **Publicação\Tópico**.

b) Digite Hello IBM MQ! no campo **Publicação\Mensagem** > **Publicar**.

4. Abra a pasta Queue Managers\QmgrName\Queues e localize MQTTExampleQueue.

O campo **Profundidade da fila atual** é 1

5. Clique com o botão direito em MQTTExampleQueue > **Procurar mensagens ...** e examine a publicação.

Windows

Linux

AIX

Aplicativos de publicação/assinatura do MQTT

Use publicação/assinatura baseada em tópico para escrever aplicativos MQTT .

Quando o cliente MQTT está conectado, publicações fluem em qualquer direção entre o cliente e o servidor. As publicações são enviadas do cliente quando informações são publicadas no cliente. As publicações são recebidas no cliente quando uma mensagem é publicada para um tópico que corresponde a uma assinatura criada pelo cliente.

O broker de publicação/assinatura do IBM MQ gerencia os tópicos e as assinaturas criados pelos clientes do MQTT. Os tópicos criados pelos clientes MQTT compartilham o mesmo espaço de tópico que os tópicos criados pelos aplicativos do IBM MQ.

Publicações que correspondem à sequência de tópicos em uma assinatura do cliente MQTT são colocadas em SYSTEM.MQTT.TRANSMIT.QUEUE com o nome do gerenciador de filas remotas configurado para o ClientIdentifier do cliente. O serviço de telemetria (MQXR) encaminha as publicações para o cliente que criou a assinatura. Ele usa ClientIdentifier, que foi configurado como o nome do gerenciador de filas remotas para identificar o cliente.

Normalmente, SYSTEM.MQTT.TRANSMIT.QUEUE deve ser definida como a fila de transmissão padrão. É possível, mas oneroso, configurar o MQTT para não usar a fila de transmissão padrão; consulte [Configurar enfileiramento distribuído para enviar mensagens para clientes MQTT](#).

Um cliente MQTT pode criar uma sessão persistente; consulte [“Sessões stateless e stateful do MQTT” na página 136](#). Assinaturas criadas em uma sessão persistente são duráveis. Publicações que chegam para um cliente com uma sessão persistente são armazenadas na SYSTEM.MQTT.TRANSMIT.QUEUE e encaminhadas para o cliente quando ele se reconecta.

Um cliente MQTT também pode publicar e assinar as publicações retidas; consulte [Publicações retidas e clientes MQTT](#). Um assinante para um tópico de publicação retida recebe a publicação mais recente para o tópico. O assinante recebe a publicação retida quando cria uma assinatura ou quando se reconecta à sua sessão anterior.

Windows

Linux

AIX

Aplicativos de Telemetria

Gravar os aplicativos de telemetria usando fluxos de mensagens do IBM MQ ou do IBM Integration Bus.

Use o JMS, MQI ou outras interfaces de programação do IBM MQ para programar aplicativos de telemetria no IBM MQ.

O serviço de telemetria (MQXR) converte entre mensagens do MQTT v3 e mensagens do IBM MQ. Ele cria assinaturas e publicações em nome de clientes MQTT e encaminha as publicações para clientes do MQTT. Uma publicação é a carga útil de uma mensagem do MQTT v3. A carga útil abrange cabeçalhos de mensagem e uma matriz de bytes em formato `jms-bytes`. O servidor de telemetria mapeia os cabeçalhos entre uma mensagem do MQTT v3 e uma mensagem do IBM MQ; consulte [“Integração do MQ Telemetry com gerenciadores de filas” na página 134](#).

Utilize os nós Publication, MQInput e JMSInput para enviar e receber publicações entre os clientes IBM Integration Bus e MQTT.

Usando fluxos de mensagens é possível integrar a telemetria com websites usando HTTP, e com outros aplicativos usando IBM MQ e WebSphere Adapters.

Windows Linux AIX **Integração do MQ Telemetry com gerenciadores de filas**

O cliente MQTT está integrado com o IBM MQ como um aplicativo de publicação/assinatura. Ele pode publicar ou se inscrever para tópicos no IBM MQ, criando novos tópicos ou usando tópicos existentes. Ele recebe publicações do IBM MQ como um resultado dos clientes MQTT, incluindo a si mesmo, ou outros aplicativos de publicação do IBM MQ nos tópicos de suas assinaturas. Regras são aplicadas para decidir os atributos de uma publicação.

Muitos dos atributos associados a tópicos, publicações, assinaturas e mensagens que são fornecidos pelo IBM MQ, não são suportados. “[Cliente MQTT para broker de publicação/assinatura do IBM MQ](#)” na página 134 e “[IBM MQ para um cliente MQTT](#)” na página 135 descrevem como atributos de publicações são configurados. As configurações dependem de se a publicação está indo para um broker de publicação/assinatura do IBM MQ, ou a partir dele.

No IBM MQ, os tópicos de publicação/assinatura são associados aos objetos do tópico administrativo. Os tópicos criados pelos clientes MQTT não são diferentes. Quando um cliente MQTT cria uma sequência de tópicos para uma publicação no IBM MQ, o broker de publicação/assinatura o associa a um objeto do tópico administrativo. O broker mapeia a sequência de tópicos na publicação para o pai do objeto do tópico administrativo mais próximo. O mapeamento é o mesmo para aplicativos IBM MQ. Se não houver um tópico criado por usuário, o tópico da publicação será mapeado para SYSTEM.BASE.TOPIC. Os atributos que são aplicados à publicação são derivados do objeto do tópico.

Quando um aplicativo IBM MQ ou um administrador cria uma assinatura, a assinatura é nomeada. Listar assinaturas usando o IBM MQ Explorer ou usando comandos `runmqsc` ou PCF Todas as assinaturas do cliente MQTT são nomeadas. Eles recebem um nome do formato: *ClientIdentifier:Topic name*

Cliente MQTT para broker de publicação/assinatura do IBM MQ

Um cliente MQTT enviou uma publicação para o IBM MQ. O serviço de telemetria (MQXR) converte a publicação em uma mensagem do IBM MQ. A mensagem do IBM MQ contém três partes:

1. [MQMD](#)
2. [RFH2](#)
3. Mensagem

As propriedades do MQMD são configuradas para seus valores padrão, exceto onde estiver anotado na Tabela 9 na página 134.

Campo MQMD	tipo	Value
Format	MQCHAR8	MQFMT_RF_HEADER_2
UserIdentifier	MQCHAR12	Configure para uma das opções a seguir: MqttClient.ClientIdentifier MqttConnectOptions.UserName Um ID do usuário configurado pelo administrador do IBM MQ para o canal de telemetria.
Priority	MQLONG	MQPRI_PRIORITY_AS_Q_DEF (Padrão para o IBM MQ, que é diferente para JMS cujo padrão é 4.)

<i>Tabela 9. Configurações para propriedades do MQMD (continuação)</i>		
Campo MQMD	tipo	Value
Persistence	MQLONG	QoS=0→MQPER_NOT_PERSISTENT QoS=1→MQPER_PERSISTENT QoS=2→MQPER_PERSISTENT

O cabeçalho RFH2 não contém uma pasta <msd> para definir o tipo da mensagem do JMS. O serviço de telemetria (MQXR) cria a mensagem do IBM MQ como uma mensagem padrão do JMS. O tipo de mensagem padrão do JMS é uma mensagem `jms-bytes`. Um aplicativo pode acessar informações do cabeçalho adicionais como propriedades de mensagem; consulte [Propriedades de Mensagem](#).

Os valores de RFH2 são configurados conforme mostrado na [Tabela 10 na página 135](#). A propriedade `Format` é configurada no cabeçalho fixo RFH2 e os outros valores são configurados nas pastas do RFH2.

<i>Tabela 10. Configurações para propriedades RFH2</i>		
Propriedade RFH2	Tipo/Pasta	Cabeçalho
<code>Format</code>	MQCHAR8	MQFMT_NONE
<code>ClientIdentifier</code>	mqtt/clientId	Cópia de <code>MqttClient.ClientIdentifier</code> com um comprimento de 1...23 bytes.
<code>QoS</code>	mqtt/qos	Cópia de QoS da mensagem recebida do MQTT.
<code>ID de mensagem</code>	mqtt/msgid	Cópia de ID de mensagem da mensagem recebida do MQTT, se QoS for 1 ou 2.
<code>MQIsRetained</code>	mmps/Ret	Configurado se a publicação original do MQTT foi enviada com a propriedade <code>RETAIN</code> configurada e a mensagem for recebida como uma publicação retida.
<code>MQTopicString</code>	mmps/Top	O tópico para o qual a mensagem do MQTT foi publicada.

A carga útil em uma publicação do MQTT é mapeada para o conteúdo de uma mensagem do IBM MQ:

<i>Tabela 11. Como a carga útil em uma publicação do MQTT é mapeada para os conteúdos da mensagem do IBM MQ</i>		
Conteúdo da Mensagem	tipo	Conteúdos da mensagem do IBM MQ
<code>Buffer</code>	MQBYTE <i>n</i>	Cópia de bytes da mensagem recebida do MQTT. O comprimento pode ser zero.

IBM MQ para um cliente MQTT

Um cliente foi inscrito para o tópico de uma publicação. Um aplicativo IBM MQ publicou no tópico, resultando em uma publicação sendo enviada para o assinante de MQTT pelo broker de publicação/assinatura do IBM MQ. Como alternativa, um aplicativo IBM MQ enviou uma mensagem não solicitada diretamente para um cliente MQTT. [Tabela 12 na página 136](#) descreve como cabeçalhos de mensagem fixos são configurados na mensagem que é enviada para o cliente MQTT. Quaisquer outros dados no cabeçalho de mensagem do IBM MQ ou quaisquer outros cabeçalhos são descartados. Os dados da mensagem na mensagem do IBM MQ são enviados como carga útil da mensagem na mensagem do MQTT, sem nenhuma mudança. A mensagem do MQTT é enviada para o cliente MQTT pelo serviço de telemetria (MQXR).

Tabela 12. Como cabeçalhos de mensagem fixos são configurados em uma mensagem do IBM MQ enviada para o cliente MQTT

campo MQTT	tipo	Value
DUP	Booleano	Configure se QoS = 1 ou 2 e se a mensagem foi enviada para esse cliente em uma transmissão anterior e não tiver sido reconhecida após um período de tempo.
QoS	int	<p>A maneira como o valor de QoS em uma publicação de saída do broker de publicação/assinatura no IBM MQ é configurado depende da publicação recebida. Ela depende de se a publicação recebida foi enviada a partir de um cliente MQTT ou a partir de um aplicativo IBM MQ.</p> <p>MQTT Diminua o valor de QoS na publicação recebida e da QoS solicitada pelo assinante.</p> <p>IBM MQ Diminua o valor da QoS derivada da publicação recebida:</p> <p style="padding-left: 40px;">MQPER_NOT_PERSISTENT→QoS=0 MQPER_PERSISTENT→QoS=2</p> <p>e QoS solicitada pelo assinante. Se a mensagem for enviada para o cliente sem uma assinatura, a QoS será configurada como 2 por padrão. Um cliente pode alterar esse valor inscrevendo-se para DEFAULT . QoS com uma QoS diferente.</p>
RETAIN	Booleano	Configure se a publicação recebida tiver a propriedade retida configurada.

Tabela 13 na página 136 descreve como cabeçalhos de mensagem variáveis são configurados na mensagem do MQTT que é enviada para o cliente MQTT.

Tabela 13. Como as propriedades do cabeçalho de variável do MQTT são configuradas em uma mensagem do MQTT enviada para o cliente do MQTT

campo MQTT	tipo	Value
Topic name	Sequência	A sequência de tópicos com a qual a mensagem foi publicada.
Message ID	Sequência	Os últimos 2 bytes da propriedade MQMD . MsgId da publicação quando ela é colocada na SYSTEM . MQTT . TRANSMIT . QUEUE.
Payload	byte []	Cópia direta dos bytes da publicação recebida para o broker de publicação/assinatura. O comprimento pode ser zero.

Windows

Linux

AIX

Sessões stateless e stateful do MQTT

Os clientes MQTT podem criar uma sessão stateful com o gerenciador de filas. Quando um cliente MQTT stateful se desconecta, o gerenciador de filas mantém as assinaturas criadas pelo cliente e as mensagens em andamento. Quando o cliente se reconecta, ele resolve a mensagem em andamento. Ele envia quaisquer mensagens enfileiradas para entrega e recebe quaisquer mensagens publicadas para suas assinaturas enquanto ele estava desconectado.

Quando um cliente MQTT se conecta a um canal de telemetria ele inicia uma nova sessão ou retoma uma sessão antiga. Uma sessão nova não tem mensagens pendentes que não foram reconhecidas, nem

assinaturas e nem publicações aguardando entrega. Quando um cliente se conecta, ele especifica se vai começar com uma sessão limpa ou se vai retomar uma sessão existente; consulte [Sessões limpas](#).

Se o cliente retomar uma sessão existente, ele continuará como se a conexão não tivesse sido interrompida. Publicações aguardando entrega são enviadas para o cliente e todas as transferências de mensagens não confirmadas são concluídas. Quando um cliente em uma sessão persistente se desconecta do serviço de telemetria (MQXR), todas as assinaturas criadas por ele permanecem. Publicações para as assinaturas são enviadas para o cliente quando ele se reconecta. Se ele se reconectar sem retomar a sessão antiga, as publicações serão descartadas pelo serviço de telemetria (MQXR).

Informações do estado da sessão são salvas pelo gerenciador de filas na fila `SYSTEM.MQTT.PERSISTENT.STATE`.

O administrador do IBM MQ pode se desconectar e limpar uma sessão.

Windows

Linux

AIX

Quando um MQTT do cliente não está conectado

Quando um cliente não está conectado, o gerenciador de filas pode continuar recebendo publicações em seu nome. Elas são encaminhadas para o cliente quando ele se reconecta. Um cliente pode criar um "última informação e testamento", que o gerenciador de filas publica em nome do cliente caso o cliente se desconecte inesperadamente.

Se você quiser ser notificado quando um cliente se desconectar inesperadamente, é possível registrar um última publicação de testamento; consulte [Última publicação de testamento](#). Ela será enviada pelo serviço de telemetria (MQXR) se detectar que a conexão com o cliente foi interrompida sem a solicitação do cliente.

Um cliente pode publicar uma publicação retida a qualquer momento; consulte [Publicações retidas e clientes MQTT](#). Uma nova assinatura para um tópico pode solicitar que seja enviada para qualquer publicação retida associada ao tópico. Se você criar a última informação e testamento como uma publicação retida, será possível usá-la para monitorar o status de um cliente.

Por exemplo, o cliente publica uma publicação retida quando se conecta informando sua disponibilidade. Ao mesmo tempo, ele cria uma publicação última informação e testamento retida que anuncia sua indisponibilidade. Além disso, antes de fazer uma desconexão planejada, ele publica sua indisponibilidade como uma publicação retida. Para saber se o cliente está disponível, você teria que se inscrever para o tópico da publicação retida. Você sempre receberia uma das três publicações.

Se o cliente tiver que receber mensagens publicadas quando estiver desconectado, reconecte o cliente à sessão anterior; consulte ["Sessões stateless e stateful do MQTT"](#) na página 136. Suas assinaturas ficarão ativas até serem excluídas ou até o cliente criar uma sessão limpa.

Windows

Linux

AIX

Loose coupling entre clientes MQTT e aplicativos

IBM MQ

O fluxo de publicações entre clientes MQTT e aplicativos IBM MQ são fracamente acoplados. Publicações podem se originar de um cliente MQTT ou aplicativo IBM MQ, sem uma ordem configurada. Os publicadores e assinantes são fracamente acoplados. Eles interagem uns com os outros indiretamente por meio de publicações e assinaturas. Você também pode enviar mensagens diretamente para um cliente MQTT a partir de um aplicativo IBM MQ.

Os clientes MQTT e aplicativos IBM MQ são fracamente acoplados em dois sentidos:

1. Publicadores e assinantes são fracamente acoplados pela associação de uma publicação e de uma assinatura com um tópico. Publicadores e assinantes normalmente não têm conhecimento do endereço ou da identidade da outra origem de uma publicação ou assinatura.
2. Clientes MQTT publicam, se inscrevem, recebem publicações e processam confirmações de entrega em encadeamentos separados.

Um aplicativos cliente MQTT não espera até uma publicação ser entregue. O aplicativo passa uma mensagem para o cliente MQTT e depois o aplicativo continua em seu próprio encadeamento. Um token

de entrega é usado para sincronizar o aplicativo com a entrega de uma publicação; consulte [Tokens de entrega](#).

Após passar uma mensagem para o cliente MQTT, o aplicativo tem a opção de esperar o token de entrega. Em vez de esperar, o cliente pode fornecer um método de retorno de chamada que é chamado quando a publicação é entregue para o IBM MQ. Ele também pode ignorar o token de entrega.

Dependendo da qualidade de serviço associada à mensagem, o token de entrega é retornado imediatamente para o método de retorno de chamada ou provavelmente após um tempo considerável. O token de entrega pode até ser retornado após o cliente se desconectar e se reconectar. Se a qualidade de serviço for *fire-forget*, o token de entrega será retornado imediatamente. Nos outros dois casos, o token de entrega será retornado apenas quando o cliente receber reconhecimento de que a publicação foi enviada para os assinantes.

Publicações enviadas para um cliente MQTT como resultado de uma assinatura do cliente são entregues para o método de retorno de chamada `messageArrived`. `messageArrived` é executado em um encadeamento diferente para o aplicativo principal.

Enviando mensagens diretamente para um cliente MQTT

É possível enviar uma mensagem para um determinado cliente MQTT de uma de duas maneiras.

1. Um aplicativo IBM MQ pode enviar uma mensagem diretamente para um cliente MQTT sem uma assinatura; consulte [Enviando uma mensagem para um cliente diretamente](#).
2. Uma abordagem alternativa é usar a convenção de nomenclatura `ClientIdentifier`. Faça todos os assinantes do MQTT criarem assinaturas usando seus `ClientIdentifier` exclusivos como um tópico. Publique para o `ClientIdentifier`. A publicação é enviada para o cliente inscrito para o tópico `ClientIdentifier`. Usando essa técnica, é possível enviar uma publicação para um determinado assinante do MQTT.

Windows

Linux

AIX

MQ Telemetrysegurança

A proteção de dispositivos de telemetria pode ser importante, já que os dispositivos têm a probabilidade de serem portáteis e usados em locais que não podem ser adequadamente controlados. É possível usar VPN para proteger a conexão do dispositivo MQTT para o serviço de telemetria (MQXR). O MQ Telemetry fornece dois outros mecanismos de segurança, TLS e JAAS.

O TLS é usado principalmente para criptografar comunicações entre o dispositivo e o canal de telemetria e para autenticar se o dispositivo está se conectando ao servidor correto; veja [Autenticação de canal de telemetria usando TLS](#). Também é possível usar o TLS para verificar se o dispositivo do cliente tem permissão para se conectar ao servidor; consulte [Autenticação do cliente MQTT usando TLS](#).

O JAAS é usado principalmente para verificar se o usuário do dispositivo tem permissão para usar um aplicativo do servidor; consulte [Autenticação do cliente MQTT usando uma senha](#). O JAAS pode ser usado com LDAP para verificar uma senha usando um diretório de conexão única.

O TLS e o JAAS podem ser usados juntos para fornecer autenticação de dois fatores. É possível restringir as cifras usadas por TLS às cifras que atendem às normas do FIPS.

Com pelo menos dezenas de milhares de usuários, nem sempre é prático fornecer perfis de segurança individuais. E também nem sempre é prático usar os perfis para autorizar usuários individuais a acessarem objetos do IBM MQ. Em vez disso, agrupe usuários em classes para autorizar publicação e assinatura para tópicos e envio de publicações para clientes.

Configure cada canal de telemetria para mapear clientes para IDs de usuário cliente comum. Use um ID de usuário comum para cada cliente que se conecta em um canal específico; consulte [Identidade e autorização do cliente MQTT](#).

A autorização de grupos de usuários não compromete a autenticação de cada indivíduo. Cada usuário individual pode ser autenticado, no cliente ou no servidor, com Nome de Usuário e Senha, e depois autorizado no servidor usando um ID de usuário comum.

A carga útil da mensagem no protocolo MQTT v3 é codificada como matriz de bytes. Geralmente, o texto de manipulação de aplicativos cria a carga útil da mensagem em UTF-8. O canal de telemetria descreve a carga útil da mensagem como UTF-8, mas não faz nenhuma conversão de página de códigos. A sequência de tópicos de publicação deve ser UTF-8.

O aplicativo é responsável por converter dados alfabéticos em dados numéricos e de página de códigos corretos para a codificação de números correta.

O cliente MQTT Java tem um método `MqttMessage.toString` conveniente. O método trata a carga útil da mensagem como sendo codificada no conjunto de caracteres padrão da plataforma local, que geralmente é UTF-8. Ele converte a carga útil para uma sequência Java. Java tem um método de sequência, `getBytes`, que converte uma sequência em uma matriz de bytes codificada usando o conjunto de caracteres padrão da plataforma local. Dois programas MQTT Java trocando texto na carga útil da mensagem entre plataformas com o mesmo conjunto de caracteres padrão fazem isso de maneira fácil e eficiente em UTF-8.

Se o conjunto de caracteres padrão de uma das plataformas não for UTF-8, os aplicativos deverão estabelecer uma convenção para a troca de mensagens. Por exemplo, um publicador especifica conversão de uma sequência em UTF-8 usando o método `getBytes("UTF8")`. Para receber o texto de uma mensagem, o assinante assume que a mensagem esteja codificada no conjunto de caracteres UTF-8.

O serviço de telemetria (MQXR) descreve a codificação de todas as publicações recebidas de mensagens de clientes MQTT como sendo UTF-8. Ele configura o `MQMD.CodedCharSetId` para UTF-8 e `RFH2.CodedCharSetId` para `MQCCSI_INHERIT`; Consulte [“Integração do MQ Telemetry com gerenciadores de filas”](#) na página 134. O formato da publicação é configurado como `MQFMT_NONE`, portanto, nenhuma conversão pode ser executada por canais ou por `MQGET`.

Considere os seguintes fatores ao gerenciar grandes números de clientes e melhorar a escalabilidade do MQ Telemetry.

Planejamento de Capacidade

Para obter informações sobre relatórios de desempenho para o MQ Telemetry, consulte [Documentos de desempenho do MQ](#).

Conexões

Os custos envolvidos em conexões incluem

- O custo da configuração de uma conexão em si em termos de tempo e uso de processador.
- Custos de rede.
- Memória usada para manter uma conexão aberta, mas sem ser usada.

Há uma carga extra incorrida quando clientes ficam conectados. Se uma conexão é mantida aberta, os fluxos do TCP/IP e mensagens do MQTT usam a rede para verificar se a conexão ainda está lá. Além disso, a memória é usada no servidor para cada conexão do cliente que é mantida aberta.

Se você estiver enviando mais de uma mensagem por minuto, mantenha sua conexão aberta para evitar o custo de iniciar uma nova conexão. Se você estiver enviando menos de uma mensagem a cada 10 - 15 minutos, considere descartar a conexão para evitar o custo de mantê-la aberta. Você talvez deseje manter uma conexão TLS aberta, mas inativa, por períodos mais longos porque é mais caro para configurar.

Além disso, considere o recurso do cliente. Se houver um recurso de armazenamento e encaminhamento no cliente, você pode armazenar mensagens em um lote e descartar a conexão entre o envio dos lotes. No

entanto, se o cliente estiver desconectado, ele não poderá receber uma mensagem do servidor. Portanto, o propósito de seu aplicativo tem influência na decisão.

Se seu sistema tiver um cliente enviando várias mensagens, por exemplo, transferências de arquivos, não espere uma resposta do servidor por mensagem. Em vez disso, envie todas as mensagens e verifique no final se todas elas foram recebidas. Alternativamente, use [Qualidade de Serviço \(QoS\)](#).

É possível variar o QoS por mensagem, entregando mensagens sem importância usando o QoS 0 e mensagens importantes usando um QoS de 2. O rendimento da mensagem pode ser aproximadamente duas vezes maior com um QoS de 0 do que com um QoS de 2.

Convenções de Nomenclatura

Se você estiver projetando seu aplicativo para muitos clientes, implemente uma convenção de nomenclatura efetiva. Para mapear cada cliente para o `ClientIdentifier` correto, faça com que o `ClientIdentifier` seja significativo. Uma boa convenção de nomenclatura facilita para o Administrador a descoberta de quais clientes estão em execução. Uma convenção de nomenclatura ajuda o administrador a filtrar uma longa lista de clientes no IBM MQ Explorer e ajuda na determinação de problemas; consulte [Identificador de cliente](#).

Rendimento

O comprimento dos nomes de tópicos afeta o número de bytes que fluem na rede. Durante uma publicação ou assinatura, o número de bytes em uma mensagem pode ser importante. Portanto, limite o número de caracteres no nome de um tópico. Quando um cliente MQTT se inscreve para um tópico do IBM MQ dá a ele um nome no formato:

```
ClientIdentifier: TopicName
```

Para visualizar todas as assinaturas para um cliente de MQTT, é possível usar o comando IBM MQ MQSC **DISPLAY**:

```
DISPLAY SUB(' ClientID1:*')
```

Definindo recursos no IBM MQ para ser utilizado por clientes do MQTT

Um cliente do MQTT se conecta a um gerenciador de filas remotas do IBM MQ. Há dois métodos básicos para um aplicativo IBM MQ enviar mensagens para um cliente MQTT : configurar a fila de transmissão padrão para `SYSTEM.MQTT.TRANSMIT.QUEUE` ou usar aliases do gerenciador de filas. Defina a fila de transmissão padrão de um gerenciador de filas, caso haja um grande número de clientes do MQTT. O uso da configuração de fila de transmissão padrão simplifica o esforço de administração; consulte [Configurar o enfileiramento distribuído para enviar mensagens para clientes MQTT](#).

Melhorando a Escalabilidade Evitando Assinaturas

Quando um cliente do MQTT V3 assina um tópico, uma assinatura é criada pelo serviço de telemetria (MQXR) no IBM MQ. A assinatura roteia publicações para o cliente no `SYSTEM.MQTT.TRANSMIT.QUEUE`. O nome do gerenciador de filas remotas no cabeçalho de transmissão de cada publicação é configurado para `ClientIdentifier` do cliente do MQTT que fez a assinatura. Se houver vários clientes, cada um fazendo suas próprias assinaturas, isso resultará em várias assinaturas de proxy que estão sendo mantidas por todo o cluster ou hierarquia de publicação/assinatura do IBM MQ. Para obter informações sobre não usar publicação/assinatura, mas usar uma solução baseada em ponto a ponto, consulte [Enviando uma mensagem para um cliente diretamente](#).

Gerenciando Grandes Números de Clientes

Para suportar diversos clientes conectados simultaneamente, aumente a memória disponível para o serviço de telemetria (MQXR) configurando os parâmetros de JVM **-Xms** e **-Xmx**. Siga estas etapas:

1. Encontre o arquivo `java.properties` no diretório de configuração do serviço de telemetria; consulte [Diretório de configuração de serviço de Telemetria \(MQXR\) no Windows](#) ou [Diretório de configuração do serviço de Telemetria no Linux](#).
2. Siga as direções no arquivo; um heap de 1 GB é suficiente para 50.000 clientes conectados simultaneamente.

```
# Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. Inclua outros argumentos da linha de comandos a serem transmitidos para a JVM executando o serviço de telemetria (MQXR) no arquivo `java.properties`; consulte [Transmitindo parâmetros da JVM para o serviço de telemetria \(MQXR\)](#).

Para aumentar o número de descritores de arquivos abertos no Linux, inclua as linhas a seguir no `/etc/security/limits.conf/` e efetue login novamente.

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

Cada soquete requer um descritor de arquivo. O serviço de telemetria requer alguns descritores de arquivo adicionais, portanto, esse número deve ser maior que o número de soquetes abertos necessário.

O gerenciador de filas usa manipulação de objetos para cada assinatura não durável. Para suportar diversas assinaturas não duráveis ativas, aumente o número máximo de manipulações ativas no gerenciador de filas; por exemplo:

```
echo ALTER QMGR MAXHANDS(99999999) | runmqsc qMgrName
```

Figura 48. Alterar número máximo de manipulações no Windows

```
echo "ALTER QMGR MAXHANDS(99999999)" | runmqsc qMgrName
```

Figura 49. Alterar número máximo de manipulações no Linux

Outras contraprestações

Ao planejar os requisitos do sistema, considere a quantidade de tempo levada para reiniciar o sistema. O tempo de inatividade planejado pode ter implicações quanto ao número de mensagens que estão enfileiradas aguardando para serem processadas. Configure o sistema para que as mensagens possam ser processadas com sucesso em um tempo aceitável. Revise o armazenamento em disco, a memória e a energia de processamento. Com alguns aplicativos clientes, talvez seja possível descartar mensagens quando o cliente se reconectar. Para descartar mensagens, configure `CleanSession` nos parâmetros de conexão do cliente; consulte [Sessões limpas](#). Alternativamente, publique e assine usando o melhor esforço de Qualidade de serviço 0 em um cliente do MQTT; consulte [Qualidade de serviço](#). Use mensagens não persistentes ao enviar mensagens a partir do IBM MQ. Mensagens com essas qualidades de serviço não são recuperadas quando o sistema ou a conexão são reiniciados.

Windows

Linux

AIX

Dispositivos suportados pelo MQ Telemetry

Os clientes do MQTT podem ser executados em vários dispositivos, de sensores e atuadores, para identificar sistemas de veículos e dispositivos em espera.

Os clientes do MQTT são pequenos e executados em dispositivos restritos por pouca memória e baixa energia de processamento. O MQTT protocol é confiável e tem cabeçalhos pequenos, o que é adequado para redes restritas por baixa largura de banda, alto custo e disponibilidade intermitente.

O MQ Telemetry se comunica com dispositivos de telemetria por meio de aplicativos clientes do MQTT. Esses aplicativos usam os seguintes recursos, todos os quais implementam o protocolo MQTT v3:

- As seguintes bibliotecas de clientes:
 - O *MQTT client for Java*, que é usado para construir aplicativos nativos para (por exemplo) dispositivos Android, OS X, Linux ou Windows. Aplicativos que usam esta biblioteca de cliente podem ser executados em todas as variações de Java desde o menor CLDC (Connected Limited Device Configuration)/MIDP (Mobile Information Device Profile) até o CDC (Connected Device Configuration) /Foundation, J2SE (Java Platform, Standard Edition) e J2EE (Java Platform, Enterprise Edition). A biblioteca de classes customizada do IBM também é suportada. A plataforma Java ME geralmente é usada em dispositivos pequenos, como atuadores, sensores, telefones celulares e outros dispositivos integrados. A plataforma Java SE geralmente é instalada em dispositivos integrados, como computadores desktop e servidores.
 - O *MQTT client for C*, que é usado para construção de aplicativos nativos para dispositivos iOS, OS X, Linux ou Windows (por exemplo). Esta biblioteca cliente fornece uma implementação de referência C juntamente com o cliente nativo pré-construído para os sistemas Windows e Linux. A implementação de referência C permite que o MQTT seja portado para uma ampla faixa de dispositivos e plataformas. Alguns sistemas Windows no Intel, incluindo Windows 7, RedHat, Ubuntu, e alguns sistemas Linux em plataformas ARM, como Eurotech Viper, implementam versões do Linux que executam o cliente C, mas a IBM não fornece suporte de serviço para as plataformas. Você deverá reproduzir os problemas com o cliente em uma plataforma suportada caso pretenda chamar seu centro de suporte IBM.
 - O *MQTT client for Java*, que é usado para construir aplicativos da web baseados em navegador.

As bibliotecas do cliente do MQTT estão disponíveis livremente por meio do Eclipse Paho e do MQTT.org. Consulte [Programas de amostra do IBM MQ Telemetry Transport](#).

Segurança no IBM MQ

No IBM MQ, há vários métodos de fornecimento de segurança: a interface de serviço de autorização; saídas do canal gravadas pelo usuário ou terceiros; segurança do canal usando Segurança da Camada de Transporte (TLS), registros de autenticação de canal e segurança de mensagem.

Interface de serviço de autorização

A autorização para usar chamadas MQI, comandos e acesso aos objetos para objetos é fornecida pelo **Object Authority Manager** (OAM), que está ativado, por padrão. O acesso às entidades IBM MQ é controlado por meio de grupos de usuários do IBM MQ e do OAM. Os administradores podem usar uma interface de linha de comandos para conceder ou revogar as autorizações, conforme necessário.

Para obter mais informações sobre a criação de componentes de serviço de autorização, consulte [Configurando segurança em sistemas AIX, Linux, and Windows](#).

Saídas de Canal Gravadas pelo Usuário ou de Terceiros

Os canais podem usar saídas de canal gravadas pelo usuário ou de terceiros. Para obter informações adicionais, consulte [Programas de saída do canal para canais de sistemas de mensagens](#).

Segurança de canal usando TLS

O protocolo Segurança da Camada de Transporte (TLS) fornece segurança de canal de padrão de mercado, com proteção contra espionagem do tráfego de rede, violação e personificação.

O TLS usa técnicas simétricas e de chave pública para fornecer confidencialidade de mensagem e integridade e autenticação mútua.

Para uma revisão abrangente de segurança no IBM MQ incluindo informações detalhadas sobre o TLS, veja [Segurança](#). Para obter uma visão geral de TLS, incluindo ponteiros para os comandos descritos nesta seção, veja [Protocolos de segurança criptográfica: TLS](#).

Registros de Autenticação de Canal

Use os registros de autenticação de canal para exercer controle preciso sobre o acesso concedido para conectar-se aos sistemas em um nível de canal. Para obter mais informações, consulte [Registros de Autenticação de Canal](#).

Segurança de Mensagem

Use o Advanced Message Security, que é um componente instalado e licenciado separadamente do IBM MQ, para fornecer proteção criptográfica às mensagens enviadas e recebidas usando o IBM MQ. Consulte o [Advanced Message Security](#).

Tarefas relacionadas

[Assegurando](#)

[Planejando para seus requisitos de segurança](#)

Suporte de TLS do cliente gerenciado pelo IBM MQ.NET

O cliente totalmente gerenciado pelo IBM MQ.NET fornece suporte de Segurança da Camada de Transporte (TLS) que é baseado no kit SSLStreams do Microsoft.NET. Isso é diferente dos outros clientes do IBM MQ, que são baseados em IBM Global Security Kit (GSKit)

É possível desenvolver aplicativos IBM MQ.NET para execução no modo gerenciado ou no modo não gerenciado.

- No modo gerenciado, os aplicativos .NET trabalham dentro do .NET CLR (Common Language Runtime) sem qualquer chamada de plataforma cruzada, tal como chamar o C MQI.
- No modo não gerenciado, o C MQI é chamado para as operações MQI subjacentes. Basicamente, a interface de modo não gerenciado compõe as classes de agrupador do .NET sobre C MQI.

O cliente IBM MQ.NET gerenciado usa as bibliotecas do Microsoft.NET Framework para implementar protocolos de soquete seguro TLS. A classe System.NET.Security.SSLStream do Microsoft é usada para implementar a segurança (TLS) no IBM MQ.NET.

O modo cliente do IBM MQ.NET não gerenciado já suporta o recurso TLS, que é baseado em C MQI (e GSKit). Ou seja, as operações de TLS são manipuladas pelo C MQI. Nesse caso, GSKit implementa os protocolos de soquete seguro TLS.

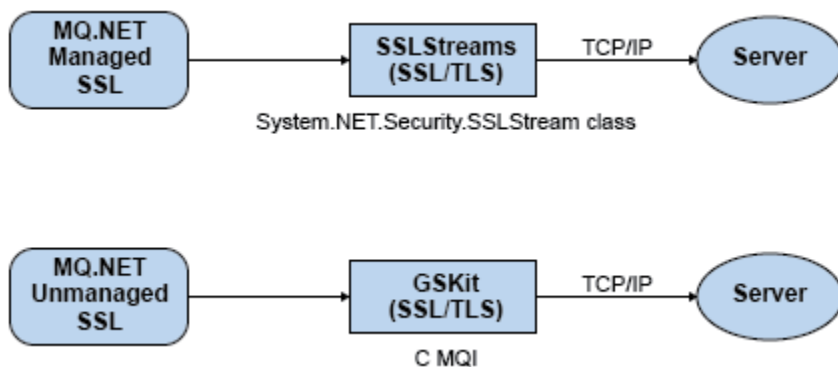


Figura 50. Comparação de TLS gerenciado e não gerenciado pelo IBM MQ.NET

A tabela a seguir resume as diferenças entre as implementações gerenciadas e não gerenciadas:

Tabela 14. Diferenças entre implementações gerenciadas e não gerenciadas

Modo	Protocolos	Implementação	Comments
SSL gerenciado pelo IBM MQ.NET	TLS	Classe System.NET.Security.SSLStream A classe SSLStream opera como um fluxo sobre um soquete TCP conectado	TLS 1.0 TLS 1.2 (com Microsoft.NET Framework v4.5 apenas)
SSL não gerenciado pelo IBM MQ.NET	TLS	GSKIT e C-MQI	Protocolos de soquete seguro TLS

Conceitos relacionados

Suporte de Secure Sockets Layer (SSL) e Segurança da Camada de Transporte (TLS) para .NET

IBM MQ MQI clients

Um IBM MQ MQI client é um componente do produto IBM MQ que pode ser instalado em um sistema no qual nenhum gerenciador de filas é executado.

Um IBM MQ MQI *cliente* é um componente que permite que um aplicativo em execução em um sistema emita chamadas MQI para um gerenciador de filas em execução em outro sistema. A saída da chamada é enviada novamente ao cliente, que a transmite novamente ao aplicativo.

Usando um IBM MQ MQI client, um aplicativo em execução no mesmo sistema que o cliente pode se conectar a um gerenciador de filas que está em execução em outro sistema. O aplicativo pode emitir chamadas MQI para esse gerenciador de fila. Esse aplicativo é chamado de um aplicativo IBM MQ MQI client e o gerenciador de filas é chamado de *gerenciador de filas*.

Um IBM MQ *servidor* é um gerenciador de filas que fornece serviços de enfileiramento para um ou mais clientes. Todos os objetos IBM MQ, por exemplo, filas, existem apenas na máquina do gerenciador de filas (a máquina servidor do IBM MQ) e não no cliente. Um servidor IBM MQ também pode oferecer suporte a aplicativos locais do IBM MQ.

A diferença entre um servidor IBM MQ e um gerenciador de filas comum é que um servidor possui um link de comunicações dedicadas com cada cliente. Para obter informações adicionais sobre como criar canais para clientes e servidores, consulte [Configurando o enfileiramento distribuído](#).

Um aplicativo IBM MQ MQI client e um gerenciador de filas do servidor se comunicam um com o outro usando um *canal MQI*. Um canal MQI é iniciado quando o aplicativo cliente emite uma chamada **MQCONN** ou **MQCONNX** para conectar-se ao gerenciador de filas e é finalizado quando o aplicativo cliente emite uma chamada **MQDISC** para se desconectar do gerenciador de fila. Os parâmetros de entrada de uma chamada MQI fluem em uma direção em um canal MQI e os parâmetros de saída fluem na direção oposta.

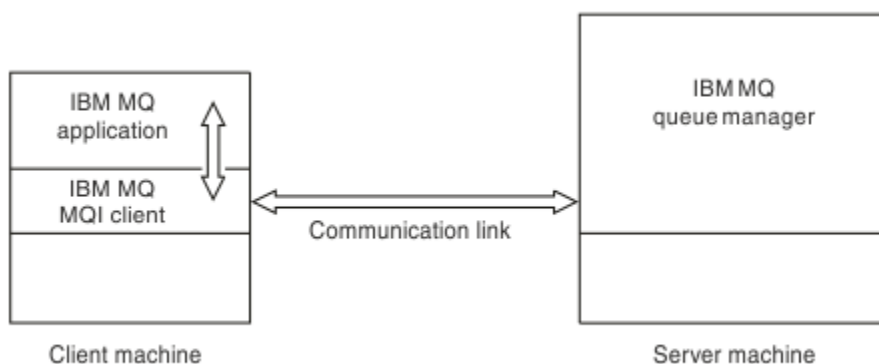


Figura 51. Link entre um Cliente e Servidor

As plataformas a seguir podem ser usadas. As combinações dependem de qual produto IBM MQ você está usando e estão descritas em [“Suporte à plataforma para clientes IBM MQ”](#) na página 146.

IBM MQ MQI client

AIX and Linux
Windows
IBM i

Servidor IBM MQ

AIX and Linux
Windows
IBM i
z/OS

O MQI está disponível para aplicativos em execução na plataforma cliente; as filas e outros objetos IBM MQ são retidos em um gerenciador de fila que você instalou em um servidor.

Um aplicativo que você quer executar no ambiente do IBM MQ MQI client deve ser vinculado primeiro à biblioteca do cliente relevante. Quando o aplicativo emite uma chamada MQI, o IBM MQ MQI client direciona a solicitação para um gerenciador de filas, em que ela é processada e de onde uma resposta é enviada de volta para o IBM MQ MQI client.

O link entre o aplicativo e o IBM MQ MQI client é estabelecido dinamicamente no tempo de execução.

Também é possível desenvolver aplicativos clientes usando o IBM MQ classes for .NET, o IBM MQ classes for Java ou o IBM MQ classes for Java Message Service (JMS) e também usar os clientes do Java e do JMS nas plataformas a seguir:

- IBM i
- AIX
- Linux
- Windows

O uso de Java e JMS não está descrito aqui. Para obter detalhes completos sobre como instalar, configurar e usar o IBM MQ classes for Java e IBM MQ classes for JMS, consulte [Usando o IBM MQ classes for Java](#) e [Usando o IBM MQ classes for JMS](#).

Aplicativos IBM MQ em um Ambiente Cliente-Servidor

Quando vinculado a um servidor, os aplicativos IBM MQ do cliente podem emitir a maioria das chamadas MQI da mesma maneira que os aplicativos locais. O aplicativo cliente emite uma chamada MQCONN para conectar-se a um gerenciador de filas especificado. Qualquer chamada MQI adicional que especificar o identificador de conexões retornado da solicitação de conexão é então processada por este gerenciador de filas.

Você deve vincular seus aplicativos para as bibliotecas de cliente apropriadas. Consulte [Construindo aplicativos para IBM MQ MQI clients](#).

Conceitos relacionados

[“Por que usar clientes IBM MQ?”](#) na página 146

O uso de clientes do IBM MQ é uma maneira eficiente de implementar o sistema de mensagens e o enfileiramento do IBM MQ.

[“O que É um Cliente Transacional Estendido?”](#) na página 148

Um cliente transacional estendido do IBM MQ pode atualizar os recursos gerenciados por outro gerenciador de recursos, sob o controle de um gerenciador de transações externo.

[“Como o Cliente se Conecta ao Servidor”](#) na página 149

Um cliente se conecta a um servidor usando MQCONN ou MQCONNX e se comunica por meio de um canal.

[“Gerenciamento de Transação e Suporte”](#) na página 150

Uma introdução ao gerenciamento de transação e como o IBM MQ suporta as transações.

[“Estendendo as instalações do gerenciador de filas”](#) na página 152

É possível estender os recursos do gerenciador de filas usando saídas do usuário, saídas da API ou serviços instaláveis.

Informações relacionadas

[Como configurar um IBM MQ MQI client](#)

Por que usar clientes IBM MQ?

O uso de clientes do IBM MQ é uma maneira eficiente de implementar o sistema de mensagens e o enfileiramento do IBM MQ.

Você pode ter um aplicativo que usa o MQI em execução em uma máquina e o gerenciador de filas em execução em uma máquina diferente (física ou virtual). Os benefícios de fazer isto são:

- Não há a necessidade de uma implementação completa do IBM MQ na máquina cliente.
- Os requisitos de hardware no sistema do cliente foram reduzidos.
- Os requisitos de administração do sistema foram reduzidos.
- Um aplicativo IBM MQ em execução em um cliente pode se conectar a múltiplos gerenciadores de filas em sistemas diferentes.
- Canais alternativos usando diferentes protocolos de transmissão podem ser usados.

Suporte à plataforma para clientes IBM MQ

O IBM MQ em todas as plataformas do servidor suportadas aceita as conexões do cliente do IBM MQ MQI clients em várias plataformas.

O IBM MQ instalado como um *Produto base e servidor* em todas as plataformas do servidor suportadas pode aceitar conexões dos IBM MQ MQI clients nas plataformas a seguir:

-  IBM i
-  AIX
-  Linux
-  Windows

As conexões do cliente estão sujeitas a diferenças no coded character set identifier (CCSID) e no protocolo de comunicação.

Quais aplicativos executar em um IBM MQ MQI client?

O MQI completo é suportado no ambiente do cliente. Isso permite que quase qualquer aplicativo IBM MQ seja configurado para execução em um sistema IBM MQ MQI client vinculando o aplicativo no IBM MQ MQI client à biblioteca MQIC, e não à biblioteca MQI. As exceções são:

- MQGET com sinal
- Um aplicativo que precisa de coordenação do ponto de sincronização com outros gerenciadores de recursos deve usar um cliente transacional estendido

Se a leitura antecipada estiver ativada, para melhorar o desempenho do sistema de mensagens persistentes, nem todas as opções MQGET estarão disponíveis. A tabela mostra as opções que são permitidas e se elas podem ser alteradas entre chamadas MQGET.

Tabela 15. Opções MQGET Permitidas quando a Leitura Antecipada Está Ativada

Valores	Permitido quando a leitura antecipada está ativada e pode ser alterada entre chamadas MQGET	Permitido quando a leitura antecipada está ativada e não pode ser alterada entre chamadas MQGET ¹	Opções MQGET que não são permitidas quando a leitura antecipada estiver ativada ²
Valores de MQGET MD	MsgId ³ CorrelId ³	Codificação CodedCharSetId	
Opções MQGET MQGMO	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF QUIESCING MQGMO_BROWSE_FIRST ⁴ MQGMO_BROWSE_NEXT ⁴ MQGMO_BROWSE_MESSAGE_UNDER_CURSOR ⁴	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQR FH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL MQGMO_SYNCPOINT MQGMO_MARK_SKIP_BACKOUT MQGMO_MSG_UNDER_CURSOR ⁴ MQGMO_LOCK MQGMO_UNLOCK
Valores de MQGMO		MsgHandle	

1. Se estas opções forem alteradas entre chamadas MQGET, um código de razão MQRC_OPTIONS_CHANGED será retornado.
2. Se estas opções forem especificadas na primeira chamada MQGET, a leitura antecipada é desativada. Se essas opções forem especificadas em uma chamada MQGET subsequente, um código de razão MQRC_OPTIONS_ERROR será retornado.
3. Os aplicativos clientes precisam estar cientes que, se os valores de MsgId e CorrelId forem alterados entre chamadas MQGET, mensagens com os valores anteriores poderão já ter sido enviadas ao cliente e permanecerão no buffer de leitura antecipada do cliente até serem consumidos (ou limpos automaticamente).
4. A primeira chamada MQGET determina se mensagens devem ser navegadas ou obtidas a partir de uma fila quando a leitura antecipada está ativada. Se o aplicativo tentar usar uma combinação de navegação e obtenção, um código de razão MQRC_OPTIONS_CHANGED será retornado.
5. MQGMO_MSG_UNDER_CURSOR não é possível com a leitura antecipada. As mensagens podem ser navegadas ou obtidas quando a leitura antecipada é ativada, mas não uma combinação de ambos.

Um aplicativo em execução em um IBM MQ MQI client pode se conectar a mais de um gerenciador de filas simultaneamente ou usar um nome de gerenciador de filas com um asterisco (*) em uma chamada MQCONN ou MQCONNX (consulte os exemplos em [Conectando aplicativos IBM MQ MQI client aos gerenciadores de filas](#)).

O que É um Cliente Transacional Estendido?

Um cliente transacional estendido do IBM MQ pode atualizar os recursos gerenciados por outro gerenciador de recursos, sob o controle de um gerenciador de transações externo.

Se você não estiver familiarizado com os conceitos de gerenciamento de transações, consulte [“Gerenciamento de Transação e Suporte”](#) na página 150.

Observe que o cliente transacional XA agora é fornecido como parte do IBM MQ.

Um aplicativo cliente pode participar de uma unidade de trabalho gerenciada por um gerenciador de filas ao qual ele está conectado. Na unidade de trabalho, o aplicativo cliente pode colocar mensagens em, e obter mensagens de, filas que são pertencentes a esse gerenciador de filas. O aplicativo cliente pode então usar a chamada **MQCMIT** para confirmar a unidade de trabalho ou a chamada **MQBACK** para restaurar a unidade de trabalho. Entretanto, na mesma unidade de trabalho, o aplicativo cliente não pode atualizar os recursos de outro gerenciador de recursos, as tabelas de um banco de dados do Db2, por exemplo. O uso de um cliente transacional estendido do IBM MQ remove essa restrição.


Um cliente transacional estendido do IBM MQ é um IBM MQ MQI client com alguma função adicional. Usando esta função um aplicativo cliente, na mesma unidade de trabalho, pode executar as seguintes tarefas:

- Colocar mensagens em, e obter mensagens de, filas pertencentes ao gerenciador de filas ao qual ele está conectado
- Atualizar os recursos de um gerenciador de recursos diferente de um gerenciador de filas do IBM MQ

Esta unidade de trabalho deve ser gerenciada por um gerenciador de transações externo que está em execução no mesmo sistema que o aplicativo cliente. A unidade de trabalho não pode ser gerenciada pelo gerenciador de filas ao qual o aplicativo cliente está conectado. Isto significa que o gerenciador de filas pode agir somente como um gerenciador de recursos, não como um gerenciador de transações. Também significa que o aplicativo cliente pode confirmar ou efetuar backout da unidade de trabalho usando somente a interface de programação de aplicativos (API) fornecida pelo gerenciador de transações externo. O aplicativo cliente não pode, portanto, usar as chamadas MQI, **MQBEGIN**, **MQCMIT** e **MQBACK**.

O gerenciador de transações externo se comunica com o gerenciador de filas como um gerenciador de recursos usando o mesmo canal MQI que o usado pelo aplicativo cliente que está conectado ao gerenciador de filas. Entretanto, em uma situação de recuperação seguinte a uma falha, quando nenhum aplicativo está em execução, o gerenciador de transações pode usar um canal MQI dedicado para recuperar qualquer unidade de trabalho incompleta na qual o gerenciador de filas estava participando no momento da falha.

Nesta seção, um IBM MQ MQI client que não possui a função transacional estendida é referido como um cliente de base do IBM MQ. É possível considerar, portanto, um cliente transacional estendido do IBM MQ consistindo em um cliente de base do IBM MQ com a inclusão da função transacional estendida.

Nota:  IBM MQ MQI client no IBM i não suporta a função transacional estendida do IBM MQ.

Suporte à Plataforma para Clientes Transacionais Estendidos


Multi

Clientes transacionais estendidos estão disponíveis para todas as Multiplataformas que suportam um cliente de base. Os clientes não estão disponíveis para o z/OS.

Um aplicativo cliente que está usando um cliente transacional estendido pode se conectar a um gerenciador de filas dos seguintes produtos IBM MQ 9.0 ou mais recentes:

-  IBM MQ for AIX

-  IBM MQ for IBM i
-  IBM MQ for Linux
-  IBM MQ for Windows

 Embora não haja clientes transacionais estendidos que são executados no z/OS, um aplicativo cliente que está usando um cliente transacional estendido pode se conectar a um gerenciador de filas que é executado no z/OS.

Para cada plataforma, os requisitos de hardware e software para o cliente transacional estendido são iguais àqueles requisitos para o cliente base do IBM MQ. Uma linguagem de programação é suportada por um cliente transacional estendido se ele é suportado pelo cliente base do IBM MQ e pelo gerenciador de transações que está sendo usado.

Para obter informações sobre os gerenciadores de transações externas para todas as plataformas, consulte [Requisitos do sistema para IBM MQ](#)

Como o Cliente se Conecta ao Servidor

Um cliente se conecta a um servidor usando MQCONN ou MQCONNX e se comunica por meio de um canal.

Um aplicativo em execução no ambiente do cliente IBM MQ deve manter uma conexão ativa entre as máquinas do cliente e do servidor.

A conexão é feita por um aplicativo emitindo uma chamada MQCONN ou MQCONNX. Clientes e servidores se comunicam por meio de *canais MQI* ou, ao usar conversações de compartilhamento, cada conversação compartilha uma instância do canal MQI. Quando a chamada é bem-sucedida, a instância do canal MQI ou conversação permanece conectada até que o aplicativo emita uma chamada MQDISC. Este é o caso para cada gerenciador de filas ao qual um aplicativo precisa se conectar.

Cliente e Gerenciador de Filas na Mesma Máquina

Também é possível executar um aplicativo no ambiente do IBM MQ MQI client quando a sua máquina também tem um gerenciador de filas instalado.

Nesta situação, você tem a opção de vincular-se às bibliotecas do gerenciador de filas ou às bibliotecas do cliente, mas lembre-se que se você vincular-se às bibliotecas do cliente, ainda precisará definir as conexões de canal. Isto pode ser útil durante a fase de desenvolvimento de um aplicativo. É possível testar seu programa em sua própria máquina, sem dependência de outros, e ter a certeza de que ele ainda funcionará quando você movê-lo para um ambiente independente do IBM MQ MQI client.

Clientes em Plataformas Diferentes

Neste exemplo, a máquina servidor se comunica com três IBM MQ MQI clients em diferentes plataformas.

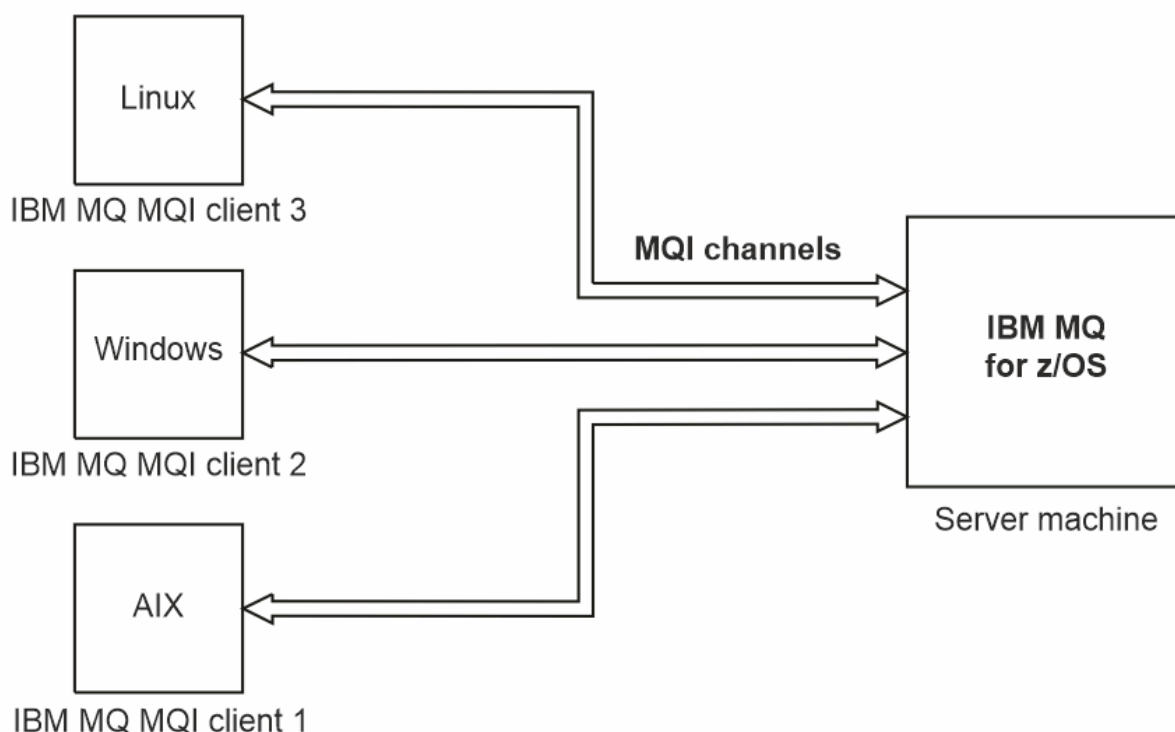


Figura 52. Servidor IBM MQ conectado aos clientes em diferentes plataformas

Outros ambientes mais complexos são possíveis. Por exemplo, um cliente do IBM MQ pode se conectar a mais de um gerenciador de filas ou a qualquer número de gerenciadores de filas conectados como parte de um grupo de filas compartilhadas.

Usando Versões Diferentes do Software do Cliente e do Servidor

Se você estiver usando versões anteriores dos produtos IBM MQ, certifique-se de que a conversão de código do CCSID do seu cliente seja suportada pelo servidor.

Um cliente do IBM MQ pode se conectar a todas as versões suportadas do gerenciador de fila. Se você estiver conectando-se a um gerenciador de filas de versão anterior, não será possível usar recursos e estruturas de uma versão mais recente do produto em seu aplicativo IBM MQ no cliente.

Um gerenciador de filas do IBM MQ pode se comunicar com clientes em diferentes versões para si mesmo negociando até o nível de protocolo mais alto mutuamente suportado. Isso significa que os clientes mais antigos podem ser usados com níveis mais recentes do gerenciador de filas. Recomenda-se que o cliente e o servidor estejam em versões do IBM MQ que estão atualmente no suporte para facilitar o diagnóstico de problemas e ativar o suporte por IBM.

Para obter mais informações, consulte as linguagens de programação suportadas em [Desenvolvendo aplicativos](#).

Gerenciamento de Transação e Suporte

Uma introdução ao gerenciamento de transação e como o IBM MQ suporta as transações.

Um *gerenciador de recursos* é um subsistema do computador que possui e gerencia recursos que podem ser acessados e atualizados por aplicativos. Aqui estão exemplos de gerenciadores de recursos:

- Um gerenciador de filas do IBM MQ, com recursos que são suas filas
- Um Db2 do banco de dados, com recursos que são suas tabelas

Quando um aplicativo atualiza os recursos de um ou mais gerenciadores de recursos, pode haver um requisito de negócios para assegurar que determinadas atualizações sejam concluídas com êxito como um grupo ou que nenhuma delas seja concluída. A razão para este tipo de requisito é que os dados de negócios seriam deixados em um estado inconsistente se algumas destas atualizações fossem concluídas com êxito mas outras não.

As atualizações nos recursos que são gerenciados desta maneira são para ocorrer em uma *unidade de trabalho* ou em uma *transação*. Um programa de aplicativo pode agrupar um conjunto de atualizações em uma unidade de trabalho.

Durante uma unidade de trabalho, um aplicativo emite pedidos para gerenciadores de recursos atualizarem seus recursos. A unidade de trabalho é finalizada quando o aplicativo emite um pedido para confirmar todas as atualizações. Até as atualizações serem confirmadas, nenhuma delas se torna visível para outros aplicativos que estão acessando os mesmos recursos. Alternativamente, se o aplicativo decidir que ele não pode concluir a unidade de trabalho por qualquer razão, ele poderá emitir um pedido para restaurar todas as atualizações que ele solicitou até esse ponto. Neste caso, nenhuma das atualizações se tornará visível para outros aplicativos. Essas atualizações são relacionadas logicamente e devem ser todas bem-sucedidas para que a integridade de dados seja preservada. Se uma atualização for bem-sucedida enquanto outra falhar, a integridade de dados será perdida.

Quando uma unidade de trabalho for concluída com êxito, será considerada *confirmada*. Uma vez confirmadas, todas as atualizações feitas nessa unidade de trabalho serão permanentes e irreversíveis. No entanto, se a unidade de trabalho falhar, todas as atualizações serão *revertidas*. Este processo, no qual as unidades de trabalho são confirmadas ou revertidas com integridade, é conhecido como *coordenação do ponto de sincronização*.

O momento em que todas as atualizações em uma unidade de trabalho são confirmadas ou revertidas é chamado de *sync point*. Uma atualização em uma unidade de trabalho deve ocorrer *no controle do ponto de sincronização*. Se um aplicativo solicitar uma atualização que está *fora do controle do ponto de sincronização*, o gerenciador de recursos confirmará a atualização imediatamente, mesmo se houver uma unidade de trabalho em progresso e a atualização não puder ser revertida posteriormente.

O subsistema do computador que gerencia unidades de trabalho é chamado de *gerenciador de transações*, ou *ponto de coordenador*.

Uma unidade de trabalho *local* é aquela na qual os únicos recursos atualizados são aqueles do gerenciador de filas do IBM MQ. Aqui a coordenação do ponto de sincronização é fornecida pelo próprio gerenciador de filas usando um processo de confirmação de fase única.

Uma unidade *global* *global* é aquela na qual os recursos que pertencem a outros gerenciadores de recursos, como bancos de dados compatíveis com XA, também são atualizados. Aqui, um procedimento two-phase commit deve ser usado e a unidade de trabalho pode ser coordenada pelo próprio gerenciador de fila ou externamente por outro gerenciador de transações compatível com XA como IBM TXSeries ou BEA Tuxedo.

Um gerenciador de transações é responsável por assegurar que todas as atualizações nos recursos em uma unidade de trabalho sejam concluídas com êxito ou que nenhuma delas seja concluída. É para um gerenciador de transações que um aplicativo emite um pedido para confirmar ou restaurar uma unidade de trabalho. Exemplos de gerenciadores de transações são CICS e WebSphere Application Server, embora ambos tenham outra função também.

Alguns gerenciadores de recursos fornecem sua própria função de gerenciamento de transação. Por exemplo, um gerenciador de filas do IBM MQ pode gerenciar unidades de trabalho que envolvem atualizações em seus próprios recursos e atualizações para tabelas do Db2. O gerenciador de filas não precisa de um gerenciador de transações separado para executar esta função, embora um possa ser usado se ele for um requisito do usuário. Se um gerenciador de transações separado for usado, ele será referido como um *gerenciador de transações externo*.

Para um gerenciador de transações externo gerenciar uma unidade de trabalho, deve haver uma interface padrão entre o gerenciador de transações e cada gerenciador de recursos que está participando da unidade de trabalho. Esta interface permite que o gerenciador de transações e um gerenciador de recursos se comuniquem entre si. Uma destas interfaces é a *Interface XA*, que é uma interface padrão

suportada por vários gerenciadores de transações e gerenciadores de recursos. A Interface XA é publicada pelo The Open Group em *Processamento de Transações Distribuídas: A Especificação XA*.

Quando mais de um gerenciador de recursos participa de uma unidade de trabalho, um gerenciador de transações deve usar um protocolo *two-phase commit* para assegurar que todas as atualizações na unidade de trabalho sejam concluídas com êxito ou que nenhuma delas seja concluída, mesmo se houver uma falha do sistema. Quando um aplicativo emitir um pedido para um gerenciador de transações confirmar uma unidade de trabalho, o gerenciador de transações fará o seguinte:

Fase 1 (Preparação para Confirmação)

O gerenciador de transações solicita a cada gerenciador de recursos que participa da unidade de trabalho para assegurar que todas as informações sobre as atualizações destinadas aos seus recursos estejam em um estado recuperável. Um gerenciador de recursos normalmente faz isto gravando as informações em um log e assegurando que as informações sejam gravadas no disco rígido. A Fase 1 é concluída quando o gerenciador de transações recebe notificação de cada gerenciador de recursos de que as informações sobre as atualizações destinadas aos seus recursos estão em um estado recuperável.

Fase 2 (Confirmação)

Quando a Fase 1 estiver concluída, o gerenciador de transações tomará a decisão irrevogável de confirmar a unidade de trabalho. Ele solicitará a cada gerenciador de recursos que participa da unidade de trabalho para confirmar as atualizações em seus recursos. Quando um gerenciador de recursos receber este pedido, ele deverá confirmar as atualizações. Ele não tem a opção de restaurar este estágio. A Fase 2 é concluída quando o gerenciador de transações recebe notificação de cada gerenciador de recursos de que ele confirmou as atualizações em seus recursos.

A Interface XA um protocolo two-phase commit.

Para obter mais informações, consulte [Cenários de suporte transacional](#).

O IBM MQ também fornece suporte para o Microsoft Transaction Server (COM+). O uso do Microsoft Transaction Server (COM+) fornece informações sobre como configurar o IBM MQ para aproveitar o suporte do COM+.

Estendendo as instalações do gerenciador de filas

É possível estender os recursos do gerenciador de filas usando saídas do usuário, saídas da API ou serviços instaláveis.

Saídas do Usuário

As saídas do usuário fornecem um mecanismo para que você insira seu próprio código em uma função do gerenciador de filas. As saídas de usuário suportadas incluem:

Saídas do canal

Essas saídas alteram a maneira como os canais operam. As saídas de canal são descritas em [Programas de Saída do Canal para Canais de Mensagem](#).

Saídas de conversão de dados

Essas saídas criam fragmentos do código de origem que podem ser colocados nos programas de aplicativos para converter os dados de um formato para outro. As saídas de conversão de dados estão descritas em [Gravando saídas de conversão de dados](#).

A saída da carga de trabalho do cluster

A função executada por esta saída é definida pelo provedor da saída. As informações de definição de chamada são fornecidas em [MQ_CLUSTER_WORKLOAD_EXIT - Descrição da Chamada](#).

Saídas de API

As saídas de API permitem gravar o código que altera o comportamento das chamadas de API do IBM MQ, como MQPUT e MQGET, e, em seguida, inserem esse código imediatamente antes ou depois dessas chamadas. A inserção é automática; o gerenciador de filas conduz o código de saída nos pontos

registrados. Para obter informações adicionais sobre as saídas de API, consulte [Usando e Gravando as Saídas de API](#).

Serviços instaláveis

Os serviços instaláveis possuem interfaces formalizadas (uma API) com diversos pontos de entrada.

Uma implementação de um serviço instalável é denominada um *componente de serviço*. É possível usar os componentes fornecidos com o IBM MQ, ou você pode gravar seu próprio componente para executar as funções que precisa.

Atualmente, são fornecidos os seguintes serviços instaláveis:

Serviço de autorização

O serviço de autorização permite construir seu próprio recurso de segurança.

O componente de serviço padrão que implementa o serviço é o Object Authority Manager (OAM). Por padrão, o OAM fica ativo e você não tem que fazer nada para configurá-lo. É possível usar a interface do serviço de autorização para criar outros componentes para substituir ou aumentar o OAM. Para obter mais informações sobre o OAM, consulte [Configurando segurança em sistemas AIX, Linux, and Windows](#).

Serviço de Nomes

O serviço de nomes permite que os aplicativos compartilhem as filas identificando as filas remotas como se fossem filas locais.

É possível gravar seu próprio componente de serviço de nomes. Talvez você queira fazer isso se pretender usar o serviço de nomes com IBM MQ, por exemplo. Para usar o serviço de nomes você deve ter um componente que seja gravado pelo usuário ou fornecido por um fornecedor de software diferente. Por padrão, o serviço de nomes está inativo.



Conceitos relacionados

[Saídas de usuário, saídas de API e serviços instaláveis do IBM MQ](#)

Interfaces de linguagem do IBM MQ Java

O IBM MQ fornece três interfaces de programação de aplicativos (APIs) para uso em Java aplicativos: IBM MQ classes for Jakarta Messaging, IBM MQ classes for JMS e IBM MQ classes for Java.

O IBM suporta e é um participante ativo em padrões abertos.

- No IBM MQ 8.0, o produto implementa o padrão JMS 2.0, que introduziu uma nova API simplificada juntamente com recursos como assinaturas compartilhadas.
-   Em IBM MQ 9.3.0, [Jakarta Messaging 3.0](#) também é suportado.
- Além disso, o WebSphere Liberty suporta JMS 2.0 e Jakarta Messaging 3.0 com IBM MQ.

No IBM MQ há três APIs para uso em aplicativos Java :

IBM MQ classes for Jakarta Messaging

IBM MQ classes for Jakarta Messaging é um provedor Jakarta Messaging que implementa a interface Jakarta Messaging para IBM MQ como o sistema de mensagens. O Jakarta Connectors Architecture fornece uma maneira padrão de conectar aplicativos em execução em um ambiente do Jakarta EE a um Enterprise Information System (EIS), como IBM MQ ou Db2.

IBM MQ classes for JMS

IBM MQ classes for JMS é um provedor JMS que implementa a interface JMS para IBM MQ como o sistema de mensagens. O Java Platform, Enterprise Edition Connector Architecture (JCA) fornece uma maneira padrão de conectar aplicativos em execução em um ambiente Java EE a um Enterprise Information System (EIS) como IBM MQ ou Db2.

IBM MQ classes for Java

O IBM MQ classes for Java permite usar o IBM MQ em um ambiente Java. O IBM MQ classes for Java permite que um aplicativo Java se conecte ao IBM MQ como um cliente IBM MQ ou se conecte diretamente a um gerenciador de filas do IBM MQ.

Nota:

- ▶ **V 9.3.0** ▶ **V 9.3.0** O JMS 2.0 foi substituído por Jakarta Messaging O IBM MQ classes for JMS continua a suportar o padrão JMS 2.0 , mas aprimoramentos futuros no sistema de mensagens do Java surgirão apenas no Jakarta Messaging, portanto, no IBM MQ classes for Jakarta Messaging. IBM MQ classes for JMS são recomendados apenas para manter e estender aplicativos JMS 2.0 existentes. IBM MQ classes for Jakarta Messaging deve ser a tecnologia preferencial para novo desenvolvimento.
- ▶ **Stabilized** O IBM MQ classes for Java é funcionalmente estabilizado no nível enviado no IBM MQ 8.0. Aplicativos existentes que usam o IBM MQ classes for Java continuarão a ser totalmente suportados, mas essa API ficará estabilizada, portanto, novos recursos não serão incluídos e solicitações de aprimoramentos rejeitadas. "Totalmente suportado" significa que os defeitos serão corrigidos, além de quaisquer mudanças necessárias, por meio de mudanças nos requisitos do sistema IBM MQ.

▶ **V 9.3.0** ▶ **JM 3.0** ▶ **V 9.3.0** Em IBM MQ 9.3, o IBM MQ classes for Java, o IBM MQ classes for JMS e o IBM MQ classes for Jakarta Messaging são construídos com o Java 8. Os ambientes de tempo de execução do Java nesses ou acima desses níveis devem ser usados para executar aplicativos usando essas interfaces

Conceitos relacionados

[Acessando o IBM MQ de Java -Opção de API](#)

▶ **V 9.3.0** ▶ **V 9.3.0** [Por que devo usar as classes do IBM MQ para Jakarta Messaging?](#)

[Por que devo usar o IBM MQ classes for JMS?](#)

[Por que devo usar o IBM MQ classes for Java?](#)

IBM MQ classes for JMS/Jakarta Messaging

IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging são os provedores de sistemas de mensagens fornecidos com IBM MQ. Cada um desses provedores também fornece dois conjuntos para a API do sistema de mensagens. Os aplicativos Java Platform, Standard Edition (Java SE) e Java Platform, Enterprise Edition (Java EE) podem usar esses provedores de sistemas de mensagens.

▶ **V 9.3.0** ▶ **JM 3.0** ▶ **V 9.3.0** IBM MQ 9.3.0 introduz o suporte para [Jakarta Messaging 3.0](#). JMS 2.0 ainda é totalmente suportado.

As especificações JMS e Jakarta Messaging definem um conjunto de interfaces que os aplicativos podem usar para executar operações do sistema de mensagens Do IBM MQ 8.0 em diante, o produto suporta a versão do JMS 2.0 do JMS padrão. Essa implementação oferece todos os recursos da API clássica, mas requer menos interfaces e é mais simples de usar. Para obter mais informações, consulte [“JMS e Jakarta Messaging modelo” na página 158](#) e a especificação do JMS 2.0 em [Java.net](#). ▶ **JM 3.0** Em IBM MQ 9.3.0, Jakarta Messaging também é suportado.

O pacote jakarta.jms ([Jakarta Messaging 3.0](#)) ou javax.jms (JMS 2.0) especifica os detalhes das interfaces de sistema de mensagens e um provedor de sistemas de mensagens implementa essas interfaces para um produto de sistema de mensagens específico. Por exemplo:

- O IBM MQ classes for JMS é um provedor do JMS que implementa as interfaces do JMS para o IBM MQ e também fornece os dois conjuntos de extensões a seguir para a API do JMS:
 - Extensões do IBM MQ JMS
 - Extensões do IBM JMS
- Um connection factory, fila ou objeto de tópico criado usando as interfaces javax.jms ou jakarta.jms ou um conjunto de extensões JMS pode ser endereçado usando qualquer uma dessas APIs, ou seja, ele

pode ser convertido para qualquer uma das interfaces. Para manter a portabilidade do aplicativo no nível mais alto, use a API mais genérica adequada para seus requisitos.

Como JMS e Jakarta Messaging compartilham muito em comum, referências adicionais a JMS neste tópico podem ser consideradas referentes a ambos. Quaisquer diferenças são destacadas conforme necessário.

Extensões do IBM MQ JMS

IBM MQ classes for JMS também fornece extensões para a API do JMS. O IBM MQ classes for JMS contém extensões que são implementadas nos objetos MQConnectionFactory, MQQueue e MQTopic. Esses objetos possuem propriedades e métodos que são específicos para IBM MQ. Os objetos podem ser objetos administrados ou um aplicativo pode criar os objetos dinamicamente no tempo de execução. Essas extensões são denominadas extensões IBM MQ JMS. Observe que, nesta documentação, os objetos criados dinamicamente por um aplicativo no tempo de execução não são considerados objetos administrados.

Extensões do IBM JMS

Além das extensões IBM MQ JMS, IBM MQ classes for JMS fornece um conjunto mais genérico de extensões para a JMS API ou Java como a linguagem de programação usada. Essas extensões são denominadas extensões do IBM JMS e têm os objetivos amplos a seguir:

- Para fornecer um nível maior de consistência entre os provedores do IBM JMS
- Para facilitar a gravação de um aplicativo de ponte entre dois sistemas de mensagens do IBM.
- Para facilitar a porta de um aplicativo de um provedor do IBM JMS para outro.

O principal foco dessas extensões refere-se à criação e configuração de connection factories e destinos dinamicamente no tempo de execução, mas as extensões também fornecem a função que não está diretamente relacionada ao sistema de mensagens, como a função para determinação de problema.

Tarefas relacionadas

Usando classes do IBM MQ para o Sistema de Mensagens JMS/Jakarta
[Configurando recursos do JMS e do Jakarta Messaging](#)

IBM MQ classes for Jakarta Messaging: uma visão geral

IBM MQ 9.3.0 apresenta suporte para Jakarta Messaging. Para Jakarta Messaging 3.0, o controle da especificação JMS foi movido de Oracle para o Java Community Process. No entanto, o Oracle retém o controle do nome "javax", que é usado em outras tecnologias Java. Portanto, embora Jakarta Messaging 3.0 seja funcionalmente equivalente a JMS 2.0, há algumas diferenças na nomenclatura.. O nome oficial para a versão 3.0 é Jakarta Messaging em vez de Java Message Servicee os nomes de pacotes e constantes são prefixados com `jakarta` em vez de `javax`.

Plano de fundo

Por muitos anos, a plataforma Java veio em dois formulários- Standard Edition e Enterprise Edition.

Java Platform, Standard Edition (às vezes abreviado como Java SE) é a linguagem principal e bibliotecas de classe, capaz de ser executado em um contexto independente. A maioria dos pacotes Java em Java SE têm nomes que começam com "java".

Java Platform, Enterprise Edition (Java EE) estende isso, incluindo funcionalidade como Sistema de Mensagens, vários Beans, transacional e assim por diante. Algumas dessas tecnologias também podem ser usadas em um contexto do Java SE A maioria dos pacotes Java no Java EE historicamente têm nomes que começam com "javax."-há algum cruzamento, no entanto, alguns pacotes Java SE têm "javax." como o prefixo para o nome deles.

O Java Message Service (JMS) faz parte de Java Platform, Enterprise Edition Java EE 7 incorpora JMS 2.0.

Até o Java EE 7, as tecnologias estavam sob a administração do Oracle

As tecnologias Java EE mudaram recentemente da administração do Oracle para um processo da comunidade supervisionado pela Fundação Eclipse .

Como o "javax". O nome não pôde ser movido para o novo projeto, a nova nomenclatura foi adotada-todos os pacotes e nomes de propriedades agora são prefixados com "jakarta". e o Java Platform, Enterprise Edition será chamado de "Jakarta EE" no futuro. A numeração da versão continuou: a versão 8 foi uma versão provisória que pode ser amplamente ignorada, e Jakarta EE 9 é o ponto em que o "jakarta". O prefixo entra em vigor

A principal tecnologia Jakarta EE que se aplica no contexto IBM MQ é Jakarta Messaging 3.0 -o sucessor de Java Message Service (JMS) 2.0. Portanto, o Jakarta EE 9 incorpora Jakarta Messaging 3.0

O IBM MQ continua a suportar Java EE 7 e JMS 2.0, enquanto introduz suporte para Jakarta EE 9 e Jakarta Messaging 3.0.

O que é entregue: Java SE

Para o Java Platform, Standard Edition, além do IBM MQ classes for JMS (que suporta operações JMS 2.0 com IBM MQ) IBM MQ 9.3.0 fornece IBM MQ classes for Jakarta Messaging. Essas classes fornecem um provedor Jakarta Messaging 3.0 que se integra ao IBM MQ, permitindo o uso de gerenciadores de filas do IBM MQ para facilitar operações do Jakarta Messaging .

Elas são fornecidas como um arquivo JAR padrão, com `ibm.mq.jakarta.client.jar`, no subdiretório `java/lib` da instalação do IBM MQ

Para uso em contêineres OSGi, como Apache Felix ou Eclipse Equinox, IBM MQ também fornece um par de pacotes configuráveis OSGi:

- `com.ibm.mq.osgi.jms30.clientprereqs_V.R.M.F.jar`
- `com.ibm.mq.osgi.jms30.client_V.R.M.F.jar`

em que *V.R.M.F* representa a versão de IBM MQ, por exemplo 9.3.0.0. Esses pacotes configuráveis podem ser localizados no subdiretório `java/lib/OSGi` da instalação IBM MQ .

O que é entregue: Jakarta EE 9

Para suportar o sistema de mensagens baseado no IBM MQ em um Jakarta EE 9 servidor de aplicativos compatível, o IBM MQ fornece um Jakarta EE 9 Adaptador de Recursos compatível: `wmq.jakarta.jmsra.rar`. Isso pode ser localizado no subdiretório `java/lib/jca` da instalação IBM MQ .

O IBM MQ continua a fornecer um Java EE 7 Adaptador de Recursos compatível, `wmq.jmsra.rar`, no subdiretório `java/lib/jca` da instalação do IBM MQ

Como esses artefatos são entregues?

Esses JARs e o arquivo RAR para o Adaptador de Recursos são empacotados com os artefatos pré-existent na mídia de instalação usual do IBM MQ -a mídia de instalação específica da plataforma, como arquivos ".rpm" e a mídia redistribuível, como os arquivos JAR do cliente redistribuíveis autoextraídos.

O que mudou entre JMS 2.0 e Jakarta Messaging 3.0

Jakarta EE 9 e Jakarta Messaging 3.0 não introduzem nenhuma nova funcionalidade Tudo o que muda são nomes. Por exemplo, quando você usa "javax.jms.Connection" em JMS 2.0, usa "jakarta.jms.Connection" em Jakarta Messaging 3.0.

À medida que a Eclipse Foundation leva a plataforma Jakarta EE adiante, ela será construída sobre essa base e essa convenção de nomenclatura será usada para novas funcionalidades introduzidas no futuro

O que mudou entre IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging

Resumo

O IBM MQ classes for JMS, que fornece suporte para o JMS 2.0, permanece disponível e é recomendado principalmente para manter e estender aplicativos existentes. Eles são totalmente suportados.

O IBM MQ classes for Jakarta Messaging, que fornece suporte para o Jakarta Messaging 3.0, é recomendado para novo desenvolvimento.

Em IBM MQ 9.3.0, essas duas ofertas são funcionalmente equivalentes.. Apenas a nomenclatura é diferente No entanto, é mais provável que a nova funcionalidade do sistema de mensagens surja no IBM MQ classes for Jakarta Messaging do que no IBM MQ classes for JMS

As duas ofertas são interoperáveis. As mensagens produzidas por IBM MQ classes for JMS podem ser consumidas por IBM MQ classes for Jakarta Messaging vice-versa Mas as duas ofertas não devem coexistir em um único aplicativo.

Mudanças de nomenclatura

<i>Tabela 16. Mudanças nos nomes de pacotes..</i>	
IBM MQ classes for JMS nome do pacote	IBM MQ classes for Jakarta Messaging nome do pacote
com.ibm.mq.jms[*]	com.ibm.mq.jakarta.jms[*]
com.ibm.jms	com.ibm.jakarta.jms
com.ibm.msg.client.jms.*	com.ibm.msg.client.jakarta.jms.*
com.ibm.msg.client.wmq.*	com.ibm.msg.client.jakarta.wmq.*

Os pacotes relacionados a serviços comuns (rastreamento, criação de log, suporte de idioma nacional, etc.) e as implementações de JMQL (local e remoto) são comuns ao IBM MQ classes for JMS e ao IBM MQ classes for Jakarta Messaging, portanto, nenhuma mudança é necessária nessas áreas.

Observe que os nomes de propriedades também foram alterados Por exemplo, a propriedade para ativar as extensões IBM MQ em IBM MQ classes for Jakarta Messaging é **com.ibm.mq.jakarta.jms.SupportMQExtensions**

Nomes de propriedades que são independentes de IBM MQ classes for JMS ou IBM MQ classes for Jakarta Messaging, como as várias propriedades **com.ibm.msg.client.commonservices.trace.***, se aplicam igualmente a ambas as ofertas.

Utilitários Administrativos

Os utilitários **crtmqenv** e **setmqenv** agora aceitam uma opção para especificar se o caminho de classe deve ser configurado para IBM MQ classes for JMS (-j 2.0) ou IBM MQ classes for Jakarta Messaging (-j 3.0) e há IBM MQ classes for Jakarta Messaging variantes dos utilitários **runjms**, chamados **runjms30** e nomes semelhantes.

O utilitário **dspmqver**, quando solicitado a relatar sobre componentes Java, inclui IBM MQ classes for Jakarta Messaging em sua saída.

Para configurar objetos IBM MQ classes for Jakarta Messaging a serem recuperados por meio de JNDI, o novo utilitário **JMS30Admin** é equivalente ao utilitário **JMSAdmin** para IBM MQ classes for JMS.

Observe que os objetos subjacentes são de pacotes diferentes. As definições JNDI criadas por **JMSAdmin** não podem ser usadas por IBM MQ classes for Jakarta Messaging, nem aquelas criadas por **JMS30Admin** não podem ser usadas por IBM MQ classes for JMS

Nota: Não há suporte para os objetos IBM MQ classes for Jakarta Messaging fornecidos pelo IBM MQ Explorer; sua integração JNDI é apenas para IBM MQ classes for JMS


Conceitos relacionados

[Por que devo usar as classes do IBM MQ para Jakarta Messaging?](#)

JMS e Jakarta Messaging modelo

O modelo JMS e Jakarta Messaging define um conjunto de interfaces que os aplicativos Java podem usar para executar operações do sistema de mensagens.. IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging são provedores de sistemas de mensagens que definem como os objetos de sistema de mensagens do Java estão relacionados aos conceitos do IBM MQ As especificações JMS e Jakarta Messaging esperam que determinados objetos do mensagens sejam objetos administrados.

A partir do IBM MQ 8.0, o produto suporta a versão JMS 2.0 do padrão JMS, que introduziu uma API simplificada, enquanto também retém a API clássica, do JMS 1.1.

 IBM MQ 9.3.0 introduz o suporte para [Jakarta Messaging 3.0](#). JMS 2.0 ainda é totalmente suportado. Como JMS e Jakarta Messaging compartilham muito em comum, referências adicionais a JMS neste tópico podem ser consideradas referentes a ambos. Quaisquer diferenças são destacadas conforme necessário.

API simplificada

O JMS 2.0 introduziu a API simplificada, enquanto também retém as interfaces específicas do domínio e independentes do domínio do JMS 1.1. A API simplificada reduz o número de objetos que são necessários para enviar e receber mensagens e consiste nas interfaces a seguir:

ConnectionFactory

Um ConnectionFactory é um objeto administrado usado por um cliente JMS para criar um Connection. Essa interface também é usada na API clássica.

JMSContexto

Esse objeto combina os objetos Connection e Session da API clássica. Os objetos JMSContext podem ser criados a partir de outros objetos JMSContext, com a conexão subjacente sendo duplicada.

JMSProdutor

Um JMSProducer é criado por um JMSContext e é usado para enviar mensagens para uma fila ou tópico. O objeto JMSProducer cria objetos que são necessários para enviar a mensagem.

Consumidor do JMS

Um JMSConsumer é criado por um JMSContext e é usado para receber mensagens de um tópico ou uma fila.

A API simplificada tem diversos efeitos:

- O objeto JMSContext sempre inicia automaticamente a conexão subjacente.
- JMSProducers e JMSConsumers agora podem trabalhar diretamente com corpos de mensagens, sem precisarem obter todo o objeto de mensagem, usando o método `getBody` da `Mensagem`.
- Propriedades de mensagens podem ser configuradas no objeto JMSProducer, usando encadeamento de método antes de enviar um 'body', um conteúdo de mensagem. O JMSProducer manipulará a criação de todos os objetos necessários para enviar a mensagem. Usando o JMS 2.0, as propriedades podem ser configuradas e uma mensagem pode ser enviada da seguinte forma:

```
context.createProducer().
setProperty("foo", "bar").
setTimeToLive(10000).
setDeliveryMode(NON_PERSISTENT).
setDisableMessageTimestamp(true).
send(dataQueue, body);
```

O JMS 2.0 também introduziu assinaturas compartilhadas em que as mensagens podem ser compartilhadas entre diversos consumidores Todas as assinaturas do JMS 1.1 são tratadas como assinaturas não compartilhadas.

API clássica

A lista a seguir resume as principais interfaces do JMS da API clássica:

Destino

Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

ConnectionFactory

Um objeto ConnectionFactory contém um conjunto de propriedades de configuração para uma conexão. Um aplicativo usa um connection factory para criar uma conexão.

Conexão

Um objeto Connection contém a conexão ativa de um aplicativo com um servidor de sistema de mensagens. Um aplicativo usa uma conexão para criar sessões.

Sessão

Uma sessão é um único contexto encadeado para enviar e receber mensagens. Um aplicativo usa uma sessão para criar mensagens, produtores de mensagens e consumidores de mensagens. Uma sessão é transacionada ou não transacionada.

Mensagem

Um objeto Message contém uma mensagem que um aplicativo envia ou recebe.

MessageProducer

Um aplicativo usa um produtor de mensagem para enviar mensagens para um destino.

MessageConsumer

Um aplicativo usa um consumidor de mensagem para receber mensagens enviadas a um destino.

Figura 53 na página 159 mostra esses objetos e seus relacionamentos.

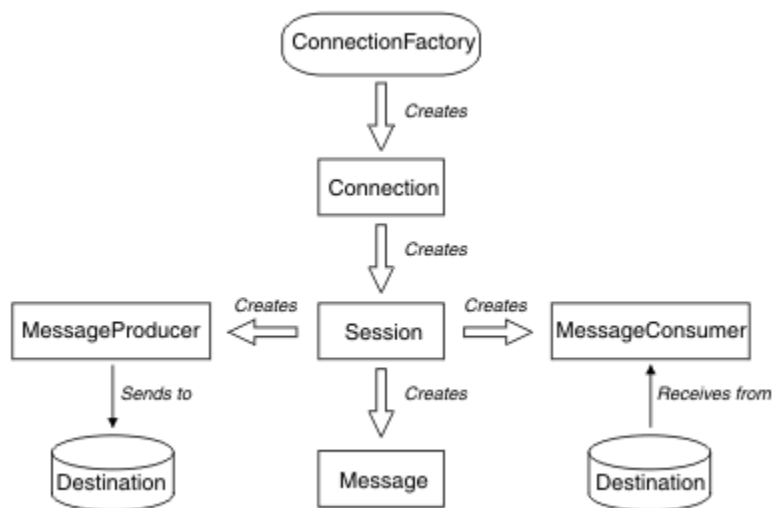


Figura 53. Os objetos do JMS e seus relacionamentos

O diagrama mostra as principais interfaces: ConnectionFactory, Connection, Session, MessageProducer, MessageConsumer, Message e Destination. Um aplicativo usa um connection factory para criar uma conexão e usa uma conexão para criar sessões. O aplicativo pode então usar uma sessão para criar mensagens, produtores de mensagens e consumidores de mensagens. O aplicativo usa um produtor de mensagem para enviar mensagens para um destino e usa um consumidor de mensagens para receber mensagens enviadas para um destino.

Um objeto Destination, ConnectionFactory ou Connection pode ser utilizado simultaneamente por diferentes encadeamentos de um aplicativo multiencadeado, mas um objeto Session, MessageProducer ou MessageConsumer não pode ser usado simultaneamente por diferentes encadeamentos. A maneira mais simples de garantir que um objeto Session, MessageProducer ou MessageConsumer não é usado simultaneamente é criar um objeto Session separado para cada encadeamento.

O JMS suporta dois estilos de sistema de mensagens:

- Sistema de mensagens ponto a ponto
- Sistema de Mensagens de Publicação/Assinatura

Esses estilos de sistema de mensagens também são referidos como *domínios do sistema de mensagens* e é possível combinar ambos os estilos do sistema de mensagens em um aplicativo. No domínio ponto a ponto, um destino é uma fila e, no domínio de publicar/assinar, um destino é um tópico.

Com versões do JMS anteriores ao JMS 1.1, a programação para o domínio ponto a ponto usa um conjunto de interfaces e métodos e a programação para o domínio de publicação/assinatura usa outro conjunto. Os dois conjuntos são semelhantes, mas separados. A partir do JMS 1.1, é possível usar um conjunto comum de interfaces e métodos que suportam ambos os domínios do sistema de mensagens. As interfaces comuns fornecem uma visualização independente de cada domínio de sistema de mensagens. [Tabela 17](#) na [página 160](#) lista as interfaces independentes de domínio do JMS e suas interfaces específicas de domínio correspondentes.

Tabela 17. O domínio do JMS independente e suas interfaces específicas de domínio correspondentes

Interfaces independentes de domínio	Interfaces específicas de domínio para o domínio ponto a ponto	Interfaces específicas de domínio para o domínio de publicar/assinar
ConnectionFactory	QueueConnectionFactory	TopicConnectionFactory
Conexão	QueueConnection	TopicConnection
Destino	Fila	Tópico
Sessão	QueueSession	TopicSession
MessageProducer	QueueSender	TopicPublisher
MessageConsumer	QueueReceiver QueueBrowser	TopicSubscriber

JMS 2.0 O IBM MQ classes for JMS 2.0 suporta ambas as interfaces específicas do domínio do JMS 1.1 anteriores e a API simplificada do JMS 2.0 IBM MQ classes for JMS 2.0 pode, portanto, ser usado para manter aplicativos existentes, incluindo o desenvolvimento de nova função em aplicativos existentes.

JM 3.0 O IBM MQ classes for Jakarta Messaging 3.0 suporta as versões do Jakarta Messaging das mesmas interfaces e é recomendado para o novo desenvolvimento de aplicativo

Em IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging, JMS objetos são relacionados a conceitos IBM MQ das seguintes maneiras:

- Um objeto Connection tem propriedades derivadas das propriedades do connection factory que foi usado para criar a conexão. Essas propriedades controlam como um aplicativo se conecta a um gerenciador de filas. Os exemplos dessas propriedades são o nome do gerenciador de filas e, para um aplicativo que se conecta ao gerenciador de filas no modo cliente, o nome do host ou o endereço IP do sistema em que o gerenciador de filas está em execução.
- Um objeto Session contém uma manipulação de conexões do IBM MQ que, portanto, define o escopo transacional da sessão.
- Cada objeto MessageProducer e cada objeto MessageConsumer contém uma manipulação de objetos do IBM MQ.

Ao usar IBM MQ classes for JMS ou IBM MQ classes for Jakarta Messaging, todas as regras normais de IBM MQ se aplicam.. Observe, especificamente, que um aplicativo pode enviar uma mensagem a uma fila remota, mas pode receber uma mensagem somente de uma fila de propriedade do gerenciador de filas ao qual o aplicativo está conectado.

A especificação do JMS espera que os objetos `ConnectionFactory` e `Destination` sejam objetos administrados. Um administrador cria e mantém objetos administrados em um repositório central e um aplicativo JMS recupera esses objetos usando a `Java Naming and Directory Interface (JNDI)`.


Em `IBM MQ classes for JMS` e `IBM MQ classes for Jakarta Messaging`, a implementação da interface de Destino é uma superclasse abstrata de Fila e Tópico e, portanto, uma instância de Destino é um objeto Fila ou um objeto Tópico. As interfaces independentes de domínio tratam uma fila ou um tópico como um destino. O domínio do sistema de mensagens para um objeto `MessageProducer` ou `MessageConsumer` é determinado por se o destino é uma fila ou um tópico.

Em `IBM MQ classes for JMS` e `IBM MQ classes for Jakarta Messaging`, portanto, os objetos dos seguintes tipos podem ser objetos administrados:

- `ConnectionFactory`
- `QueueConnectionFactory`
- `TopicConnectionFactory`
- Fila
- Tópico
- `XAConnectionFactory`
- `XAQueueConnectionFactory`
- `XATopicConnectionFactory`

IBM MQ classes for JMS/Jakarta Messaging arquitetura

`IBM MQ classes for JMS` e `IBM MQ classes for Jakarta Messaging` têm uma arquitetura em camadas. A camada mais superior do código é uma camada comum que qualquer provedor de sistemas de mensagens do IBM Java pode utilizar.

 IBM MQ 9.3.0 introduz o suporte para [Jakarta Messaging 3.0](#). JMS 2.0 ainda é totalmente suportado.

`IBM MQ classes for JMS` e `IBM MQ classes for Jakarta Messaging` possuem uma arquitetura em camadas, conforme mostrado no diagrama [Figura 54](#) na [página 162](#). A camada superior do código é uma camada comum que pode ser usada por qualquer provedor IBM JMS ou Jakarta Messaging. Quando um aplicativo chama um método JMS ou Jakarta Messaging, qualquer processamento da chamada que não seja específico para um sistema de mensagens é executado por uma camada comum, que também fornece uma resposta consistente para a chamada. Qualquer processamento da chamada que seja específico a um sistema de mensagens é delegado a uma camada inferior. No diagrama a seguir, o provedor de sistemas de mensagens do IBM MQ é mostrado na camada inferior, juntamente com dois provedores de sistema de mensagens adicionais (Provedor de sistemas de mensagens A e Provedor de sistemas de mensagens B).

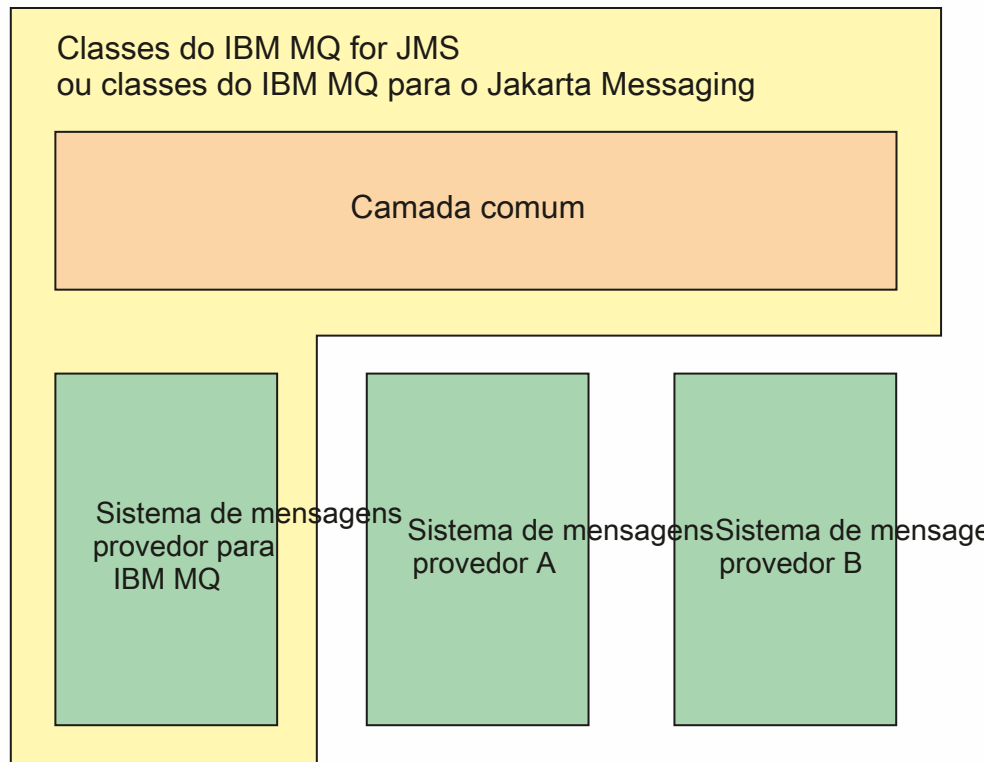


Figura 54. A arquitetura em camadas para os provedores IBM JMS e Jakarta Messaging

Uma arquitetura em camadas atinge os seguintes objetivos:

- Para melhorar a consistência do comportamento dos vários provedores IBM JMS e Jakarta Messaging
- Facilitar a gravação de um aplicativo de ligação entre dois sistemas de mensagens IBM
- Para facilitar a porta de um aplicativo de um provedor do IBM JMS ou Jakarta Messaging para outro

Tarefas relacionadas

[Usando classes do IBM MQ para o Sistema de Mensagens JMS/Jakarta](#)

Suporte para objetos administrados

IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging suportam o uso de objetos administrados.

V 9.3.0 **JM 3.0** **V 9.3.0** A partir de IBM MQ 9.3.0, Jakarta Messaging 3.0 é suportado para desenvolver novos aplicativos. O IBM MQ 9.3.0 continua a suportar JMS 2.0 para aplicativos existentes. Não é suportado usar a API Jakarta Messaging 3.0 e a API JMS 2.0 no mesmo aplicativo. Para obter mais informações, consulte [Usando IBM MQ classes para JMS/Jakarta Messaging](#).

O fluxo de lógica em um aplicativo JMS ou IBM MQ classes for Jakarta Messaging começa com ConnectionFactory e objetos de Destino. O aplicativo usa um objeto ConnectionFactory para criar um objeto Connection, que representa a conexão ativa do aplicativo com um servidor de sistema de mensagens. O aplicativo usa o objeto Connection para criar um objeto Session, que é um único contexto encadeado para produzir e consumir mensagens. O aplicativo pode então usar o objeto Session e um objeto Destination para criar um objeto MessageProducer, que o aplicativo usa para enviar mensagens para o destino especificado. O destino é uma fila ou um tópico no sistema de mensagens e é contido pelo objeto Destination. O aplicativo também pode usar o objeto Session e um objeto Destination para criar um objeto MessageConsumer, que o aplicativo usa para receber mensagens que foram enviadas para o destino especificado.

As especificações JMS e Jakarta Messaging esperam que ConnectionFactory e objetos de Destino sejam objetos administrados. Um administrador cria e mantém objetos administrados em um repositório central e um aplicativo JMS ou Jakarta Messaging recupera esses objetos usando Java Naming Directory Interface (JNDI). O repositório de objetos administrados pode variar de um arquivo simples para um diretório LDAP (Lightweight Directory Access Protocol)..

IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging suportam o uso de objetos administrados. Um aplicativo pode usar todos os recursos do IBM MQ classes for JMS ou IBM MQ classes for Jakarta Messaging que são expostos por meio do IBM MQ sem ter nenhuma informação específica do IBM MQ codificada permanentemente no próprio aplicativo. Esta organização fornece ao aplicativo um grau de independência da configuração do IBM MQ subjacente.

Para alcançar essa independência, o aplicativo pode usar JNDI para recuperar connection factories e destinos armazenados como objetos administrados e usar apenas as interfaces definidas no pacote `javax.jms` (JMS 2.0) ou `jakarta.jms` (Jakarta Messaging 3.0) para executar operações do sistema de mensagens.

JMS 2.0 Para JMS 2.0, um administrador pode usar a IBM MQ JMS ferramenta de administração **JMSAdmin** ou IBM MQ Explorer para criar e manter objetos administrados em um repositório central.

JM 3.0 Para Jakarta Messaging 3.0, não é possível administrar o JNDI usando IBM MQ Explorer. A administração JNDI é suportada pela variante Jakarta Messaging 3.0 de **JMSAdmin**, que é **JMS30Admin**.

Um servidor de aplicativos geralmente fornece seu próprio repositório para objetos administrados e suas ferramentas para criar e manter os objetos. Um aplicativo Java EE **JM 3.0** ou Jakarta EE pode, portanto, usar JNDI para recuperar objetos administrados do repositório do servidor de aplicativos ou de um repositório central.

Tarefas relacionadas

[Configurando recursos do JMS e do Jakarta Messaging](#)

Tipos de comunicação suportados nas plataformas Java EE e Jakarta EE

Nas plataformas Java EE e Jakarta EE, IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging suportam dois tipos de comunicação entre um componente de um aplicativo e um gerenciador de filas do IBM MQ.

V 9.3.0 **JM 3.0** **V 9.3.0** IBM MQ 9.3.0 introduz o suporte para Jakarta Messaging 3.0. JMS 2.0 ainda é totalmente suportado. Como JMS e Jakarta Messaging compartilham muito em comum, referências adicionais a JMS neste tópico podem ser consideradas referentes a ambos. Quaisquer diferenças são destacadas conforme necessário.

Os dois tipos de comunicação a seguir entre um componente de um aplicativo e um gerenciador de filas do IBM MQ são suportados:

- Comunicação de saída
- Comunicação de entrada

Comunicação de saída

Usando a API JMS ou Jakarta Messaging diretamente, um componente de aplicativo cria uma conexão com um gerenciador de filas e, em seguida, envia e recebe mensagens..

Por exemplo, o componente de aplicativo pode ser um aplicativo cliente, um servlet, um JavaServer Page (JSP), um enterprise Java bean (EJB), ou um bean acionado por mensagens (MDB). Neste tipo de comunicação, o contêiner do servidor de aplicativos fornece somente funções de baixo nível para suportar operações do sistema de mensagens, como definição do conjunto de conexões e gerenciamento de encadeamentos.

Comunicação de entrada

No caso da comunicação de entrada, uma mensagem que chega a um destino é entregue a um MDB que, em seguida, processa a mensagem.

Os aplicativos Java EE **JM 3.0** e Jakarta EE usam MDBs para processar mensagens assincronamente. Um MDB age como um listener de mensagem do JMS e é implementado por um método `onMessage()`, que define como uma mensagem é processada. Um MDB está implementado no contêiner EJB de um servidor de aplicativos. A maneira precisa na qual um MDB é configurado depende de qual servidor de aplicativos que está sendo usado, mas as informações de configuração devem especificar a qual gerenciador de filas conectar-se, como conectar-se ao gerenciador de filas, qual destino monitorar para mensagens e o comportamento transacional do MDB. Estas informações são então usadas pelo contêiner EJB. Quando uma mensagem satisfazendo os critérios de seleção do MDB chega ao destino especificado, o contêiner EJB usa IBM MQ classes for JMS ou IBM MQ classes for Jakarta Messaging para recuperar a mensagem do gerenciador de filas e, em seguida, entrega a mensagem ao MDB chamando seu método `onMessage()`.

Relacionamento com o IBM MQ classes for Java

IBM MQ classes for Java, IBM MQ classes for Jakarta Messaging e IBM MQ classes for JMS são peers que usam uma interface comum do Java para o MQI

A [Figura 55 na página 164](#) mostra o relacionamento entre IBM MQ classes for JMS, IBM MQ classes for Jakarta Messaging e IBM MQ classes for Java

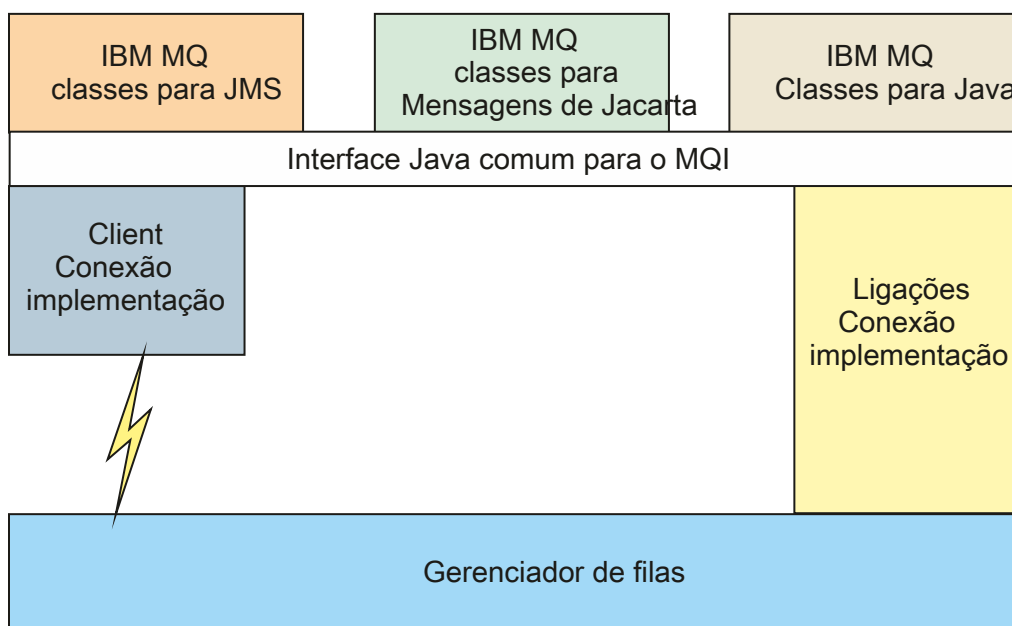







Figura 55. O relacionamento entre IBM MQ classes for JMS, IBM MQ classes for Jakarta Messaging e IBM MQ classes for Java

Em geral, os programas Java devem usar apenas uma interface para fazer a interface com IBM MQ - IBM MQ classes for Java, IBM MQ classes for Jakarta Messaging ou IBM MQ classes for JMS Interfaces de combinação não são suportadas, com uma exceção. Para manter a compatibilidade com liberações anteriores do IBM WebSphere MQ 7.0, as classes de saída do canal que são gravadas no Java ainda podem usar as interfaces do IBM MQ classes for Java, mesmo se as classes de saída do canal sejam chamadas por meio do IBM MQ classes for JMS. No entanto, usar as interfaces IBM MQ classes for Java significa que seus aplicativos ainda são dependentes de:

- **JMS 2.0** O IBM MQ classes for Java arquivo JAR com `.ibm.mq.jar`. Se você não quiser `com.ibm.mq.jar` em seu caminho de classe, será possível usar o conjunto de interfaces no pacote `com.ibm.mq.exits`.

-    Uso do `com.ibm.mq.jakarta.client.jar`, ao interoperar com IBM MQ classes for Jakarta Messaging.

Conceitos relacionados

  [Por que devo usar as classes do IBM MQ para Jakarta Messaging?](#)

[Por que devo usar as classes IBM MQ para JMS?](#)

[Por que devo usar classes do IBM MQ para Java?](#)

Provedor de sistema de mensagens do IBM MQ

O provedor de mensagens do IBM MQ possui três modos de operação: modo normal, o modo normal com restrições e modo de migração.

O provedor de sistemas de mensagens IBM MQ tem três modos de operação:

- Modo normal do provedor de mensagens IBM MQ
- Modo normal com restrições do provedor de mensagens IBM MQ
- Modo de migração do provedor de mensagens IBM MQ

O modo normal do provedor de mensagens IBM MQ usa todos os recursos de um gerenciador de filas IBM MQ para implementar JMS. Esse modo é otimizado para usar a API e a funcionalidade do JMS

2.0    ou [Jakarta Messaging 3.0](#)

Se:

- O cliente especifica uma versão do provedor de 6 em um **ConnectionFactory**, o cliente se comporta de maneira compatível com o cliente fornecido com IBM WebSphere MQ 6.0. Apenas interfaces JMS 1.1 e JMS 2 são suportadas, mas algumas funcionalidades JMS 2, como assinaturas compartilhadas, atraso de entrega e envio assíncrono, estão desativadas. Não há compartilhamento de conexão..
- O cliente especifica uma versão do provedor de 7 em um **ConnectionFactory**, as interfaces JMS 1.1 e JMS 2 são totalmente suportadas.
- Nenhuma versão do provedor foi especificada, foi feita uma tentativa de conexão com o provedor versão 7. Se isso falhar, uma nova tentativa será feita com o provedor versão 6.

Se você deseja conectar-se ao IBM Integration Bus usando o IBM MQ Enterprise Transport, use o modo de migração. Se você usar o IBM MQ Real-Time Transport, o modo de migração será selecionado automaticamente, porque você selecionou explicitamente as propriedades no objeto de connection factory. A conexão com o IBM Integration Bus usando o IBM MQ Enterprise Transport segue as regras gerais para seleção de modo descritas em [Configurando a propriedade JMS PROVIDERVERSION](#).

Tarefas relacionadas

[Configurando recursos do JMS](#)



Conceitos do IBM MQ for z/OS

Alguns dos conceitos usados pelo IBM MQ for z/OS são exclusivos para a plataforma z/OS. Por exemplo, o mecanismo de criação de log, as técnicas de gerenciamento de armazenamento, a disposição da unidade de recuperação e os grupos de filas compartilhadas são fornecidos apenas com o IBM MQ for z/OS. Use este tópico como uma introdução para obter informações adicionais a respeito desses conceitos

Os conceitos incluem uma visão geral dos objetos que o IBM MQ for z/OS usa, incluindo:

- O gerenciador de filas
- O Inicializador de Canais
- Filas compartilhadas e grupos de filas compartilhadas
- Fila do intra-grupo

Os tópicos a seguir também abrangem vários procedimentos necessários, incluindo:

- Definições do sistema em z/OS ..
- Gerenciamento de Armazenamento
- Recuperação e Reinicialização
- Conceitos de segurança no IBM MQ for z/OS

Conceitos relacionados

“O gerenciador de filas no z/OS” na página 167

Antes de deixar que seus programas de aplicativo usem o IBM MQ em seu sistema z/OS, você deve instalar o produto IBM MQ for z/OS e iniciar um gerenciador de filas. O gerenciador de filas possui e gerencia o conjunto de recursos que são usados pelo IBM MQ.

“O inicializador de canais no z/OS” na página 168

O inicializador de canais fornece e gerencia os recursos que ativam o enfileiramento distribuído do IBM MQ. IBM MQ usa o *Message Channel Agents* (MCAs) para enviar mensagens de um gerenciador de filas a outro.

“Termos e tarefas para gerenciar o IBM MQ for z/OS” na página 169

Use este tópico como uma introdução à terminologia e tarefas que são específicas para IBM MQ for z/OS.

“Filas compartilhadas e grupos de filas compartilhadas” na página 172

É possível usar filas compartilhadas e grupos de filas compartilhadas, para implementar a alta disponibilidade de recursos do IBM MQ. As filas compartilhadas e os grupos de filas compartilhadas são funções exclusivas para o IBM MQ for z/OS na plataforma z/OS.

“Fila do intra-grupo” na página 220

Esta seção descreve o enfileiramento intragrupo, uma função exclusiva do IBM MQ for z/OS para a plataforma z/OS. Essa função está disponível apenas para gerenciadores de filas definidos para um grupo de filas compartilhadas.

“Gerenciamento de armazenamento no z/OS” na página 234

IBM MQ for z/OS requer estruturas de dados temporários e permanentes e usa os conjuntos de páginas e os buffers de memória para armazenar esses dados. Estes tópicos fornecem detalhes adicionais sobre como o IBM MQ usa os conjuntos de páginas e os buffers.

“Efetuando login no IBM MQ for z/OS” na página 239

IBM MQ mantém os *logs* das mudanças de dados e eventos significativos conforme eles ocorrem. Esses logs podem ser usados para recuperar os dados para um estado anterior, se necessário.

“Recuperação e reinicialização no z/OS” na página 261

Use os links neste tópico para saber sobre os recursos do IBM MQ for z/OS para reinicialização e recuperação.

“Conceitos de segurança no IBM MQ for z/OS” na página 280

Use este tópico para entender a importância da segurança para o IBM MQ, e as implicações de não ter configurações de segurança adequadas em seu sistema.

“Disponibilidade no z/OS” na página 286

IBM MQ for z/OS possui muitos recursos para alta disponibilidade. Este tópico descreve algumas das considerações para disponibilidade.

“Disposição de unidade de recuperação no z/OS” na página 291

Determinados aplicativos transacionais podem usar uma disposição da unidade de recuperação GROUP, em vez de um QMGR, quando conectado a um gerenciador de filas em um grupo de filas compartilhadas (QSG), especificando o nome do QSG ao invés do nome do gerenciador de filas ao se conectarem. Isso permite que a recuperação de transação seja mais flexível e robusta, removendo o requisito para se reconectar ao mesmo gerenciador de filas no QSG.

Referências relacionadas

“Definição do sistema em z/OS” na página 250

IBM MQ for z/OS usa muitas definições de objeto padrão, e fornece JCL de amostra para criar esses objetos padrão. Use este tópico para entender esses objetos padrão e o JCL de amostra.

“Monitoramento e estatísticas no IBM MQ for z/OS” na página 290

O IBM MQ for z/OS tem um conjunto de recursos para monitorar o gerenciador de filas e reunir estatísticas.

z/OS O gerenciador de filas no z/OS

Antes de deixar que seus programas de aplicativo usem o IBM MQ em seu sistema z/OS, você deve instalar o produto IBM MQ for z/OS e iniciar um gerenciador de filas. O gerenciador de filas possui e gerencia o conjunto de recursos que são usados pelo IBM MQ.

O gerenciador de filas

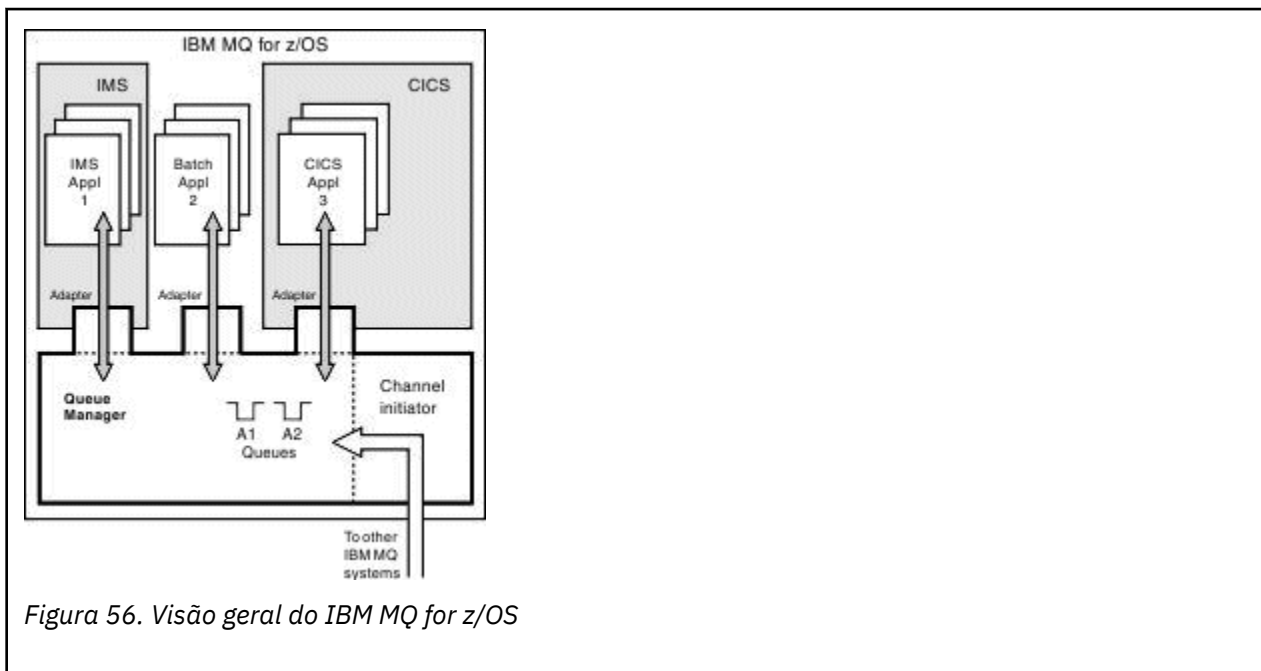
Um *gerenciador de filas* é um programa que fornece serviços do sistema de mensagens para os aplicativos. Os aplicativos que usam o Message Queue Interface (MQI) podem colocar as mensagens nas filas e obter as mensagens das filas. O gerenciador de filas assegura que as mensagens sejam enviadas para a fila correta ou sejam roteadas para outro gerenciador de filas. O gerenciador de filas processa as chamadas MQI que são emitidas e os comandos que são enviados para ele (independente da origem). O gerenciador de filas gera códigos de conclusão apropriados para cada chamada ou comando.

Os recursos gerenciados pelo gerenciador de filas incluem:

- Conjuntos de páginas que retêm as definições de objeto do IBM MQ e a mensagem de dados
- Logs que são usados para recuperar as mensagens e os objetos no caso de falha no gerenciador de filas
- Armazenamento de Processador
- Conexões por meio das quais diferentes ambientes de aplicativo (CICS, IMS e Batch) podem acessar a API do IBM MQ
- O inicializador de canais do IBM MQ, que permite a comunicação entre o IBM MQ em seu sistema z/OS e outros sistemas

O gerenciador de filas tem um nome e os aplicativos podem se conectar a ele usando esse nome.

Figura 56 na página 167 ilustra um gerenciador de filas, mostrando as conexões aos diferentes ambientes de aplicativo e inicializador de canais.



O subsistema do gerenciador de filas no z/OS

No z/OS, IBM MQ é executado como um subsistema z/OS que é iniciado no tempo do IPL. No subsistema, o gerenciador de filas é iniciado executando um procedimento JCL que especifica os conjuntos de

dados do z/OS que contêm informações sobre os logs e que retêm as definições de objeto e dados de mensagem (conjuntos de páginas). O subsistema e o gerenciador de filas possuem o mesmo nome, de até quatro caracteres. Todos os gerenciadores de fila na sua rede devem ter nomes exclusivos, mesmo se estiverem em diferentes sistemas, sysplexes ou plataformas.

z/OS O inicializador de canais no z/OS

O inicializador de canais fornece e gerencia os recursos que ativam o enfileiramento distribuído do IBM MQ. IBM MQ usa o *Message Channel Agents* (MCAs) para enviar mensagens de um gerenciador de filas a outro.

Para enviar mensagens do gerenciador de filas A para o gerenciador de filas B, um MCA de *envio* no gerenciador de filas A deve configurar um link de comunicações com o gerenciador de filas B. Um MCA de *recebimento* deve ser iniciado no gerenciador de filas B para receber mensagens a partir do link de comunicações. Este caminho unidirecional que consiste no MCA de envio, no link de comunicações e no MCA de recebimento é conhecido como um *canal*. O MCA de envio recebe as mensagens de uma fila de transmissão e as envia para um canal para o MCA de recebimento. O MCA de recebimento recebe as mensagens e as coloca nas filas de destino.

No IBM MQ for z/OS, o envio e recebimento MCAs todos executados dentro do inicializador de canais (o inicializador de canais também é conhecido como *mover*). O inicializador de canais é executado como um espaço de endereço do z/OS sob o controle do gerenciador de filas. Pode haver apenas um único inicializador de canais conectado a um gerenciador de filas e ele é executado dentro da mesma imagem do z/OS como o gerenciador de filas. Pode haver milhares de processos de MCA sendo executados dentro do inicializador de canais simultaneamente.

Figura 57 na página 168 mostra dois gerenciadores de filas em um sysplex. Cada gerenciador de filas tem um inicializador de canais e uma fila local. Mensagens enviadas pelos gerenciadores de filas no AIX e Windows são colocadas na fila local, a partir de onde elas são recuperadas por um aplicativo. As mensagens de resposta são retornadas por uma rota semelhante.

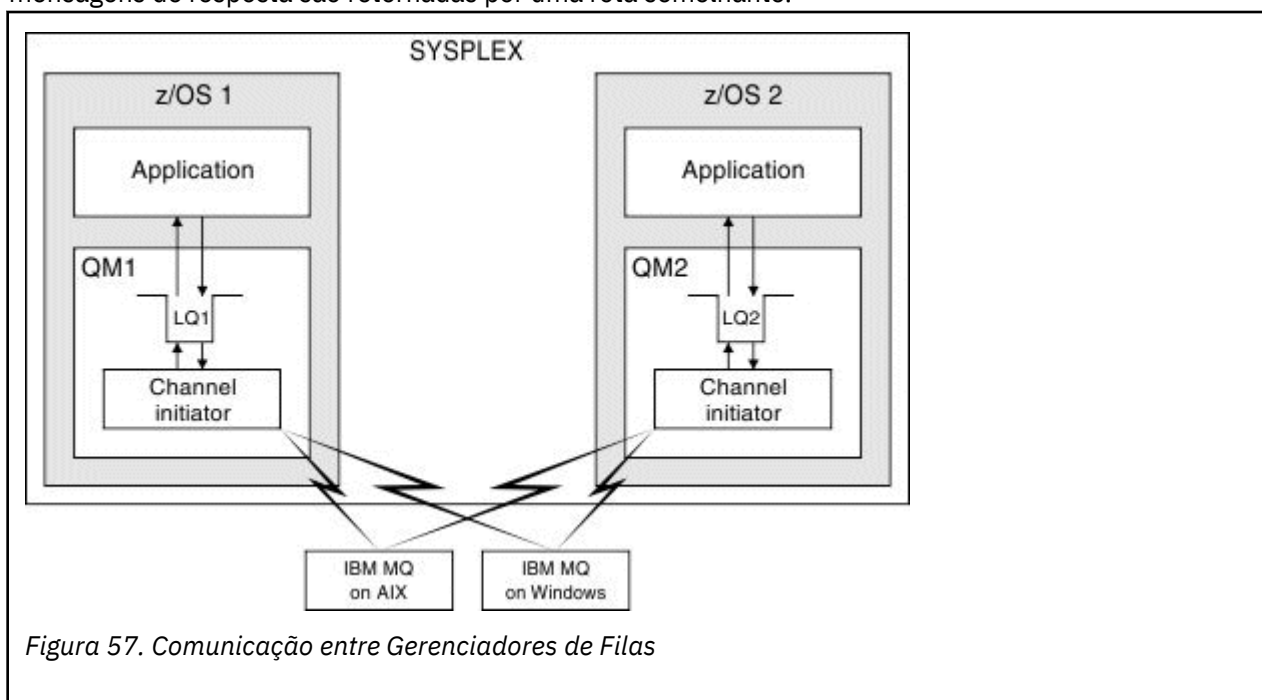


Figura 57. Comunicação entre Gerenciadores de Filas

O inicializador de canais também contém outros processos relativos ao gerenciamento dos canais. Esses processos incluem:

Listeners

Esses processos atendem solicitações de canal de entrada em um subsistema de comunicações como TCP e iniciam um MCA nomeado quando uma solicitação de entrada for recebida.

Supervisor

Isso gerencia o espaço de endereço do inicializador de canais, por exemplo, é responsável por reiniciar os canais após uma falha.

Servidor de nomes

Isso é usado para resolver os nomes TCP nos endereços.

Tarefas do TLS

Elas são usadas para executar a criptografia e a decriptografia e verificar as listas de revogação de certificado.

z/OS Registros SMF para o inicializador de canais

O inicializador de canais (CHINIT) pode produzir registros das estatísticas de SMF e registros de contabilidade com informações sobre tarefas e canais.

O CHINIT pode produzir registros das estatísticas de SMF e registros de contabilidade com os seguintes tipos de informações:

- As tarefas: dispatcher, adaptador, Domain Name Server (DNS) e SSL. Essas tarefas formam o que é chamado de estatísticas CHINIT.
- Canais: fornece informações sobre contabilidade semelhantes às aquelas disponíveis com o comando DISCHSTATUS. Isso é chamado de contabilidade do canal.

IBM MQ for Multiplatforms fornece informações semelhantes gravando mensagens PCF no SYSTEM.ADMIN.STATISTICS.QUEUE. Veja [Dados da mensagem de estatísticas do canal](#) para obter informações adicionais sobre como as informações de estatísticas são registradas no IBM MQ for Multiplatforms.

Dados estatísticos

É possível usar essas informações para descobrir as seguintes informações:

- Se você precisa mais das tarefas CHINIT, tal como o número de TCBs SSL e quanta CPU é usada por estas tarefas.
- O tempo médio para solicitações nessas tarefas.
- A solicitação de duração mais longa no intervalo, e a hora do dia dessa ocorrência, para tarefas de DNS e SSL. É possível correlacionar essa hora do dia com problemas que podem ocorrer com o canal.

Dados de Contabilidade

É possível usar estas informações para monitorar o uso do canal e descobrir as seguintes informações:

- Os canais com o rendimento mais alto.
- A taxa em que as mensagens foram enviadas, e a taxa de envio de dados em MB/segundo.
- O tamanho do lote alcançado. Se o tamanho do lote alcançado está próximo ao tamanho do lote especificado para o canal, o canal pode estar perto de seu limite para enviar mensagens.

É possível usar os comandos [START TRACE](#) e [STOP TRACE](#) para controlar a coleta do rastreamento de contabilidade e do rastreamento de estatísticas. Você pode utilizar as opções [STATCHL](#) e [STATACLS](#) no canal e gerenciador de filas para controlar se os canais produzem dados SMF.

z/OS Termos e tarefas para gerenciar o IBM MQ for z/OS

Use este tópico como uma introdução à terminologia e tarefas que são específicas para IBM MQ for z/OS.

Alguns dos termos e tarefas necessários para gerenciar IBM MQ for z/OS são específicos para a plataforma z/OS. A lista a seguir contém alguns desses termos e tarefas.

- [Filas compartilhadas](#)
- [Conjuntos de páginas e buffer pools](#)

- [Criação de log](#)
- [Customizando o ambiente do gerenciador de filas](#)
- [Reinicialização e recuperação](#)
- [Segurança](#)
- [Disponibilidade](#)
- [Manipulando objetos](#)
- [Monitoramento e estatísticas](#)
- [Ambientes de aplicativos](#)

Filas compartilhadas

As filas podem ser *não compartilhadas*, pertencentes e acessíveis a apenas um gerenciador de filas, ou *compartilhadas*, pertencentes a um *grupo de filas compartilhadas*. Um grupo de filas compartilhadas consiste em vários gerenciadores de filas em execução em um único sysplex do z/OS, que podem acessar as mesmas definições de objeto e dados da mensagem do IBM MQ simultaneamente. Dentro de um grupo de filas compartilhadas, as definições de objeto compartilhável são armazenadas em um banco de dados do Db2 compartilhado. As mensagens de fila compartilhada são retidas dentro de uma ou mais estruturas do recurso de acoplamento (estruturas CF). Se os dados da mensagem forem muito grandes para armazenar diretamente na estrutura (mais de 63 KB de tamanho) ou se a mensagem for grande o suficiente para que regras definidas pela instalação a selecionem para transferência, as informações de controle da mensagem ainda serão armazenadas na entrada do recurso de acoplamento, mas os dados da mensagem serão transferidos para um conjunto de dados de mensagem compartilhada (SMDS) ou para um banco de dados compartilhado do Db2. Os conjuntos de dados de mensagem compartilhada, o banco de dados Db2 compartilhado e as estruturas do recurso de acoplamento são recursos que são gerenciados em conjunto por todos os gerenciadores de filas no grupo.

Conjuntos de Páginas e Buffer Pools

Quando uma mensagem é colocada em uma fila não compartilhada, o gerenciador de filas armazena os dados em um conjunto de páginas de maneira que possa ser recuperada quando uma operação subsequente obtiver uma mensagem da mesma fila. Se a mensagem for removida da fila, o espaço no conjunto de páginas que retém os dados será liberado posteriormente para reutilização. Conforme o número de mensagens retidas em uma fila aumenta, também aumenta a quantia de espaço usado no conjunto de páginas e conforme o número de mensagens em uma fila diminui, também diminui o espaço usado no conjunto de páginas.

Para reduzir o custo de desempenho da gravação de dados e leitura de dados dos conjuntos de páginas, o gerenciador de filas armazena em buffer as atualizações no armazenamento de processador. A quantia de armazenamento usada para armazenar em buffer o acesso do conjunto de páginas é controlada por meio dos objetos IBM MQ denominados *buffer pools*.

Para obter informações adicionais sobre os conjuntos de páginas e os buffer pools, consulte o [Gerenciamento de Armazenamento](#).

Criação de Log

Quaisquer mudanças nos objetos retidos nos conjuntos de páginas e operações nas mensagens persistentes são registradas como registros de log. Esses registros de log são gravados em um conjunto de dados do log denominado *log ativo*. O nome e o tamanho do conjunto de dados de log ativo são retidos em um conjunto de dados denominado *conjunto de dados de autoinicialização* (BSDS).

Quando o conjunto de dados de log ativo é preenchido, o gerenciador de filas alterna para outro conjunto de dados de log para que a criação de log possa continuar e copia o conteúdo do conjunto de dados

de log ativo para um conjunto de dados do *log de archive*. As informações sobre essas ações, incluindo o nome do conjunto de dados do log de archive, são retidas no conjunto de dados de autoinicialização. Conceitualmente, existe um anel de conjuntos de dados de log ativo que o gerenciador de filas percorre; quando um log ativo é preenchido, os dados do log são transferidos para um log de archive e o conjunto de dados de log ativo fica disponível para reutilização.

Para obter informações adicionais sobre os conjuntos de dados de autoinicialização e de log, consulte [“Efetuando login no IBM MQ for z/OS”](#) na página 239.

Customizando o Ambiente do Gerenciador de Filas

Quando o gerenciador de filas é iniciado, um conjunto de parâmetros de inicialização que controlam como o gerenciador de filas é operado é lido. Além disso, os conjuntos de dados que contêm os comandos do IBM MQ são lidos e os comandos que eles contêm são executados. Geralmente, esses conjuntos de dados contêm definições dos objetos do sistema necessárias para que o IBM MQ seja executado e você pode padronizá-los para definir ou inicializar os objetos do IBM MQ necessários para o seu ambiente operacional. Quando esses conjuntos de dados tiverem sido lidos, quaisquer objetos definidos por eles serão armazenados em um conjunto de páginas ou no Db2.

Para obter informações adicionais sobre os parâmetros de inicialização e os objetos do sistema, consulte [“Definição do sistema em z/OS”](#) na página 250.

Recuperação e Reinicialização

A qualquer momento durante a operação do IBM MQ, pode haver mudanças retidas no armazenamento de processador que ainda não tenham sido gravadas no conjunto de páginas. Essas mudanças são gravadas no conjunto de páginas que é menos usado recentemente por uma tarefa em segundo plano no gerenciador de filas.

Se o gerenciador de filas for finalizado de maneira anormal, a fase de recuperação de reinicialização do gerenciador de filas pode recuperar as mudanças do conjunto de páginas perdidas porque os dados de mensagem persistentes são retidos nos registros de log. Isso significa que o IBM MQ pode recuperar os dados de mensagem persistentes e as mudanças de objeto direto no ponto de falha.

Se um gerenciador de filas que é um membro de um grupo de filas compartilhadas encontrar uma falha do recurso de acoplamento, as mensagens persistentes nessa fila poderão ser recuperadas apenas se você tiver feito backup da estrutura do recurso de acoplamento.

Para obter informações adicionais sobre a recuperação e a reinicialização, consulte [“Recuperação e reinicialização no z/OS”](#) na página 261.

Segurança

É possível usar um gerenciador de segurança externa, como Servidor de segurança (anteriormente conhecido como RACF) para proteger os recursos que o IBM MQ possui e gerenciar o acesso dos usuários não autorizados. Também é possível usar a Segurança da Camada de Transporte (TLS) para segurança do canal. O TLS é incluído como parte do produto IBM MQ.

Para obter mais informações sobre a segurança do IBM MQ, consulte [“Conceitos de segurança no IBM MQ for z/OS”](#) na página 280.

Disponibilidade

Existem diversos recursos do IBM MQ que são projetados para aumentar a disponibilidade do sistema no caso de falha no subsistema de comunicações ou do gerenciador de filas. Para obter informações adicionais sobre esses recursos, consulte [“Disponibilidade no z/OS”](#) na página 286.

Manipulando Objetos

Quando o gerenciador de filas estiver em execução, será possível manipular objetos do IBM MQ por meio de uma interface de console do z/OS ou por meio de um utilitário de administração que usa os serviços ISPF em TSO. Ambos os mecanismos permitem a você definir, alterar ou excluir os objetos do IBM MQ. Também é possível controlar e exibir o status das várias funções do IBM MQ e do gerenciador de filas.

Para obter mais informações sobre esses recursos, consulte [Origens a partir das quais é possível emitir comandos MQSC e PCF no IBM MQ for z/OS](#)

Também é possível manipular os objetos do IBM MQ usando o IBM MQ Explorer, uma interface gráfica com o usuário que fornece uma maneira visual de trabalhar com filas, gerenciadores de filas e outros objetos.

>Monitoramento e Estatísticas

Diversos recursos estão disponíveis para monitorar os seus gerenciadores de fila e os inicializadores de canais. Também é possível coletar as estatísticas para avaliação de desempenho e propósitos de contabilidade.

Para obter informações adicionais sobre esses recursos, consulte [“Monitoramento e estatísticas no IBM MQ for z/OS”](#) na página 290.

Ambientes de Aplicativos

Quando o gerenciador de filas tiver sido iniciado, os aplicativos podem se conectar a ele e iniciar usando a API do IBM MQ. Eles podem ser CICS, IMS, Batch, ou aplicativos do WebSphere Application Server. Os aplicativos IBM MQ também podem acessar os aplicativos nos sistemas CICS e IMS que não estão cientes do IBM MQ, usando o CICS e pontes do IMS.

Para obter informações adicionais sobre esses recursos, consulte [“IBM MQ e outros Produtos z/OS”](#) na página 294.

Para obter informações sobre a gravação de aplicativos IBM MQ, consulte a seguinte documentação:

- [Desenvolvendo Aplicativos](#)
- [Using C++](#)
- [Usando o IBM MQ classes for Java](#)

z/OS

Filas compartilhadas e grupos de filas compartilhadas

É possível usar filas compartilhadas e grupos de filas compartilhadas, para implementar a alta disponibilidade de recursos do IBM MQ. As filas compartilhadas e os grupos de filas compartilhadas são funções exclusivas para o IBM MQ for z/OS na plataforma z/OS.

Esta seção descreve os atributos e os benefícios, e oferece informações sobre como os diversos gerenciadores de filas podem compartilhar as mesmas filas e as mensagens nessas filas.

O que é uma fila compartilhada?

Uma fila compartilhada é um tipo de fila local. As mensagens nessa fila podem ser acessadas por um ou mais gerenciadores de filas que estão em um sysplex.

Um grupo de filas compartilhadas

Os gerenciadores de filas que podem acessar o mesmo conjunto de filas compartilhadas formam um grupo denominado *grupo de filas compartilhadas*.

Qualquer Gerenciador de Filas Pode Acessar as Mensagens

Qualquer gerenciador de filas no grupo de filas compartilhadas pode acessar uma fila compartilhada. Isso significa que é possível colocar uma mensagem em uma fila compartilhada em um gerenciador de filas e obter a mesma mensagem da fila a partir de um gerenciador de filas diferente. Isso fornece um mecanismo rápido para comunicação dentro de um grupo de filas compartilhadas que não requer que os canais estejam ativos entre os gerenciadores de filas.

IBM MQ suporta a transferência de mensagens para Db2 ou um conjunto de dados de mensagem compartilhada (SMDS). A transferência de mensagens de qualquer tamanho é configurável.

A [Figura 58 na página 173](#) mostra três gerenciadores de filas e um recurso de acoplamento, formando um grupo de filas compartilhadas. Todos os três gerenciadores de filas podem acessar a fila compartilhada no recurso de acoplamento.

Um aplicativo pode se conectar a qualquer um dos gerenciadores de filas dentro do grupo de filas compartilhadas. Como todos os gerenciadores de filas no grupo de filas compartilhadas podem acessar todas as filas compartilhadas, o aplicativo não depende da disponibilidade de um gerenciador de filas específico. Qualquer gerenciador de filas no grupo de filas compartilhadas pode atender à fila.

Isso fornece uma maior disponibilidade porque todos os outros gerenciadores de filas no grupo de filas compartilhadas poderão continuar processando a fila se um dos gerenciadores de filas tiver um problema.

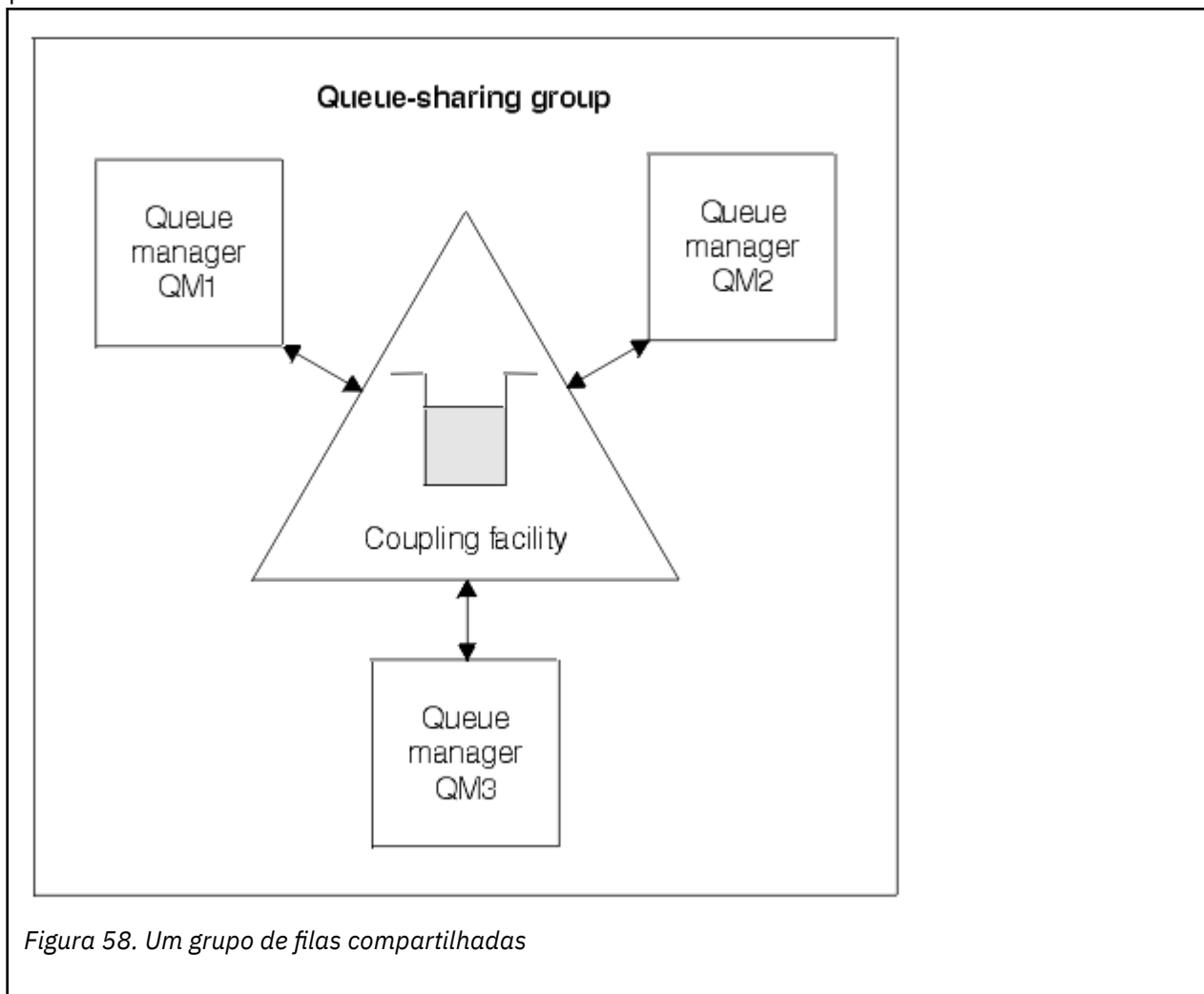


Figura 58. Um grupo de filas compartilhadas

Definição de Filas Compartilhada por Todos os Gerenciadores de Filas

Definições de fila compartilhada são armazenadas na tabela de banco de dados OBJ_B_QUEUE do Db2. Por causa disso, é necessário definir a fila apenas uma vez e, em seguida, ela pode ser acessada por todos

os gerenciadores de filas no grupo de filas compartilhadas. Isso significa que existem menos definições a serem feitas.

Por outro lado, a definição de uma fila não compartilhada é armazenada no conjunto de páginas zero do gerenciador de filas que possui a fila (conforme descrito nos [Conjuntos de Páginas](#)).

Não será possível definir uma fila compartilhada, se uma fila com esse nome já tiver sido definida nos conjuntos de páginas do gerenciador de filas de definição. Da mesma maneira, não é possível definir uma versão local de uma fila nos conjuntos de páginas do gerenciador de filas, se existir uma fila compartilhada com o mesmo nome.

O que é um grupo de filas compartilhadas?

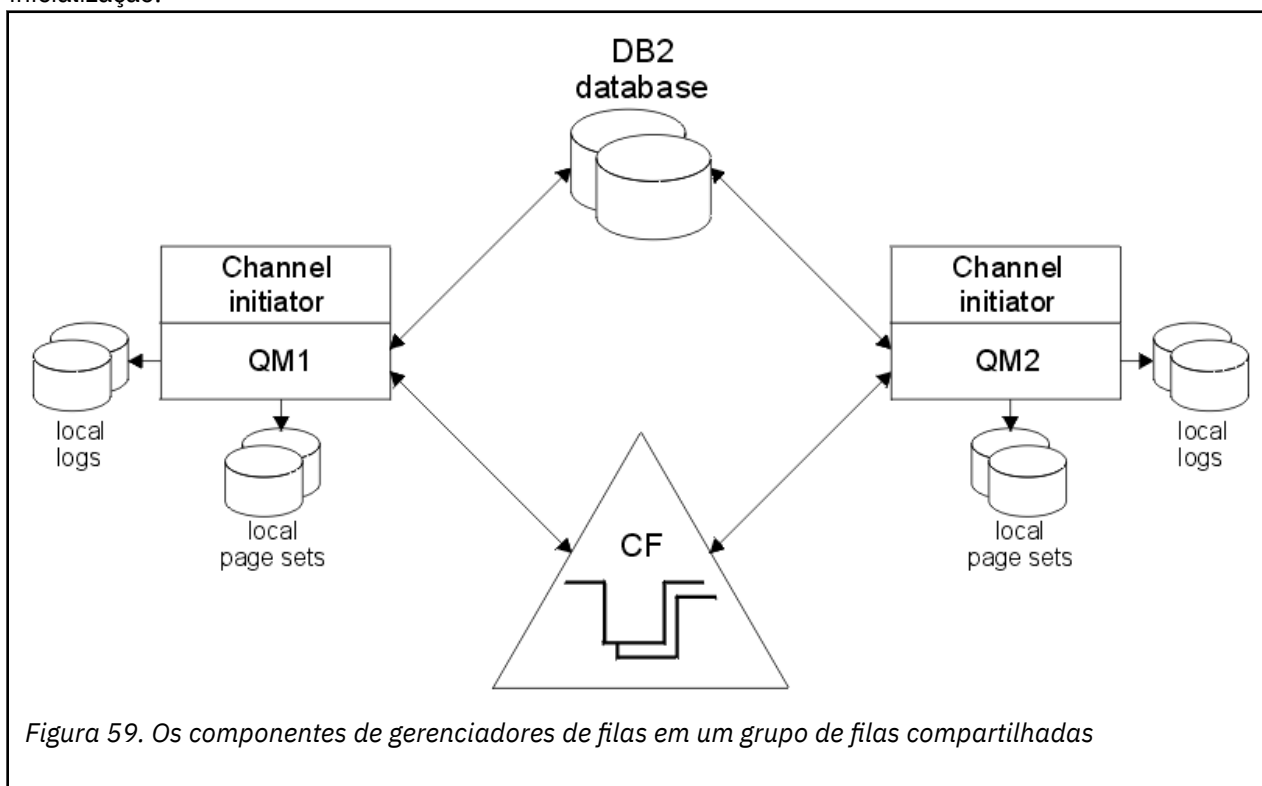
Um grupo de gerenciadores de filas que pode acessar as mesmas filas compartilhadas é chamado de grupo de filas compartilhadas. Cada membro do grupo de filas compartilhadas tem acesso ao mesmo conjunto de filas compartilhadas.

Os grupos de filas compartilhadas têm um nome de até quatro caracteres. O nome deve ser exclusivo em sua rede e deve ser diferente de qualquer nome de gerenciador de filas.

A [Figura 59](#) na [página 174](#) ilustra um grupo de filas compartilhadas que contém dois gerenciadores de filas. Cada gerenciador de filas tem um inicializador de canais e seus próprios conjuntos de páginas e conjuntos de dados do log.

Cada membro do grupo de filas compartilhadas também deve se conectar a um sistema Db2. O Db2 os sistemas devem estar no mesmo Db2 grupo de compartilhamento de modo que os gerenciadores de filas possam acessar o repositório compartilhado Db2 usado para reter as definições de objeto compartilhado. Essas são definições de qualquer tipo de objeto do IBM MQ (por exemplo, filas e canais) que são definidas apenas uma vez e, em seguida, qualquer gerenciador de filas no grupo pode usá-los. Elas são chamadas de definições *globais* e são descritas em [Definições Privadas e Globais](#).

Mais de um grupo de filas compartilhadas pode fazer referência a um determinado grupo de compartilhamento de dados. Você especifica o nome do subsistema Db2 e qual grupo de compartilhamento de dados um gerenciador de filas usa nos parâmetros do sistema IBM MQ na inicialização.



Quando um gerenciador de filas tiver se associado a um grupo de filas compartilhadas, ele terá acesso aos objetos compartilhados definidos para esse grupo e será possível usar esse gerenciador de filas para definir novos objetos compartilhados dentro do grupo. Se as filas compartilhadas estiverem definidas no grupo, será possível usar esse gerenciador de filas para colocar as mensagens nessas filas compartilhadas e obter as mensagens delas. Qualquer gerenciador de filas no grupo pode recuperar as mensagens retidas em uma fila compartilhada.

É possível inserir um comando do MQSC uma vez e fazer com que ele seja executado em todos os gerenciadores de filas no grupo de filas compartilhadas como se tivesse sido inserido em cada gerenciador de filas individualmente. O atributo *escopo do comando* é usado para isso. Este atributo é descrito em [Direcionando Comandos para Diferentes Gerenciadores de Filas](#).

Quando um gerenciador de filas é executado como um membro de um grupo de filas compartilhadas, deve ser possível distinguir entre os objetos do IBM MQ definidos privadamente para esse gerenciador de filas e objetos do IBM MQ definidos globalmente que estão disponíveis para todos os gerenciadores de filas no grupo de filas compartilhadas. O atributo de *disposição do grupo de filas compartilhadas* é usado para isso. Este atributo é descrito em [Definições Privadas e Globais](#).

É possível definir um único conjunto de perfis de segurança que controla o acesso aos objetos do IBM MQ em qualquer lugar no grupo. Isso significa que o número de perfis que você tem que definir é muito reduzido.

Um gerenciador de filas pode pertencer a apenas um grupo de filas compartilhadas e todos os gerenciadores de filas no grupo devem estar no mesmo sysplex. Você especifica a qual grupo de filas compartilhadas o gerenciador de filas pertence nos parâmetros do sistema na inicialização.

Conceitos relacionados

[“Onde Estão Retidas as Mensagens de Filas Compartilhadas?” na página 175](#)

Cada mensagem em uma fila compartilhada é representada por uma entrada em uma estrutura de lista do recurso de acoplamento do z/OS. Se os dados da mensagem forem muito grandes para se ajustarem na mesma entrada, eles serão transferidos para um conjunto de dados de mensagem compartilhada (SMDS) ou para Db2.

[“Vantagens de Usar Filas Compartilhadas” na página 194](#)

A fila compartilhada permite que aplicativos IBM MQ sejam escaláveis, altamente disponíveis e permite que o balanceamento de carga seja implementado.

[“Enfileiramento distribuído e grupos de filas compartilhadas” na página 215](#)

O enfileiramento distribuído e os grupos de filas compartilhadas são duas técnicas que você pode usar para aumentar a disponibilidade dos sistemas do aplicativo. Use este tópico para localizar informações adicionais sobre essas técnicas.

[“Influenciando a Distribuição de Carga de Trabalho com Filas Compartilhadas” na página 218](#)

Use este tópico para entender os fatores que afetam a distribuição de carga de trabalho com filas compartilhadas em um grupo de filas compartilhadas.

Referências relacionadas

[“Onde localizar mais informações sobre filas compartilhadas e grupos de filas compartilhadas?” na página 219](#)

Use a tabela neste tópico para localizar mais informações sobre como o IBM MQ for z/OS usa as filas compartilhadas e os grupos de filas compartilhadas.

Onde Estão Retidas as Mensagens de Filas Compartilhadas?

Cada mensagem em uma fila compartilhada é representada por uma entrada em uma estrutura de lista do recurso de acoplamento do z/OS. Se os dados da mensagem forem muito grandes para se ajustarem na mesma entrada, eles serão transferidos para um conjunto de dados de mensagem compartilhada (SMDS) ou para Db2.

Se a estrutura CF tiver sido configurada para usar o System Class Memory (SCM), o IBM MQ poderá usá-lo sem configuração adicional.

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

Armazenamento da Mensagem da Fila Compartilhada

As mensagens que são colocadas em filas compartilhadas não são armazenadas nos conjuntos de páginas e não usam os buffer pools.

As mensagens em filas compartilhadas têm entradas em estruturas de lista no recurso de acoplamento (CF) do z/OS. Vários gerenciadores de filas no mesmo sysplex pode acessar essas mensagens usando a estrutura de lista CF.

Os dados da mensagem para pequenas mensagens de fila compartilhada normalmente são incluídos na entrada do recurso de acoplamento. Para mensagens maiores, os dados da mensagem podem ser armazenados em um conjunto de dados de mensagem compartilhada (SMDS), ou como um ou mais objetos binários grandes (BLOBs) em uma tabela do Db2 que é compartilhada por um grupo de compartilhamento de dados do Db2. Os dados da mensagem que excedem 63 KB são sempre transferidos para SMDS ou Db2. Opcionalmente, mensagens menores também podem ser transferidas da mesma maneira para economizar espaço na estrutura do recurso de acoplamento. Consulte [“Especificando Opções de Transferência para Mensagens Compartilhadas”](#) na página 178 para obter mais detalhes.

Mensagens colocadas em uma fila compartilhada são referidas em uma estrutura do recurso de acoplamento até serem recuperadas por um MQGET. As operações do recurso de acoplamento são usadas para:

- Procurar a próxima mensagem recuperável
- Bloquear mensagens não confirmadas em filas compartilhadas
- Notificar gerenciadores de filas interessados sobre a recuperação de mensagens confirmadas

As operações MQPUT e MQGET em mensagens persistentes são registradas no log do gerenciador de filas que executa essa operação. Isso minimiza o risco de perda de dados no caso de uma falha do recurso de acoplamento.

O Recurso de Acoplamento

As mensagens retidas em filas compartilhadas são referidas dentro de um recurso de acoplamento. O recurso de acoplamento fica fora de qualquer uma das imagens do z/OS no sysplex e normalmente é configurado para ser executado em uma fonte de alimentação diferente. O recurso de acoplamento é, portanto, resiliente para falhas de software e pode ser configurado como resiliente para falhas de hardware ou indisponibilidades de energia. Isso significa que mensagens armazenadas no recurso de acoplamento ficam altamente disponíveis.

Cada estrutura de lista de recurso de acoplamento usada pelo IBM MQ é dedicada a um grupo de filas compartilhadas específico, mas um recurso de acoplamento pode conter estruturas para mais de um grupo de filas compartilhadas. Os gerenciadores de filas em diferentes grupos de filas compartilhadas não podem compartilhar os dados. Até 32 gerenciadores de filas em um grupo de filas compartilhadas podem se conectar a uma estrutura de lista de recurso de acoplamento ao mesmo tempo.

Uma única estrutura de lista de recurso de acoplamento pode conter até 512 filas compartilhadas. A quantidade total de dados da mensagem armazenados na estrutura é limitada pela capacidade de estrutura. No entanto, com o **CFLEVEL (5)**, é possível usar os parâmetros de transferência para transferir dados para mensagens menores que 63 KB, aumentando assim o número de mensagens que podem ser armazenadas na estrutura, embora cada mensagem ainda requeira pelo menos uma entrada do recurso

de acoplamento mais pelo menos 768 bytes de dados, compostos por 256 bytes para a entrada e 512 bytes para os dois elementos de cabeçalho e descritor.

O tamanho da estrutura de lista está registro pelos seguintes fatores:

- Ele deve ficar dentro de um único recurso de acoplamento.
- Ele pode compartilhar o armazenamento do recurso de acoplamento disponível com outras estruturas para IBM MQ e outros produtos.

As estruturas de lista de recursos de acoplamento podem ter memória de classe de armazenamento associada a elas. Em determinadas situações, esta memória de classe de armazenamento pode ser útil quando usada com filas compartilhadas. Consulte [“Uso da memória de classe de armazenamento com filas compartilhadas”](#) na página 195 para obter mais informações.

Planejando o Tamanho da Estrutura CF

Se você precisar de orientação sobre o tamanho de suas estruturas CF, poderá usar o [MP16: IBM MQ for z/OS Planejamento de capacidade e ajuste de supportpac](#). Também é possível usar a ferramenta baseada na web [CFSizer](#), que é fornecida pela IBM para ajudá-lo com tamanhos de CF.

O Objeto da Estrutura CF

O uso do gerenciador de filas de uma estrutura do recurso de acoplamento é especificado em um objeto IBM MQ da estrutura de CF (CFSTRUCT).

Esses objetos de estrutura são armazenados em Db2.

Ao usar os comandos ou definições do z/OS relacionados a uma estrutura do recurso de acoplamento, os quatro primeiros caracteres do nome do grupo de filas compartilhadas são necessários. No entanto, um objeto CFSTRUCT do IBM MQ sempre existe dentro de um único grupo de filas compartilhadas e, portanto, seu nome não inclui os quatro primeiros caracteres do nome do grupo de filas compartilhadas. Por exemplo, o CFSTRUCT (MYDATA) definido no grupo de filas compartilhadas que começa com SQ03 usaria a estrutura de lista de recurso de acoplamento SQ03MYDATA.

As estruturas CF possuem um atributo CFLEVEL que determinar suas capacidades funcionais:

- 1, 2 - pode ser usado para mensagens não persistentes com menos de 63 KB
- 3 - pode ser usado para mensagens não persistentes e persistentes com menos de 63 KB
- 4 - pode ser usado para mensagens não persistentes e persistentes de até 100 MB
- 5 – pode ser usado para mensagens não persistentes e de até 100 MB e transferido seletivamente para conjuntos de dados de mensagem compartilhada (SMDS) ou Db2.

Nota: Ao usar o IBM MQ , é possível criptografar uma estrutura do recurso de acoplamento Consulte [Criptografando dados da estrutura do recurso de acoplamento](#) para obter mais informações

Backup e Recuperação do Recurso de Acoplamento

É possível fazer backup de estruturas da lista de recurso de acoplamento usando o comando BACKUP CFSTRUCT do IBM MQ. Isso coloca uma cópia das mensagens persistentes atualmente na estrutura CF no conjunto de dados de log ativo do gerenciador de filas que faz o backup e grava um registro do backup para o Db2.

Se o recurso de acoplamento falhar, é possível usar o comando RECOVER CFSTRUCT do IBM MQ. Isso usa o registro de backup de Db2 para localizar e restaurar as mensagens persistentes do backup da estrutura CF. Qualquer atividade desde o último backup é reproduzida usando os logs de todos os gerenciadores de filas no grupo de filas compartilhadas e a estrutura de CF é restaurada até o ponto antes da falha.

Consulte os comandos [BACKUP CFSTRUCT](#) e [RECOVER CFSTRUCT](#) para obter detalhes adicionais.

Conceitos relacionados

“Especificando Opções de Transferência para Mensagens Compartilhadas” na página 178

É possível escolher onde os dados da mensagem para uma mensagem de fila compartilhada serão armazenados, em uma tabela do Db2 ou um conjunto de dados de mensagem compartilhada (SMDS). Também é possível selecionar mais mensagens são transferidas, com base no tamanho da mensagem e no uso atual da estrutura do recurso de acoplamento (CF).

“Gerenciando seu Ambiente de Shared Message Data Set (SMDS)” na página 180

Se selecionar conjuntos de dados de mensagem compartilhada para transferir mensagens grandes, você deverá estar ciente das informações que o IBM MQ usa para gerenciar esses conjuntos de dados e os comandos usados para trabalhar com essas informações. Use este tópico para entender como gerenciar conjuntos de dados de mensagem compartilhada.

Especificando Opções de Transferência para Mensagens Compartilhadas

É possível escolher onde os dados da mensagem para uma mensagem de fila compartilhada serão armazenados, em uma tabela do Db2 ou um conjunto de dados de mensagem compartilhada (SMDS). Também é possível selecionar mais mensagens são transferidas, com base no tamanho da mensagem e no uso atual da estrutura do recurso de acoplamento (CF).

Os dados da mensagem para filas compartilhadas podem ser transferidos do recurso de acoplamento e armazenados em uma tabela Db2 ou em um conjunto de dados gerenciado IBM MQ denominado *conjunto de dados de mensagem compartilhada* (SMDS).

Para mensagens maiores que o tamanho da entrada do recurso de acoplamento de 63 KB, a transferência de dados da mensagem para um SMDS pode ter uma melhoria de desempenho significativa comparada à transferência para Db2.

Cada mensagem da fila compartilhada ainda é gerenciada usando uma entrada da lista em uma estrutura do recurso de acoplamento, mas quando os dados da mensagem são transferidos para o SMDS, a entrada do recurso de acoplamento contém apenas algumas informações de controle e uma lista de referências para os blocos de disco relevantes em que a mensagem é armazenada. Usar esse mecanismo significa que a quantidade do armazenamento de elemento do recurso de acoplamento necessária para cada mensagem é apenas uma fração do tamanho real da mensagem.

Selecionando onde as Mensagens de Fila Compartilhadas São Armazenadas

A seleção de armazenamento de mensagens compartilhadas SMDS ou Db2 é controlada com o parâmetro **OFFLOAD (SMDS | DB2)** na definição **CFSTRUCT**. **OFFLOAD (SMDS)** é o valor padrão.

Esse parâmetro também requer que o **CFSTRUCT** use **CFLEVEL (5)** ou superior.

O parâmetro **OFFLOAD** é válido somente a partir de **CFLEVEL (5)**. Consulte [DEFINE CFSTRUCT](#) para obter detalhes adicionais.

OFFLOAD (DB2) é suportado principalmente para propósitos de migração.

Selecionando quais mensagens de fila compartilhadas são transferidas

Os dados da mensagem são transferidos para o SMDS ou Db2, com base no tamanho dos dados da mensagem e no uso atual da estrutura do recurso de acoplamento. Existem três regras e cada regra especifica um par correspondente de parâmetros. Esses parâmetros são uma porcentagem de limite de uso da estrutura do recurso de acoplamento correspondente (**OFFLDnTH**) e um limite de tamanho da mensagem (**OFFLDnSZ**).

A implementação atual das três regras é especificada usando os seguintes pares de palavra-chave:

- OFFLD1TH e OFFLD1SZ
- OFFLD2TH e OFFLD2SZ
- OFFLD3TH e OFFLD3SZ

Par de regras	Valor padrão	Descrição
Par de regras 1	OFFLD1TH(70) e OFFLD1SZ(32K)	Se a estrutura do recurso de acoplamento for maior que 70% da capacidade de dados de transferência para mensagens que excedem 32 KB
Par de regras 2	OFFLD2TH(80) e OFFLD2SZ(4K)	Se a estrutura do recurso de acoplamento for maior que 80% da capacidade dos dados de transferência para mensagens que excedem 4 KB
Par de regras 3	OFFLD3TH(90) e OFFLD3SZ(0K)	Se a estrutura do recurso de acoplamento for maior que 90% da capacidade de dados de transferência para mensagens que excedem 0 KB (todas as mensagens)

Se uma regra de transferência tiver o valor OFFLD x SZ de 64 K isso indicará que a regra não está em vigor. Nesse caso, as mensagens serão transferidas somente se outra regra de transferência estiver em vigor ou se a mensagem for maior que 63,75 KB e, portanto, muito grande para ser armazenada na estrutura.

Cada mensagem que é transferida ainda requer 0,75 KB de armazenamento no recurso de acoplamento.

As três regras de transferência que podem ser especificadas para cada estrutura devem ser usadas da seguinte forma.

- Desempenho

- Quando houver muito espaço na estrutura do aplicativo, os dados da mensagem só deverão ser transferidos se forem muito grandes para serem armazenados na estrutura ou se excederem algum limite de tamanho de mensagem inferior, de modo que o valor do desempenho do seu armazenamento na estrutura não compense a quantidade de espaço de estrutura que seria necessária.
- Se um limite de tamanho de mensagem específico for necessário, ele será especificado convencionalmente com o uso da primeira regra de transferência.

- Capacidade

- Quando houver pouco espaço na estrutura do aplicativo, a quantidade máxima de dados da mensagem deverá ser transferida para que seja feito o melhor uso possível do espaço restante.
- A terceira regra de transferência é usada convencionalmente para indicar que quando a estrutura está quase cheia, a maioria das mensagens é transferida, assim as entradas na estrutura do aplicativo ficam do tamanho mínimo (exigindo cerca de 0.75K bytes).
- O parâmetro de limite de uso deve ser escolhido com base no tamanho da estrutura do aplicativo e na lista não processada máxima prevista. Por exemplo, se a lista não processada máxima prevista for 1M mensagens, a quantidade de armazenamento de estrutura necessária para esse número de mensagens será de aproximadamente 0.75G bytes. Isso significa, por exemplo, que se a estrutura for de cerca de 10G bytes, o limite de uso para a transferência de todas as mensagens deverá ser configurado para 92% ou menos.
- O espaço da estrutura é dividido em elementos e entradas, e mesmo que haja espaço suficiente no geral, um pode ficar sem antes que o outro. O sistema fornece recursos AUTOALTER para ajustar a proporção quando necessário, mas eles não têm muito sensibilidade, portanto, a quantidade de espaço disponível de fato pode ser menor. Portanto, talvez seja melhor usar não

mais que 90% do espaço máximo da estrutura, de forma que, no exemplo anterior, o limite de uso para transferir todas as mensagens seria melhor configurado em torno de 80%.

- Transição em cushion:
 - Conforme a quantidade de espaço restante na estrutura do recurso de acoplamento diminui, seria indesejável ter uma grande mudança repentina nas características de desempenho. Também seria indesejável para o gerenciamento de recurso de acoplamento ter uma mudança de limite repentina na proporção típica de entradas para elementos sendo usados.
 - A segunda regra de transferência é usada convencionalmente para fornecer um cushion intermediário entre as regras de transferência influenciadas por desempenho e capacidade. Ela pode ser configurada para causar um aumento significativo na atividade de transferência quando o espaço usado na estrutura do recurso de acoplamento excede um limite intermediário. Isso significa que o espaço restante será usado mais lentamente e dará ao processamento de mudança automática do recurso de acoplamento mais tempo para se adaptar aos altos níveis de uso.

Se a estrutura do recurso de acoplamento não puder ser expandida e houver uma necessidade de armazenar pelo menos algum número de mensagens predeterminado, a terceira regra pode ser modificada conforme necessário para assegurar que a transferência de dados para todas as mensagens seja iniciada em um limite apropriada para assegurar que o espaço seja reservado para esse número de mensagens predeterminado.

Por exemplo, se o tamanho da estrutura do recurso de acoplamento for 4 GB, e o número predeterminado de mensagens for 1 milhão, então $1.000.000 * 0,75$ KB serão necessários, que é 768 MB, 18,75% de 4 GB. Neste caso, o limite para transferir todas as mensagens precisa ser configurado em torno de 80% em vez de 90%. Isso fornece parâmetros OFFLD3TH(80) e OFFLD3SZ(0K). Os outros parâmetros também precisariam ser ajustados.

Se for descoberto que a transferência de mensagens muito pequenas possui um impacto significativo no desempenho, mas o impacto relativo for menor para mensagens maiores, então os limites de uso para as outras regras podem ser reduzidos para transferir as mensagens anteriores, deixando mais espaço na estrutura para pequenas mensagens antes que elas precisem ser transferidas.

Por exemplo, se as mensagens que excedem 32KB ocorrerem com frequência, mas o desempenho do tempo decorrido para transferi-las (conforme determinado a partir do desempenho do aplicativo ou das estatísticas RMF) for muito semelhante ao para mantê-las em um recurso de acoplamento, então o limite para a primeira regra poderia ser configurada para 0% para transferir todas essas mensagens. Isso fornece parâmetros OFFLD1TH(0) e OFFLD1SZ(32K). Novamente, os outros parâmetros de transferência precisariam ser ajustados.

Se houver várias mensagens a respeito dos tamanhos intermediários específicos, como 16 KB e 6 KB, pode ser útil alterar as opções de tamanho da mensagem para a segunda regra para que os maiores sejam transferidos em um limite de uso bem menor, economizando uma quantidade de espaço significativa, mas os menores ainda são armazenados apenas no recurso de acoplamento.

Gerenciando seu Ambiente de Shared Message Data Set (SMDS)

Se selecionar conjuntos de dados de mensagem compartilhada para transferir mensagens grandes, você deverá estar ciente das informações que o IBM MQ usa para gerenciar esses conjuntos de dados e os comandos usados para trabalhar com essas informações. Use este tópico para entender como gerenciar conjuntos de dados de mensagem compartilhada.

Objetos SMDS

As propriedades e o status de cada conjunto de dados da mensagem compartilhada são rastreados em um objeto SMDS compartilhado que pode ser atualizado por meio de qualquer gerenciador de filas no grupo de filas compartilhadas.

Há um conjunto de dados de mensagem compartilhada para cada gerenciador de filas que pode acessar cada estrutura do aplicativo do recurso de acoplamento. O conjunto de dados de mensagem compartilhada é identificado pelo nome do gerenciador de filas de propriedade, especificado usando

a palavra-chave SMDS e pelo nome de estrutura do aplicativo, especificado usando a palavra-chave CFSTRUCT.

Nota: Ao definir conjuntos de dados SMDS para uma estrutura, deve-se ter um para cada gerenciador de filas.

O objeto SMDS é armazenado em uma matriz (com uma entrada por gerenciador de filas no grupo) que forma uma extensão do objeto CFSTRUCT correspondente armazenado no Db2.

Não há comando para definir (DEFINE) ou excluir (DELETE) o objeto SMDS porque ele é criado ou excluído como parte do objeto CFSTRUCT, mas existe um comando para alterá-lo (ALTER) para as configurações para o gerenciador de filas de propriedade individual.

Para obter informações adicionais sobre comandos SMDS, consulte [“Comandos Relacionados ao SMDS”](#) na página 193

Informações SMDSCONN

É possível para um conjuntos de dados de mensagem compartilhada estar em um estado normal, mas para um ou mais gerenciadores de filas não conseguir conectar a ele, por exemplo, devido a um problema com uma definição de segurança ou com conectividade de dispositivo de acesso direto. Portanto, é necessário para cada gerenciador de filas manter o controle do status de conexão e das informações de disponibilidade para cada conjuntos de dados de mensagem compartilhada, indicando, por exemplo, se pode conectar atualmente a ele e, se não, o motivo para isso.

As informações SMDSCONN representam uma conexão do gerenciador de filas para um conjunto de dados de mensagem compartilhada. Como para o conjunto de dados de mensagem compartilhada em si, isso é identificado pelo gerenciador de filas que possui o conjunto de dados de mensagem compartilhada (conforme especificado na palavra-chave SMDS para o próprio objeto compartilhado) combinado com o nome CFSTRUCT.

Não há parâmetro para identificar o gerenciador de filas de conexão porque os comandos endereçados a um gerenciador de filas específico podem ser referir apenas às informações SMDSCONN para o mesmo gerenciador de filas.

As entradas de informações SMDSCONN são mantidas no armazenamento principal no gerenciador de filas proprietário e são recriadas quando o gerenciador de filas é reiniciado. No entanto, se uma conexão de um gerenciador de filas individual tiver sido parado explicitamente, essas informações também serão armazenadas como um sinalizador em uma matriz de conexão no objeto CFSTRUCT ou SMDS correspondente, de modo que persiste em uma reinicialização do gerenciador de filas.

Status e Informações de Disponibilidade

As informações de status indicam o estado de um recurso ou conexão (por exemplo, se ainda não está sendo usado, está em uso normal ou está precisando de recuperação). Geralmente, é descrito usando a palavra-chave STATUS. Os valores possíveis dependem do tipo de objeto.

As informações de status são atualizadas normalmente automaticamente, por exemplo, quando um erro é detectado enquanto usa o recurso ou a conexão. No entanto, em alguns casos um comando também pode ser usado para atualizar o status, para permitir em casos quando não for possível para um gerenciador de filas determinar o status correto automaticamente.

As informações de disponibilidade indicam se o recurso ou a conexão pode ser usada e geralmente é determinada primariamente pelas informações de status. Para os tipos de recurso ou conexão usados no suporte do conjunto de dados de mensagem compartilhada, três níveis de disponibilidade são implementados:

Disponível

Isso significa que o recurso está disponível para ser usado normalmente. Isso não significa necessariamente que está em uso no presente (que pode ser determinado em vez do valor STATUS). Para um conjunto de dados, se precisar de processamento de reinicialização, isso permite que o gerenciador de filas de propriedade o abra, mas outros gerenciadores de filas devem esperar até que o conjunto de dados volte no estado ACTIVE.

Indisponível Devido a Erro

Isso significa que o recurso ficou indisponível automaticamente devido a um erro e não espera-se que fique disponível novamente até que alguma forma de processamento de reparo ou recuperação tenha sido executado. No entanto, as tentativas de disponibilizá-lo novamente são permitidas sem intervenção do operador. Essa tentativa também pode ser acionada por um comando para marcar o recurso como ativado ou um comando que altera o status de maneira a indicar que o processamento de recuperação foi concluído.

A razão pela qual o recurso ficou indisponível normalmente é óbvia a partir do valor STATUS relacionado, mas em alguns casos pode haver outras razões para tornar o recurso indisponível, no caso de um valor REASON separado ser fornecido para indicar a razão.

Indisponível Devido ao Comando do Operador

Isso indica que o acesso ao recurso foi explicitamente desativado por um comando. Isso pode ser disponibilizado apenas usando um comando para ativá-lo novamente.

Disponibilidade SMDS

Para o objeto SMDS compartilhado, a disponibilidade é descrita pela palavra-chave ACCESS, com os valores possíveis ENABLED, SUSPENDED e DISABLED.

A disponibilidade pode ser atualizada usando um comando **RESET SMDS** para o objeto compartilhado relevante a partir de qualquer gerenciador de filas no grupo para configurar ACCESS(ENABLED) ou ACCESS(DISABLED).

Se a disponibilidade era anteriormente ACCESS(SUSPENDED), alterá-la para ACCESS(ENABLED) acionará uma nova tentativa para usar o conjunto de dados de mensagem compartilhada, mas se o erro anterior ainda estiver presente, a disponibilidade será reconfigurada de volta para ACCESS(SUSPENDED).

Disponibilidade SMDSCONN

Para uma entrada de informações SMDSCONN local, a disponibilidade é descrita pela palavra-chave AVAIL, com valores possíveis NORMAL, ERROR ou STOPPED. A disponibilidade pode ser atualizada usando um comando **START SMDSCONN** ou **STOP SMDSCONN** endereçado a um gerenciador de filas específico para ativar ou desativar sua conexão

Se a disponibilidade era anteriormente AVAIL(ERROR), alterá-la para AVAIL(NORMAL) acionará uma nova tentativa de usar o conjuntos de dados de mensagem compartilhada, mas se o erro anterior ainda estiver presente, a disponibilidade será reconfigurada de volta para AVAIL(ERROR).

Disponibilidade e Status Compartilhado do Conjunto de Dados de Mensagem Compartilhada

A disponibilidade de cada conjunto de dados de mensagem compartilhada é gerenciada dentro do grupo usando informações de status compartilhadas, que podem ser exibidas usando o comando **DISPLAY CFSTATUS** com TYPE (SMDS) Isso exibe informações de status para cada gerenciador de filas que ativou um conjunto de dados para cada estrutura. Cada conjunto de dados pode estar em um dos seguintes estados:

NOTFOUND

Isso significa que o conjunto de dados correspondente ainda não foi ativado. Esse status aparece apenas quando um gerenciador de filas específico for especificado, porque os conjuntos de dados que não foram ativados são ignorados quando todos os gerenciadores de fila forem selecionados.

Novo

O conjunto de dados está sendo aberto e iniciado pela primeira vez, pronta para ser ativada.

ATIVO

Isso significa que o conjunto de dados está totalmente disponível e deve ser alocado e aberto por todos os gerenciadores de fila ativos para a estrutura.

FAILED

Isso significa que o conjunto de dados não está disponível (exceto para o processamento de recuperação) e deve ser fechado e desalocado por todos os gerenciadores de fila.

INRECOVER

Isso significa que a recuperação de mídia (usando RECOVER CFSTRUCT) está em andamento para este conjunto de dados.

RECOVERED

Isso indica que um comando foi emitido para alternar um conjunto de dados com falha de volta para o estado ativo, mas é necessário um processamento de reinicialização adicional e que ainda não está completo; portanto, o conjunto de dados pode ser aberto apenas pelo gerenciador de filas de propriedade para o processamento de reinicialização.

EMPTY

O conjunto de dados não contém mensagens. O conjunto de dados é colocado neste estado se for fechado normalmente pelo gerenciador de filas de propriedade, no momento em que não contém qualquer mensagem. Ele também pode ser colocado no estado EMPTY quando o conteúdo do conjunto de dados anterior tiver que ser descartado porque a estrutura do aplicativo foi esvaziada (usando **RECOVER CFSTRUCT** com TYPE PURGE ou, para uma estrutura não recuperável apenas, excluindo a instância anterior da estrutura. Na próxima vez em que o conjunto de dados for aberto por seu gerenciador de filas de propriedade, o mapa de espaço é reconfigurado como vazio e o status é mudado para ACTIVE. Como o conteúdo do conjunto de dados anterior não é mais necessário, um conjunto de dados neste estado pode ser substituído por um conjunto de dados recém-alocado, por exemplo, para alterar a alocação de espaço ou movê-la para outro volume.

A saída de comando inclui a data e a hora em que a criação de log de recuperação foi ativada, se houver, e a data e hora em que o conjunto de dados falhou, se não estiver atualmente ativa.

Um conjunto de dados de mensagem compartilhada pode ser colocado em um estado FAILED por um comando **RESET SMDS** ou automaticamente quando qualquer um dos seguintes tipos de erro for detectado:

- O conjunto de dados não pode ser alocado ou aberto pelo gerenciador de filas de propriedade.
- A validação do cabeçalho do conjunto de dados falha depois que tiver sido aberta com êxito por qualquer gerenciador de filas.
- Ocorre um erro de E/S permanente quando o gerenciador de filas de propriedade estiver lendo ou gravando dados.
- Um erro de E/S permanente ocorre quando outro gerenciador de filas estiver lendo os dados de um conjunto de dados que concluiu com êxito a validação e o processamento aberto.

Quando um conjunto de dados estiver no estado FAILED ou INRECOVER, ele não ficará disponível para uso normal, de modo que o estado de disponibilidade seja ACCESS(ENABLED), seja mudado para ACCESS(SUSPENDED).

Se um conjunto de dados tiver sido colocado no estado FAILED, mas nenhuma recuperação de mídia for necessária, por exemplo, porque os dados ainda eram válidos, mas o dispositivo de armazenamento estava temporariamente offline, o comando **RESET SMDS** pode ser usado para solicitar a mudança de status diretamente para o estado RECOVERED.

Quando o conjunto de dados entra no estado RECOVERED, na conclusão do processamento de recuperação ou como resultado do comando **RESET SMDS**, ele fica pronto para ser usado novamente assim que o processamento de reinicialização tiver sido concluído. Se estava no estado ACCESS(SUSPENDED), ele será alternado automaticamente de volta para o estado ACCESS(ENABLED), que permite que o gerenciador de filas de propriedade execute o processamento de reinício. Quando o processamento de reinicialização é concluído, o estado é mudado para ACTIVE e todos os outros gerenciadores de fila podem então se conectar ao conjunto de dados novamente.

Disponibilidade e Status da Conexão do Conjuntos de Dados de Mensagem Compartilhada

Cada gerenciador de filas mantém informações de disponibilidade e status local para sua conexão a cada conjunto de dados de mensagem compartilhada própria e de propriedade de outros gerenciadores de filas no grupo. Essas informações podem ser exibidas usando o comando **DISPLAY SMDSCONN**.

Se não for possível acessar um conjunto de dados de mensagem compartilhada no estado ACTIVE que pertence a outro gerenciador de filas, ele sinalizará a conexão como estando indisponível de seu próprio ponto de vista.

Se o erro indicar definitivamente um problema com o conjunto de dados em si, o gerenciador de filas também vai alterar automaticamente o status compartilhado para indicar que o conjunto de dados agora esteja no estado FAILED. No entanto, se o erro pudesse ser causado por um problema ambiental, por exemplo, por não estar autorizado a abrir o conjunto de dados, o gerenciador de filas emitirá mensagens de erro e tratará o conjunto de dados como estando indisponível, mas não modificará o status do conjunto de dados compartilhado. Se o erro ambiental se transformar em um problema com o conjunto de dados de alguma forma (por exemplo, se foi alocado em um dispositivo que não possa ser acessado por algum dos gerenciadores de fila), um operador poderá usar o comando RESET SMDS especificando STATUS(FAILED) para permitir que o conjunto de dados seja recuperado ou reparado conforme necessário.

Se uma conexão a um conjunto de dados de mensagem compartilhada não puder ser estabelecida, mas o conjunto de dados parecer ser válido, uma nova tentativa de usá-lo pode ser acionada emitindo o comando **START SMDSCONN** para o gerenciador de filas de propriedade.

Se houver uma necessidade operacional de finalizar a conexão entre um gerenciador de filas específico e um conjunto de dados temporariamente, mas o conjunto de dados em si não estiver danificado, o conjunto de dados pode ser fechado e desalocado usando o comando **STOP SMDSCONN**. Se o conjunto de dados estiver em uso, o gerenciador de filas o fechará normalmente (embora quaisquer solicitações para os dados nesse conjunto de dados ainda sejam rejeitados com um código de retorno). Se esse for o conjunto de dados de propriedade, o gerenciador de filas salvará o mapa de espaço durante o processamento CLOSE, evitando a necessidade de processamento de reinicialização.

Se um conjunto de dados precisar ser retirado de serviço temporariamente de todos os gerenciadores de filas (por exemplo, para movê-lo), mas não estiver danificado, será melhor usar **STOP SMDSCONN** para o conjunto de dados relevante com a opção CMDSCOPE (*) para parar os gerenciadores de filas usando-o primeiro, pois isso evitará a necessidade de reiniciar o processamento quando o conjunto de dados for trazido de volta ao serviço. Por outro lado, se o conjunto de dados estiver marcado como FAILED, isso informa aos gerenciadores de fila que eles devem parar de usá-lo imediatamente, o que significa que o mapa de espaço não será salvo e precisará ser reconstruído pelo processamento de reinicialização.

O acesso a quaisquer conjuntos de dados de mensagem compartilhada anteriormente no estado ACCESS(SUSPENDED) será recuperado se o gerenciador de filas for reiniciado.

Log de Recuperação do Conjunto de Dados de Mensagem Compartilhada

As mensagens compartilhadas persistentes são registradas em log para propósitos de recuperação de mídia. Isso significa que as mensagens podem ser recuperadas após qualquer falha de estruturas do recurso de acoplamento ou conjuntos de dados de mensagem compartilhada, contanto que os logs de recuperação ainda estejam intactos. As mensagens persistentes também podem ser recriadas por meio dos logs de recuperação em outro site para propósitos de recuperação de desastre.

Quando os dados da mensagem são gravados para um conjunto de dados de mensagem compartilhada, cada bloco gravado no conjunto de dados é registrado em log separadamente seguido pela entrada de mensagem (incluindo o mapa de dados) conforme descrito no recurso de acoplamento. O processo de recuperação sempre recupera a estrutura do recurso de acoplamento, mas não precisa recuperar conjuntos de dados de mensagens compartilhadas individuais, exceto quando o status do conjunto de dados é FAILED ou quando o status é ACTIVE, mas o registro de cabeçalho do conjunto de dados não é mais válido, indicando que o conjunto de dados foi recriado. Um conjunto de dados não é selecionado

para recuperação se seu status for ACTIVE e o cabeçalho do conjunto de dados ainda for válido, nem se o status for EMPTY, indicando que nenhuma mensagem foi armazenada nele no momento da falha.

Backups do Conjunto de Dados de Mensagem Compartilhada

Quando BACKUP CFSTRUCT é usado para fazer um backup das mensagens compartilhadas em uma estrutura do aplicativo, quaisquer dados das mensagens persistentes armazenadas nos conjuntos de dados de mensagem compartilhada terão backup feito ao mesmo tempo, como para as mensagens persistentes armazenadas em DB.

Recuperação do Conjunto de Dados de Mensagem Compartilhada

Se um conjunto de dados de mensagem compartilhada for corrompido ou perdido, ele precisará ser colocado no estado FAILED para impedir que os gerenciadores de fila o use até que tenha sido reparado. Isso normalmente acontece automaticamente, mas também pode ser feito usando o comando **RESET SMDS** especificando STATUS (FAILED).

Se o conjunto de dados de mensagem compartilhada continha quaisquer mensagens persistentes, elas podem ser recuperadas usando o comando RECOVER CFSTRUCT. Este comando primeiro restaura quaisquer dados de mensagem persistente para esse conjunto de dados de mensagem compartilhada a partir do comando BACKUP CFSTRUCT mais recente e, em seguida, aplica todas as mudanças registradas em log desde esse momento. Se nenhum comando **BACKUP CFSTRUCT** foi executado desde o momento em que o conjunto de dados foi ativado pela primeira vez, ele será reconfigurado para esvaziar todas as mudanças desde quando a ativação foi aplicada.

Se o conteúdo CFSTRUCT e todos os conjuntos de dados de mensagens compartilhadas estiverem indisponíveis, por exemplo, em uma situação de recuperação de desastres, todos eles poderão ser recuperados em um único comando **RECOVER CFSTRUCT**

Se o conjunto de dados de mensagem compartilhada estiver danificado, mas a recuperação não estava ativada para CFSTRUCT ou o log que contém o BACKUP CFSTRUCT mais recente estiver indisponível ou inutilizável, as mensagens transferidas para esse conjunto de dados não poderão ser recuperadas. Neste caso, o comando **RECOVER CFSTRUCT** com o parâmetro TYPE(PURGE) pode ser usado para marcar o conjunto de dados de mensagem compartilhada como vazio ou excluir quaisquer mensagens da estrutura que tinha dados armazenados no conjunto de dados.

Quando o comando **RECOVER CFSTRUCT** é emitido, o status do conjunto de dados de mensagem compartilhada muda de FAILED para INRECOVER. Se a recuperação for concluída com êxito, o status será automaticamente mudado para RECOVERED; caso contrário, será mudado de volta para FAILED.

Quando o conjunto de dados é mudado para o estado RECOVERED, isso informa ao gerenciador de filas de propriedade que agora pode tentar abrir o conjunto de dados e executar o processamento de reinicialização.

Pontos de Sincronização e Recuperação do Conjunto de Dados de Mensagem Compartilhada

O processo de recuperação do conjunto de dados de mensagem compartilhada reaplica as mudanças para todos os registros de log concluídos até o fim do log, independente dos pontos de sincronização.

Se foram feitas mudanças no ponto de sincronização, o processamento de reinicialização ou recuperação para CFSTRUCT pode resultar na restauração de solicitações não confirmadas; portanto, algumas mudanças recuperadas podem não ser usadas realmente, mas não haverá nenhum prejuízo em recuperá-las de alguma maneira.

Também é possível que uma mensagem MQPUT não confirmada tenha sido gravada na estrutura, mas os dados correspondentes podem não ter sido gravados para o conjunto de dados ou log (porque a conclusão de E/S é forçada apenas no início do ponto de sincronização). Isso é inofensivo porque o processamento de reinicialização voltará a entrada da mensagem na estrutura, de modo que o fato de se referir a dados não recuperados não importa.

Processamento de Reinicialização do Conjunto de Dados de Mensagem Compartilhada

Se uma conexão do gerenciador de filas para um CFSTRUCT for finalizado normalmente, o gerenciador de filas gravará o mapa de espaço de bloco livre para cada conjunto de dados de mensagem compartilhada para uma área do ponto de verificação no conjunto de dados, pouco antes do fechamento do conjunto de dados. O mapa de espaço pode então ser lido novamente no tempo de reinicialização de conexão, contanto que o CFSTRUCT ou o conjunto de dados de mensagem compartilhada não precise de qualquer processamento de recuperação antes da próxima reinicialização.

No entanto, um gerenciador de filas finaliza de maneira anormal ou a estrutura ou conjunto de dados requerem algum processamento de recuperação, então o processamento adicional é necessário para reconstruir o mapa de espaço dinamicamente quando a conexão do gerenciador de filas para a estrutura for reiniciada.

Considerando que o próprio conjunto de dados não precisava ser recuperado, a reinicialização do gerenciador de filas simplesmente varre o conteúdo atual da estrutura para localizar as referências para os dados de mensagens de propriedade do gerenciador de filas atual e marca os blocos de dados relevantes como de propriedade do mapa de espaço. Outros gerenciadores de fila podem continuar usando a estrutura e ler os dados de propriedade o gerenciador de filas de reinicialização enquanto o mapa de espaço estiver sendo reconstruído.

Reinicialização do Conjunto de Dados de Mensagem Compartilhada após Recuperação

Se um conjunto de dados de mensagem compartilhada tivesse que ser recuperado a partir de um backup, todas as mensagens não persistentes armazenadas no conjunto de dados terão sido perdidas e se o conjunto de dados não fosse recuperado usando TYPE(PURGE) todas as mensagens armazenadas no conjunto de dados terão sido perdidas. Até que a recuperação seja concluída, o conjunto de dados será marcado como FAILED ou INRECOVER, de modo que qualquer tentativa de ler uma das mensagens afetadas de outro gerenciador de filas retorne um código de erro indicando que o conjunto de dados está temporariamente indisponível.

Quando o conjunto de dados tiver sido recuperado, o status será mudado para RECOVERED, o que permite que o gerenciador de filas de propriedade o abra reiniciando o processamento, mas o conjunto de dados permanece indisponível para outros gerenciadores de fila. A reinicialização do gerenciador de filas varre a estrutura para reconstruir o mapa de espaço para quaisquer mensagens restantes. A varredura também verifica as mensagens para as quais os dados foram perdidos e as exclui da estrutura (ou, se necessário, as sinalizada como perdidas, para serem excluídas posteriormente).

O status do conjunto de dados é mudado automaticamente de RECOVERED para ACTIVE quando essa varredura de reinicialização é concluída, ponto em que outros gerenciadores de fila podem começar a usá-lo novamente.

Informações de Uso do Conjunto de Dados de Mensagem Compartilhada

O comando DISPLAY USAGE agora também mostra informações sobre o espaço de conjunto de dados de mensagem compartilhada e uso de buffer pool para quaisquer conjuntos de dados de mensagem compartilhada atualmente abertos. Essas informações são exibidas se a nova opção TYPE(SMDS) ou a opção TYPE(ALL) existente for especificada.

Considerações sobre Capacidade e Desempenho dos Dados da Mensagem Compartilhada

Monitorando o uso do conjunto de dados

A atual porcentagem cheia de cada conjunto de dados de mensagem compartilhada com proprietário pode ser exibida pelo comando **DISPLAY USAGE** com a opção **TYPE(SMDS)**.

O gerenciador de filas expandirá normalmente um conjunto de dados de mensagens compartilhadas de forma automática quando chegar a 90% cheio, desde que a opção **DSEXPAND(YES)** esteja em vigor para a definição SMDS. Isso se aplica quando a opção SMDS é configurada como **DSEXPAND(YES)** ou a opção SMDS é configurada como **DSEXPAND(DEFAULT)** e a opção padrão CFSTRUCT é configurada como **DSEXPAND(YES)**.

Se a tentativa de expansão falhar porque nenhum tamanho de alocação secundário foi especificado quando o conjunto de dados foi criado (fornecendo a mensagem IEC070I com código de razão 203), o gerenciador de filas repete a solicitação de expansão usando uma alocação secundária de substituição de aproximadamente 20% do tamanho atual.

Quando um conjunto de dados é expandido, as novas extensões de conjunto de dados são formatadas como parte do processamento de expansão, o que pode levar dezenas de segundos ou até minutos, para extensões muito grandes. O novo espaço se torna disponível para uso após a formatação ser concluída e o catálogo ser atualizado para mostrar o novo intervalo de controle alto usado.

Se novas mensagens estiverem sendo criadas muito rapidamente, é possível que o conjunto de dados existente fique cheio antes de o processamento de expansão ser concluído. Nesse caso, qualquer solicitação que não pôde alocar espaço é temporariamente suspensa, até que a tentativa de expansão seja concluída e o novo espaço fique disponível para uso. Se a expansão foi bem-sucedida, automaticamente será feita uma nova solicitação.

Se uma tentativa de expansão falhar, por falta de espaço disponível ou porque as extensões máximas já foram atingidas, uma mensagem será emitida dando a razão para a falha, em seguida, a opção de substituição para o SMDS afetado será alterada automaticamente para **DSEXPAND(NO)** para evitar novas tentativas de expansão. Nesse caso, há risco de o conjunto de dados ficar cheio, em cujo caso pode ser necessário tomar outras ações, conforme descrito em [Conjunto de dados fica cheio](#).

Monitorando o uso da estrutura do aplicativo

O nível de uso de uma estrutura de aplicativo pode ser exibido usando o comando MVS **DISPLAY XCF, STRUCTURE** especificando o nome completo da estrutura de aplicativo (incluindo o prefixo do grupo de filas compartilhadas). A mensagem de resposta IXC360I mostra o atual uso de elementos e entradas.

Quando o uso da estrutura exceder o valor **FULLTHRESHOLD** especificado na política CFRM, o sistema emitirá a mensagem IXC585E e poderá executar ações **ALTER** automáticas se elas estiverem especificadas, o que pode alterar a entrada na proporção do elemento ou aumentar o tamanho da estrutura.

Otimizando tamanhos de buffer pool

Cada buffer em um buffer pool compartilhado é usado para ler ou gravar um intervalo contíguo de páginas para uma mensagem até o tamanho do bloco lógico. Se a mensagem vazar para outros blocos, cada intervalo de páginas em um bloco separado exigirá um buffer separado.

Buffers contendo dados de mensagem após uma operação de gravação ou leitura são retidos no armazenamento e reutilizados usando um esquema de cache menos utilizado recentemente (LRU), de modo que uma solicitação para ler novamente os mesmos dados logo em seguida não precise ser gravada em disco. Isso oferece uma otimização significativa quando mensagens compartilhadas são gravadas e lidas logo em seguida pelos aplicativos em execução no mesmo sistema. Se mensagens pertencentes a outro gerenciador de filas forem navegadas para propósitos de seleção e depois recuperadas, isso eliminará a necessidade de reler a mensagem no disco.

Isso significa que o número de buffers requerido para cada estrutura de aplicativo é um para cada solicitação de API simultânea que lê ou grava mensagens grandes para essa estrutura de aplicativo, mais algum número de buffers adicionais que serão usados para salvar dados acessados recentemente para otimizar acessos de leitura subsequentes.

Para buffer pools compartilhados, se os buffers forem insuficientes, as solicitações de API simplesmente aguardarão se não houver nenhum buffer disponível. Porém, essa situação deve ser evitada, já que pode causar uma degradação significativa no desempenho.

As estatísticas do comando **DISPLAY USAGE** para buffer pools compartilhados mostram se houve alguma espera de buffer no intervalo de estatísticas atual e também mostra o número mais baixo de buffers livres (ou um valor negativo indicando o número máximo de encadeamentos que esperaram por um buffer a qualquer momento), o número de buffers que salvaram dados e a porcentagem de vezes que uma solicitação de buffer localizou com êxito os dados salvos na cadeia de LRU ("ocorrências de LRU") em vez de ter que ler ("perdas de LRU")¹.

- Se houve alguma espera, o número de buffers deve ser aumentado.
- Se houver muitos buffers não usados, o número de buffers pode ser reduzido para disponibilizar mais armazenamento na região para outros propósitos.
- Se houver vários buffers contendo dados salvos, mas a proporção de leituras que ocorreram nesses dados salvos for muito pequena, o número de buffers pode ser reduzido se o armazenamento puder ser melhor aproveitado para outros propósito. Porém, o número de buffers não deve ser mais reduzido do que o número mais baixo de buffers livres, já que isso poderia acionar esperas, e deve, de preferência, ser alto o suficiente para que a contagem de buffers livres mais baixa normalmente fique acima de zero.

Excluindo os Conjuntos de Dados de Mensagem Compartilhada

O comando **DELETE CFSTRUCT** (que só é permitido quando todas as filas compartilhadas na estrutura estão vazias e fechadas) não exclui os próprios conjuntos de dados de mensagens compartilhadas, mas eles podem ser excluídos da maneira usual depois que esse comando for concluído. Se o mesmo conjunto de dados tiver que ser reutilizado como um conjunto de dados de mensagem compartilhada, ele deve ser reformatado primeiro para reconfigurá-lo para o estado vazio.

Situações de Exceção para Conjuntos de Dados de Mensagem Compartilhada

Há inúmeras situações de exceção que podem ocorrer durante um uso normal, mesmo quando não há nenhum erro de software ou hardware presente.

Conjunto de dados fica cheio

Se um conjunto de dados ficar cheio, mas não puder ser expandido ou se a tentativa de expansão falhar, os aplicativos usando o gerenciador de filas correspondente para gravar mensagens grandes na estrutura do aplicativo correspondente receberão o erro 2192, **MQRC_STORAGE_MEDIUM_FULL** (também conhecido como **MQRC_PAGESET_FULL**).

Um conjunto de dados pode ficar cheio devido a uma falha no aplicativo que deve processar os dados, fazendo com que uma grande lista não processada de mensagens seja acumulada. Nesse caso, a expansão do conjunto de dados será uma solução temporária, e é importante que o aplicativo de processamento recomece o mais rápido possível.

Se mais espaço puder ser disponibilizado, o comando **ALTER SMDS** poderá ser usado para configurar **DSEXPAND(YES)** ou **DSEXPAND(DEFAULT)** (supondo que YES foi configurado ou foi assumido como o **DSEXPAND** padrão para a definição **CFSTRUCT**) para acionar uma nova tentativa. Entretanto, se a razão da falha foi porque as extensões máximas foram atingidas, a nova tentativa de expansão será rejeitada com uma mensagem e **DSEXPAND(NO)** será configurado novamente. Nesse caso, a única maneira de expandi-lo ainda mais é realocá-lo, o que envolve torná-lo temporariamente indisponível, conforme descrito a seguir.

Conjunto de dados precisa ser movido ou realocado

Se um conjunto de dados precisar ser movido ou expandido, mas estiver em uso normal, ele pode ser retirado de uso temporariamente para permitir sua movimentação ou realocação. Qualquer solicitação de API que tentar usar o conjunto de dados enquanto ele estiver indisponível receberá o código de razão **MQRC_DATA_SET_NOT_AVAILABLE**.

¹ $(\text{Hits} / (\text{Hits} + \text{Misses})) * 100$

1. Use o comando **RESET SMDS** para marcar o conjunto de dados como **ACCESS(DISABLED)**. Isso fará com que ele seja fechado normalmente e desalocado por todos os gerenciadores de filas atualmente conectados.
2. Mova ou realoque o conjunto de dados conforme necessário, copiando o conteúdo antigo para o conjunto de dados recém-alocado, por exemplo, usando o comando **REPRO** do Access Method Services (AMS).

Não tente pré-formatar o novo conjunto de dados antes de copiar os antigos dados nele, já que isso faria com que os dados copiados fossem anexados ao final do conjunto de dados formatado.

3. Use o comando **RESET SMDS** para marcar o conjunto de dados como **ACCESS(ENABLED)** novamente para disponibilizá-lo para uso.

Se o antigo conteúdo for menor que o tamanho do novo conjunto de dados, o restante do espaço será pré-formatado automaticamente quando o novo conjunto de dados for aberto.

Se o antigo conteúdo for maior que o tamanho do novo conjunto de dados, o gerenciador de filas terá que varrer as mensagens na estrutura do recurso de acoplamento e reconstruir o mapa de espaço para assegurar que nenhum dos dados ativos seja perdido. Se for localizada alguma referência a um bloco de dados fora das novas extensões, o conjunto de dados será marcado como **STATUS(FAILED)** e deverá ser reparado substituindo o conjunto de dados por um do tamanho correto, copiando novamente nele o conjunto de dados antigo ou usando **RECOVER CFSTRUCT** para recuperar quaisquer mensagens persistentes.

Estrutura do recurso de acoplamento tem pouco espaço

Se a estrutura do recurso de acoplamento estiver ficando sem espaço, causando uma mensagem IXC585E, vale a pena verificar se as regras de transferência foram configuradas para assegurar que, nesse caso, a quantidade máxima de dados está sendo transferida. Se não, as regras de transferência poderão ser modificadas usando o comando **ALTER CFSTRUCT**.

Situações de Erro para Conjuntos de Dados de Mensagem Compartilhada

Há inúmeros problemas dos quais devemos tomar conhecimento que só podem ser causados por erros e que não ocorrem em situações operacionais normais.

Conjunto de dados com proprietário não pode ser aberto

Se o gerenciador de filas que possui um conjunto de dados de mensagens compartilhadas não puder alocá-lo ou abri-lo, ou os atributos do conjunto de dados não forem suportados, o gerenciador de filas configurará um valor de status **SMDSCONN** apropriado de **ALLOCFAIL** ou **OPENFAIL** e configurará a disponibilidade do **SMDSCONN** como **AVAIL (ERROR)**. Ele também configura a disponibilidade do SMDS como **ACCESS (SUSPENDED)**. Quando o erro tiver sido corrigido, use o comando **RESET SMDS** para configurar **ACCESS(ENABLED)** para acionar uma nova tentativa ou emita o comando **START SMDSCONN** para o gerenciador de filas proprietário.

Conjunto de dados somente leitura não pode ser aberto

Se um gerenciador de filas não puder alocar ou abrir um conjunto de dados de mensagens compartilhadas de propriedade de outro gerenciador de filas e marcado como **STATUS(ACTIVE)**, ele assumirá que isso provavelmente se deve a um problema específico com a sua conexão ao conjunto de dados (representado pelo objeto **SMDSCONN**) em vez de um problema com o próprio conjunto de dados.

Ele marca o **SMDSCONN** como **STATUS(ALLOCFAIL)** ou **STATUS(OPENFAIL)** conforme apropriado e marca a disponibilidade do **SMDSCONN** como **AVAIL(ERROR)** para evitar novas tentativas de usá-lo.

Se o problema puder ser corrigido sem afetar o status do próprio conjunto de dados, use o comando **START SMDSCONN** para acionar uma tentativa.

Se o problema for um problema com o próprio conjunto de dados, o comando **RESET SMDS** poderá ser usado para marcar o conjunto de dados como **STATUS(FAILED)** até que ele tenha

sido recuperado. Quando o conjunto de dados tiver sido recuperado, a ação de mudança do status de volta para **STATUS (ACTIVE)** fará com que outros gerenciadores de filas sejam notificados. Se o **SMDSCONN** for marcado como **AVAIL (ERROR)**, ele será automaticamente alterado de volta para **AVAIL (NORMAL)** para acionar uma nova tentativa de abrir o conjunto de dados.

Cabeçalho do conjunto de dados está corrompido

Se o conjunto de dados foi aberto com sucesso, mas o formato das informações do cabeçalho estiver incorreto, o gerenciador de filas se fechará e desalocará o conjunto de dados e configurará o status como **STATUS (FAILED)** e a disponibilidade como **ACCESS (SUSPENDED)**. Isso permite que **RECOVER CFSTRUCT** seja usado para recuperar conteúdo.

Se o erro surgiu porque o conjunto de dados continha dados residuais de outro uso e não foi pré-formatado subsequentemente, pré-formate o conjunto de dados e use o comando **RESET SMDS** para alterar o status para **STATUS (RECOVERED)**.

Caso contrário, o conjunto de dados deverá ser recuperado.

Conjunto de dados fica vazio inesperadamente

Se o gerenciador de filas abrir um conjunto de dados marcado como **STATUS (ACTIVE)**, mas descobrir que ele é não inicializado ou recentemente pré-formatado, mas é válido, o gerenciador de filas se fechará e desalocará o conjunto de dados de mensagens compartilhadas, em seguida, configurará o status como **STATUS (FAILED)** e a disponibilidade como **ACCESS (SUSPENDED)**.

Conjunto de dados tem erros de E/S permanentes

Se um conjunto de dados tiver erros de E/S permanentes após um processamento de **OPEN** bem-sucedido, provavelmente ele precisa de recuperação. O gerenciador de filas marcará o conjunto de dados como **STATUS (FAILED)** para que todos os gerenciadores de filas atualmente conectados se fechem e o desaloquem.

Conjunto de dados tem erros de E/S recuperáveis

Se houver problemas de hardware com o conjunto de dados, é possível que isso resulte em erros de E/S recuperáveis, que não são refletidos de volta para o gerenciador de filas, mas que causam uma degradação significativa no desempenho e que também indicam um risco de erros de E/S permanentes no futuro.

Nesse caso, o conjunto de dados pode ser desativado para recuperação usando o comando **RESET SMDS** para marcá-lo como **STATUS (FAILED)**. Isso fará com que ele seja fechado e desalocado por todos os gerenciadores de filas; portanto, por exemplo, ele poderia ser movido para um novo volume antes de ser disponibilizado novamente.

Quando um conjunto de dados é indisponibilizado dessa maneira, o mapa de espaço não é salvo, portanto, o processamento de reinicialização da conexão do gerenciador de filas precisará varrer a estrutura do recurso de acoplamento para localizar mensagens no conjunto de dados e reconstruir o mapa de espaço antes de o conjunto de dados poder ser disponibilizado novamente. Como alternativa, se o conjunto de dados de mensagem compartilhada continuar utilizável, ele poderá ser indisponibilizado sutilmente por meio do uso do comando **RESET SMDS** para marcar o conjunto de dados **ACCESS (DISABLED)** até que ele esteja pronto para ser disponibilizado novamente.

Conteúdo do conjunto de dados está incorreto

O gerenciador de filas não pode detectar diretamente se um conjunto de dados contém dados incorretos ou não está atualizado, por exemplo, porque um volume incluindo esse conjunto de dados tinha que ser restaurado a partir dos backups. Porém, ele faz verificações de integridade que mostram que é improvável que esses erros façam com que dados de mensagens incorretos sejam vistos por programas de aplicativo.

Para propósitos de verificação de integridade, cada bloco de mensagens no conjunto de dados é prefixado com uma cópia do ID de entrada do recurso de acoplamento correspondente, incluindo um registro de data e hora exclusivo, que é verificado sempre que o bloco de mensagens é lido, antes de os dados da mensagem serem passados para o programa do usuário. Se o prefixo do bloco

de mensagens não corresponder ao ID de entrada (e a entrada do recurso de acoplamento não foi excluída no tempo médio), o bloco de mensagens será considerado danificado ou inutilizável.

Se a mensagem danificada for persistente, o conjunto de dados será marcado como **STATUS (FAILED)** e o conteúdo da estrutura deverá ser recuperado com o uso do comando **RECOVER CFSTRUCT**. Se a mensagem danificada for não persistente, não há como recuperá-la, portanto uma mensagem de diagnóstico será emitida e a entrada da mensagem do recurso de acoplamento correspondente será excluída.

Se nenhum mapa de espaço salvo estiver disponível quando o conjunto de dados for aberto, ele será reconstruído por meio da varredura da estrutura do recurso de acoplamento em busca de referências a dados no conjunto de dados. Durante esta varredura, o gerenciador de filas executa várias ações:

1. O gerenciador de filas determina o local da mensagem mais recente (se houver) restante atualmente no conjunto de dados.
2. O gerenciador de filas e, em seguida, lê essa mensagem a partir do conjunto de dados para assegurar que o prefixo de bloqueio corresponda ao ID de entrada da mensagem

Estas ações asseguram que o gerenciador de filas detecte qualquer caso em que o conjunto de dados é de nível inferior e marca o conjunto de dados como FAILED. Essa verificação, no entanto, tolera casos em que o conjunto de dados foi restaurado de uma cópia anterior e nenhuma mensagem nova foi incluída desde então ou em que todas as mensagens incluídas desde essa cópia foram lidas e excluídas subsequentemente.

Para proteger contra dados de nível inferior no caso em que o conjunto de dados foi encerrado normalmente, o gerenciador de filas executa várias ações:

1. O gerenciador de filas salva uma cópia do registro de data e hora do mapa de espaço no objeto SMDS dentro do Db2 quando o conjunto de dados é encerrado normalmente.
2. O gerenciador de filas e, em seguida, verifica se o registro de data e hora do mapa de espaço é o mesmo, quando o conjunto de dados é aberto novamente

Se o registro de data e hora não corresponder, isso sugere que uma cópia de nível inferior do conjunto de dados pode ter sido usada, portanto, o gerenciador de filas ignora o mapa de espaço existente e o reconstrói; essa ação só será bem-sucedida se nenhum dado da mensagem tiver sido perdido.

Nota: Essas verificações de integridade não garantem a detecção de um conjunto de dados danificado ou de nível inferior em todos os casos teoricamente possíveis. Por exemplo, elas não detectarão um caso no qual o início de um bloco de mensagens é válido, mas o restante dos dados foi parcialmente sobrescrito.

Cenários de Recuperação para Conjuntos de Dados de Mensagem Compartilhada

Esta seção descreve cenários de recuperação de conjuntos de dados de mensagem compartilhada.

Recuperação de conjunto de dados onde não há dados perdidos

Em alguns casos, o conteúdo correto de um conjunto de dados com falha pode ser restaurado sem a necessidade de uma recuperação real. Um exemplo é quando um conjunto de dados contém dados residuais de um uso anterior e não é pré-formatado novamente, o que pode ser corrigido com uma pré-formatação. Outro caso é quando um conjunto de dados foi movido, mas ocorreu um erro no processo de cópia dos dados, o que pode ser corrigido copiando os dados novamente de forma correta.

Nesses casos, o conjunto de dados corrigido pode ser disponibilizado novamente usando o comando **RESET SMDS** para configurar **STATUS (RECOVERED)**. Se a disponibilidade for atualmente **ACCESS (SUSPENDED)**, isso o configurará automaticamente de volta para **ACCESS (ENABLED)**.

Quando o gerenciador de filas proprietário é notificado de que o conjunto de dados foi recuperado, ele varre o conteúdo da estrutura para reconstruir o mapa de espaço, em seguida, muda o status para **STATUS (ACTIVE)**. Os outros gerenciadores de filas poderão começar a ler o conjunto de dados novamente.

Recuperação do conjunto de dados com TYPE(NORMAL)

Se o conteúdo de um conjunto de dados foi perdido, mas a estrutura do aplicativo foi definida com **RECOVER (YES)** e os logs de recuperação apropriados estiverem disponíveis, o comando **RECOVER CFSTRUCT** poderá ser usado para recuperar quaisquer mensagens persistentes armazenadas na estrutura, incluindo dados da mensagem persistente transferidos para conjuntos de dados de mensagem compartilhada. Esse comando restaura o estado atual usando informações registradas pelo comando **BACKUP CFSTRUCT**, mais todas as mudanças registradas em mensagens persistentes desde o momento do backup.

O comando **RECOVER CFSTRUCT** sempre recupera todas as mensagens persistentes na estrutura do recurso de acoplamento junto com os dados da mensagem transferidos armazenados no Db2. Para dados transferidos armazenados em conjuntos de dados de mensagens compartilhadas, cada conjunto de dados será selecionado para o processamento de recuperação somente se ele já estiver marcado como **STATUS (FAILED)** ou se for encontrado inesperadamente vazio ou inválido quando aberto pelo processamento de recuperação. Nenhum conjunto de dados de mensagem compartilhada marcado como ativo e que passar pelas verificações de validação precisará ser recuperado, já que os dados da mensagem existentes já estão corretos, mas o cabeçalho será atualizado para indicar que quaisquer mapas de espaço salvos precisarão ser reconstruídos após a recuperação.

O processamento de recuperação só é possível quando a estrutura é marcada como com falha, já que o conteúdo completo da estrutura precisa ser reconstruído pelo processamento de recuperação. No entanto, se pelo menos um conjunto de dados de mensagem compartilhada tiver sido marcado como com falha, o comando **RECOVER CFSTRUCT** marcará automaticamente a estrutura como com falha, se necessário, para permitir que o processamento de recuperação continue.

A recuperação pode ser executada por meio de qualquer gerenciador de filas no grupo de filas compartilhadas, contanto que o acesso de gravação tenha sido fornecido para os conjuntos de dados relevantes.

Apenas mensagens persistentes são submetidas a backup e registradas, portanto, um processamento de recuperação normal restaurará todas as mensagens persistentes, mas fará com que mensagens não persistentes na estrutura sejam perdidas.

Quando a recuperação tiver sido concluída, qualquer conjunto de dados selecionado para recuperação será mudado automaticamente para **STATUS (RECOVERED)** e, se a disponibilidade foi **ACCESS (SUSPENDED)**, ele será mudado para **ACCESS (ENABLED)**. O gerenciador de filas reconstrói o mapa de espaço para cada conjunto de dados varrendo as mensagens no recurso de acoplamento, em seguida, marca o conjunto de dados como **STATUS (ACTIVE)** para que ele possa ser usado novamente.

Recuperação do conjunto de dados com TYPE(PURGE)

Para uma estrutura recuperável, se o conteúdo do conjunto de dados tiver sido perdido, mas a recuperação não for possível por alguma razão, por exemplo, porque os logs de recuperação não estão disponíveis ou porque a recuperação demoraria muito tempo, o comando **RECOVER CFSTRUCT** poderá ser usado com **TYPE (PURGE)** para que a estrutura volte para o estado utilizável. Isso reconfigura a estrutura para o estado vazio e marca todos os conjuntos de dados associados como **STATUS (EMPTY)**.

Excluindo a estrutura do aplicativo

Se uma estrutura de aplicativo não recuperável for excluída usando o comando MVS **SETXCF FORCE** ou como resultado de falha de estrutura, na próxima vez em que a estrutura for conectada, a mensagem CSQE028I será emitida para informar que a estrutura foi reconfigurada e todas as mensagens existentes foram descartadas e quaisquer conjuntos de dados existentes serão automaticamente reconfigurados para **STATUS (EMPTY)** também. Essa ação torna uma estrutura não recuperável utilizável novamente após uma perda de dados na estrutura ou em qualquer um dos conjuntos de dados associados.

Se uma estrutura de aplicativo recuperável for excluída, ela será tratada como se a estrutura tivesse falhado.

Falhas de recuperação de conjunto de dados

Se **RECOVER CFSTRUCT** não puder ser concluído por alguma razão, por exemplo, porque um conjunto de dados de log não está mais disponível ou porque o gerenciador de fila foi finalizado enquanto a recuperação estava em andamento, então qualquer conjunto de dados para o qual a recuperação foi pelo menos iniciada será marcado no cabeçalho para mostrar que a recuperação parcial foi tentada e o conjunto de dados será deixado no estado **STATUS (FAILED)** .

Nesse caso, as opções são repetir a solicitação de recuperação original ou se recuperar com **TYPE (PURGE)**, descartando os dados existentes.

Se for feita uma tentativa de marcar o conjunto de dados como **STATUS (RECOVERED)** sem realmente recuperá-lo, na próxima vez que ele for aberto, o gerenciador de filas verá que o cabeçalho indica recuperação incompleta e o marcará como **STATUS (FAILED)** novamente.

Recuperação de desastre externa

Para recuperação de desastre externa, mensagens compartilhadas persistentes podem ser recriadas usando somente os logs e os objetos compartilhados do Db2 contendo as definições de CFSTRUCT e as informações de status do SMDS associado.

Após configurar as tabelas do Db2 contendo as definições, a estrutura do aplicativo e os conjuntos de dados de mensagem compartilhada podem ser configurados como vazios. Quando um gerenciador de filas se conecta a eles e descobre que eles ficam vazios inesperadamente, ele os marca como com falha; depois disso, um comando **RECOVER CFSTRUCT** único pode ser usado para recuperar todas as mensagens persistentes para todas as estruturas afetadas.

Comandos Relacionados ao SMDS

Este tópico descreve e fornece acesso aos comandos relacionados aos Shared Message Data Sets.

Exibir e alterar as opções **CFSTRUCT** relacionadas a grandes regras de transferência de mensagens (**OFFLOAD** e regras de transferência) e conjuntos de dados de mensagens compartilhadas (**DSGROUP**, **DSBLOCK**, **DSBUFS**, **DSEXPAND**):

- [DISPLAY CFSTRUCT](#)
- [DEFINE CFSTRUCT](#)
- [ALTER CFSTRUCT](#)
- [DELETE CFSTRUCT](#)

Exiba o status de **CFSTRUCT** relacionado à transferência de mensagens grandes (**OFFLDUSE**):

- [DISPLAY CFSTATUS](#)

Exiba e altere opções do conjunto de dados de substituição (**DSEXPAND** e **DSBUFS**) para gerenciadores de filas individuais:

- [DISPLAY SMDS](#)
- [ALTER SMDS](#)

Exiba ou modifique o status e a disponibilidade dos conjuntos de dados dentro do grupo de filas compartilhadas:

- [DISPLAY CFSTATUS TYPE\(SMDS\)](#)
- [RESET SMDS](#)

Exiba informações de uso do espaço do conjunto de dados SMDS e de uso do buffer para um gerenciador de filas:

- [DISPLAY USAGE TYPE\(SMDS\)](#)

Exiba ou modifique o status e a disponibilidade das conexões (**SMDSCONN**) com os conjuntos de dados de um gerenciador de filas individual:

- [DISPLAY SMDSCONN](#)

- [START SMDSCONN](#)
- [STOP SMDSCONN](#)

Faça backup e recupere mensagens compartilhadas, incluindo dados de mensagens grandes no SMDS quando necessário:

- [BACKUP CFSTRUCT](#)
- [RECOVER CFSTRUCT](#)

Vantagens de Usar Filas Compartilhadas

A fila compartilhada permite que aplicativos IBM MQ sejam escaláveis, altamente disponíveis e permite que o balanceamento de carga seja implementado.

Vantagens das filas compartilhadas

A arquitetura da fila compartilhada, em que os servidores clonados inserem o trabalho a partir de uma única fila compartilhada, tem algumas propriedades úteis:

- Ela é escalável por meio da inclusão de novas instâncias do aplicativo do servidor ou mesmo de uma nova imagem do z/OS com um gerenciador de filas (no grupo de filas compartilhadas) e uma cópia do aplicativo.
- Está altamente disponível.
- Ela executa naturalmente balanceamento de carga de trabalho *pull*, com base na capacidade de processamento disponível de cada gerenciador de filas no grupo de filas compartilhadas.

Usando Filas Compartilhadas para Alta Disponibilidade

Os exemplos a seguir ilustram como é possível usar uma fila compartilhada para aumentar a disponibilidade do aplicativo.

Considere um cenário do IBM MQ onde os aplicativos clientes sendo executados na rede desejam fazer as solicitações dos aplicativos do servidor em execução no z/OS. O aplicativo cliente constrói uma mensagem de solicitação e a coloca em uma fila de solicitações. Em seguida, o cliente espera por uma resposta a partir do servidor, enviado para a fila de resposta denominada no descritor de mensagens da mensagem de solicitação.

O IBM MQ gerencia o transporte da mensagem de solicitação do computador cliente para a fila de entrada do servidor no z/OS e da resposta do servidor de volta para o cliente. Ao definir a fila de entrada do servidor como uma fila compartilhada, qualquer mensagem colocada na fila pode ser recuperada em qualquer gerenciador de filas no grupo de filas compartilhadas. Isso significa que é possível configurar um gerenciador de filas em cada imagem do z/OS no sysplex e, conectando-os todos ao mesmo grupo de filas compartilhadas, qualquer um deles pode acessar mensagens na fila de entrada do servidor.

As mensagens na fila de entrada do servidor ainda ficam disponíveis, mesmo se um dos gerenciadores de fila for terminado de maneira anormal ou se você pará-lo por razões administrativas. É possível colocar imagem a inteira do z/OS offline e as mensagens ainda ficarão disponíveis.

Para tirar proveito dessa disponibilidade de mensagens em uma fila compartilhada, execute uma instância do aplicativo do servidor em cada imagem do z/OS no sysplex para fornecer maior capacidade de aplicativo do servidor e disponibilidade, conforme mostrado em [Figura 60 na página 195](#).

Uma instância do aplicativo do servidor recupera uma mensagem de solicitação da fila compartilhada e, com base no conteúdo, executa seu processamento, produzindo um resultado que é enviado de volta ao cliente como uma mensagem do IBM MQ. A mensagem de resposta é destinada para a fila de resposta e o gerenciador de filas de respostas denominado no descritor de mensagens da mensagem de solicitação.

Existem inúmeras opções que você pode usar para configurar o caminho de retorno. Para obter informações adicionais sobre essas opções, consulte [“Enfileiramento distribuído e grupos de filas compartilhadas”](#) na página 215.

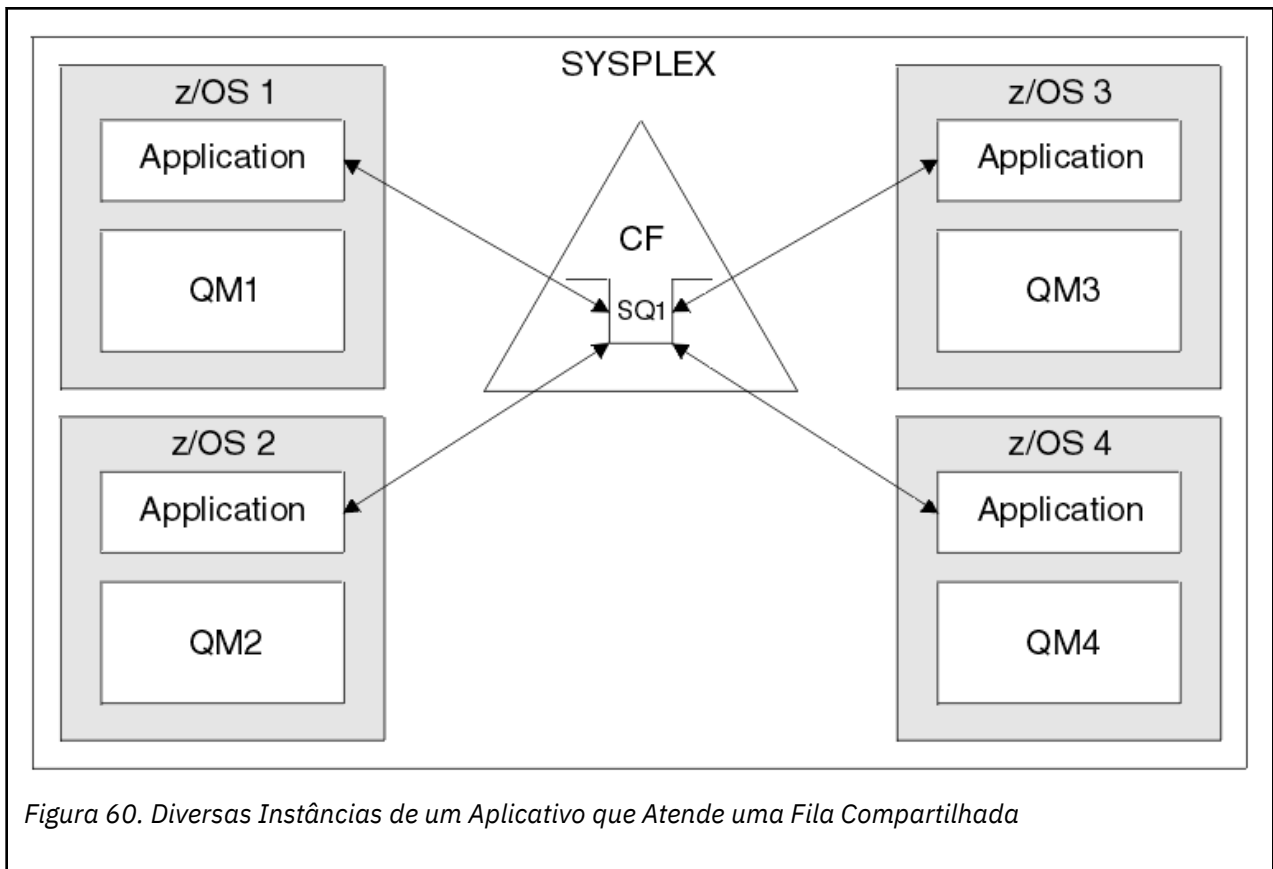


Figura 60. Diversas Instâncias de um Aplicativo que Atende uma Fila Compartilhada

Recuperação Peer-to-peer

Quando possível, para aprimorar ainda mais a disponibilidade de mensagens em um grupo de filas compartilhadas, o IBM MQ detectará se outro gerenciador de filas no grupo se desconectar anormalmente do recurso de acoplamento e concluirá as unidades de trabalho para esse gerenciador de filas que ainda estiverem pendentes. Esse recurso é conhecido como *recuperação peer-to-peer*.

Imagine que um gerenciador de filas seja finalizado de maneira anormal em um ponto no qual um aplicativo tenha recuperado uma mensagem de solicitação a partir de uma fila no ponto de sincronização, mas ainda não tenha colocado a mensagem de resposta ou confirmado a unidade de trabalho. Outro gerenciador de filas no grupo de filas compartilhadas detecta a falha e faz o backup das unidades de trabalho em andamento sendo executadas no gerenciador de filas com falha. Isso significa que a mensagem de solicitação é colocada de volta na fila de solicitações e fica disponível para uma das outras instâncias do servidor processar, sem esperar que o gerenciador de filas com falha seja reiniciado.

Se o IBM MQ não puder resolver uma unidade de trabalho automaticamente, será possível resolver a parte compartilhada manualmente para ativar outro gerenciador de filas no grupo de filas compartilhadas para continuar o processamento desse trabalho.

► z/OS **Uso da memória de classe de armazenamento com filas compartilhadas**

O uso da memória de classe de armazenamento (SCM) pode ser vantajoso quando usada com filas compartilhadas do IBM MQ for z/OS.

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

As máquinas z13, zEC12 e zBC12 permitem a instalação de cartões Flash Express. Esses cartões contêm unidades de estado sólido (SSD) flash. Após a instalação, o armazenamento flash dos cartões pode ser alocado para um ou mais LPARs em que ele é tipicamente conhecido como SCM.

O SCM fica entre o armazenamento real e o dispositivo de armazenamento de acesso direto (DASD) em termos de latência de E/S e custo. Como o SCM não tem partes móveis, ele exibe latências de E/S bem menores do que o DASD.

O SCM é também muito mais barato do que o armazenamento real. Como resultado, uma grande quantidade de armazenamento pode ser instalada para um custo relativamente baixo; por exemplo, um par de cartões Express Flash contém 1424 GB de armazenamento utilizável.

Essas características significam que o SCM é útil quando um grande volume de dados deve ser tirado do armazenamento real em um curto período de tempo, porque esses dados podem ser gravados no SCM muito mais rapidamente do que pode ser gravado no DASD. Esse ponto específico pode ser muito útil ao usar estruturas de lista de recurso de acoplamento (CF) contendo filas compartilhadas do IBM MQ.

Por que as estruturas de lista são preenchidas

Quando uma estrutura CF é definida, ela é configurada com um atributo SIZE que descreve o tamanho máximo da estrutura. Como as estruturas CF sempre são permanentemente residentes no armazenamento real, a soma dos atributos SIZE das estruturas que são definidas em um CF deve ser menor que a quantidade do armazenamento real alocado para o CF.

Como resultado, há uma pressão constante para manter o valor SIZE de qualquer estrutura determinada para o valor mínimo possível, de modo que mais estruturas possam se ajustar ao CF. No entanto, assegurar que as estruturas sejam grandes o suficiente para atingir seu propósito pode resultar em uma pressão conflitante, porque tornar uma estrutura muito pequena significa que ela poderá ser preenchida, prejudicando aplicativos ou subsistemas que a utilizam.

Há uma forte necessidade de dimensionar com exatidão uma estrutura com base em seu uso esperado. No entanto, essa tarefa é difícil de ser executada porque as cargas de trabalho podem mudar ao longo do tempo e a contabilidade de suas flutuações não é fácil.

As filas compartilhadas do IBM MQ usam estruturas de lista CF para armazenar mensagens. O IBM MQ chama as estruturas CF, que contêm mensagens e estruturas de aplicativos.

As estruturas de aplicativos são referenciadas usando as informações armazenadas em objetos CFSTRUCT do IBM MQ. Quando uma mensagem menor que 63 KB é colocada em uma fila compartilhada, a mensagem é armazenada inteiramente em uma estrutura de aplicativo como uma entrada de lista única e zero ou mais elementos de lista.

Como as filas compartilhadas do IBM MQ usam estruturas de lista, as pressões descritas também afetam as filas compartilhadas. Nesse caso, o número máximo de mensagens que podem ser armazenadas em uma fila compartilhada é uma função do:

- Tamanho das mensagens na fila
- Tamanho máximo da estrutura
- Número de entradas e elementos disponíveis na estrutura

Como até 512 filas compartilhadas podem usar a mesma estrutura, e competir efetivamente pelas entradas e elementos, isso complica ainda mais as coisas

As filas do IBM MQ são usadas para a transferência de dados entre aplicativos, de modo que uma situação comum é um aplicativo que coloca mensagens em uma fila, quando o aplicativo parceiro, que deve receber essas mensagens, não está em execução.

Quando isso acontece, o número de mensagens na fila aumenta com o tempo até que uma ou mais das situações a seguir ocorrem:

- O aplicativo de colocação para de colocar mensagens.
- O aplicativo de obtenção começa a obter mensagens.
- As mensagens existentes no início da fila expiram e são removidas da fila.

- A profundidade máxima da fila é atingida; nesse caso, um código de razão MQRC_Q_FULL é retornado ao aplicativo de colocação.
- A estrutura que contém a fila compartilhada atinge seu tamanho máximo, ou o CF que contém a estrutura fica sem armazenamento disponível. Em qualquer caso, um código de razão MQRC_STORAGE_MEDIUM_FULL é retornado ao aplicativo de colocação.

Nos últimos três casos, a fila está cheia. Nesse ponto, o aplicativo de colocação tem um problema porque não existe lugar para onde ir suas mensagens. O aplicativo de colocação geralmente resolve esse problema usando uma ou mais das soluções a seguir:

- Tenta de novo repetidamente colocar a mensagem, opcionalmente com um atraso entre as novas tentativas.
- Coloca as mensagens em algum outro lugar, como um banco de dados ou um arquivo. As mensagens podem ser acessadas posteriormente e colocadas na fila da forma normal.
- Descarta a mensagem se ela for não persistente.

No entanto, para algumas classes de aplicativos, por exemplo, aquelas com um grande volume de mensagens recebidas, ou sem acesso a um sistema de arquivos, essas soluções não são práticas. Há uma necessidade real de assegurar que as filas nunca, ou é altamente improvável que, sejam preenchidas em primeiro lugar e isso é pertinente especialmente para filas compartilhadas.

SMDS e regras de transferência

As regras de transferência introduzidas no IBM WebSphere MQ 7.1 fornecem uma maneira de reduzir a probabilidade de uma estrutura de aplicativo ser preenchida.

Cada estrutura de aplicativo tem três regras associadas a ela, especificadas usando três pares de palavras-chave:

- OFFLD1SZ e OFFLD1TH
- OFFLD2SZ e OFFLD2TH
- OFFLD3SZ e OFFLD3TH

Cada regra especifica as condições que devem ser atendidas para que dados da mensagem sejam transferidos para o mecanismo de armazenamento que está associado à estrutura do aplicativo. Dois tipos de mecanismos de armazenamento estão disponíveis atualmente:

- Db2
- Grupo de conjuntos de dados lineares Virtual Storage Access Method (VSAM), que o IBM MQ chama de conjunto de dados de mensagens compartilhadas (SMDS).

O exemplo a seguir mostra o comando MQSC para criar uma estrutura de aplicativo denominada LIST1, usando o comando DEFINE CFSTRUCT.

Essa estrutura tem as regras de transferência padrão estabelecidas e usa o SMDS como o mecanismo de transferência. Isso significa que quando a estrutura está 70% cheia (OFFLD1TH), todas as mensagens com 32 KB ou mais (OFFLD1SZ) são transferidos para SMDS.

Da mesma forma, quando a estrutura está 80% cheia (OFFLD2TH) todas as mensagens com 4 KB ou mais (OFFLD2SZ) são transferidas. Quando a estrutura está 90% cheia (OFFLD3TH) todas as mensagens (OFFLD3SZ) são transferidas.

```
DEFINE CFSTRUCT(LIST1)
CFLEVEL(5)
OFFLOAD(SMDS)
OFFLD1SZ(32K) OFFLD1TH(70)
OFFLD2SZ(4K) OFFLD2TH(80)
OFFLD3SZ(0K) OFFLD3TH(90)
```

Uma mensagem transferida é armazenada na mídia de transferência e um ponteiro para a mensagem é armazenado na estrutura. Embora as regras de transferência reduzam a chance do preenchimento da estrutura, colocando menos dados de mensagens na estrutura conforme ela fica sem armazenamento,

alguns dados ainda são gravados na estrutura para cada mensagem. Ou seja, o ponteiro para a mensagem transferida.

Além disso, as regras de transferência são fornecidos com um custo de desempenho. Gravar uma mensagem para uma estrutura é relativamente rápido e é amplamente dominado pelo tempo gasto para enviar a solicitação da gravação para o CF. A gravação real na estrutura é rápida, acontecendo em velocidades de armazenamento real.

A gravação de uma mensagem no SMDS é muito mais lenta porque ela inclui a gravação na estrutura para o ponteiro da mensagem e a gravação dos dados da mensagem no SMDS. Essa segunda operação de gravação é feita na velocidade do DASD e tem o potencial de incluir latência. Se o Db2 for usado como o mecanismo de transferência, o custo de desempenho será muito maior.

 *Como a memória de classe de armazenamento funciona com o IBM MQ for z/OS*

Uma visão geral do uso da memória de classe de armazenamento (SCM) com filas compartilhadas do IBM MQ for z/OS.

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

Um recurso de acoplamento (CF) que está em CFLEVEL 19, ou maior, pode ter SCM alocado para ele. As estruturas definidas nesse CF então podem ser configuradas para fazer uso do SCM para reduzir as chances de preenchimento de estruturas (conhecido como condição de estrutura cheia). Quando uma estrutura configurada para fazer uso do SCM é preenchida passando de um ponto determinado pelo sistema, o CF começa a mover os dados da estrutura para o SCM, o que libera espaço na estrutura para novos dados.

Nota: Como o próprio SCM pode ser preenchido, a alocação de SCM para uma estrutura só reduz a probabilidade de uma condição de estrutura cheia, mas não elimina totalmente a chance de uma ocorrência.

Uma estrutura é configurada para usar SCM especificado as palavras-chave **SCMALGORITHM** e **SCMMAXSIZE** na política do Coupling Facility Resource Manager (CFRM), contendo a definição dessa estrutura.

Observe que após essas palavras-chave serem especificadas, e a política CFRM ser aplicada, a estrutura deve ser reconstruída, ou desalocada para que possam tomar efeito.

Palavra-chave SCMALGORITHM

Como a velocidade de entrada/saída do SCM é mais lenta que a do armazenamento real, o CF usa um algoritmo customizado para o uso esperado da estrutura a fim de reduzir o impacto de gravação ou leitura no SCM.

O algoritmo é configurado pela palavra-chave **SCMALGORITHM** na política CFRM para a estrutura, usando o valor **KEYPRIORITY1**. Observe que é necessário usar o valor **KEYPRIORITY1** somente com estruturas de lista usadas pelas filas compartilhadas do IBM MQ.

O algoritmo **KEYPRIORITY1** funciona supondo que a maioria dos aplicativos obterá mensagens de uma fila compartilhada em ordem de prioridade; ou seja, quando um aplicativo obtém uma mensagem, ele obtém a mensagem mais antiga com a prioridade mais alta.

Quando uma estrutura começa a ser preenchida passando do limite definido pelo sistema de 90%, o CF começa a migrar de forma assíncrona as mensagens com a menor probabilidade de serem as próximas obtidas. Essas são as mensagens com prioridade mais baixa que foram mais recentemente colocadas na fila.

Essa migração assíncrona de mensagens da estrutura para o SCM é conhecida como "pré-migração".

A pré-migração reduz o custo de desempenho do uso de SCM porque reduz a chance de um aplicativo ser bloqueado durante a ocorrência de entrada/saída síncrona para SCM.

Além da pré-migração, o algoritmo *KEYPRIORITY1* também traz de volta de forma assíncrona mensagens do SCM para a estrutura quando espaço livre suficiente está disponível. Para o algoritmo *KEYPRIORITY1*, isso significa que quando a estrutura é menor ou igual a 70% cheia.

O ato de trazer as mensagens do SCM para a estrutura é conhecido como "pré-busca".

A pré-busca reduz a probabilidade de um aplicativo tentar obter uma mensagem que foi pré-migrada para o SCM e ter de aguardar enquanto o CF traz de volta de maneira síncrona a mensagem para a estrutura.

Palavra-chave SCMMAXSIZE

A palavra-chave **SCMMAXSIZE** define a quantidade máxima de SCM que pode ser usada por uma estrutura. Como o SCM é alocado para a estrutura pelo CF quando é necessário, é possível especificar um **SCMMAXSIZE** que é maior que a quantidade total de SCM livre disponível. Isso é conhecido como "supercomprometimento".

Importante: Nunca supercomprometa o SCM. Se você fizer isso, os aplicativos que dependem dele não obterão o comportamento que esperam. Por exemplo, aplicativos IBM MQ que usam filas compartilhadas podem obter códigos de razão MQRD_STORAGE_MEDIUM_FULL inesperados.

O CF usa várias estruturas de dados para controlar seu uso de SCM. Essas estruturas de dados residem no armazenamento real que é alocado para o CF e, como resultado, reduzem a quantidade de armazenamento real que pode ser utilizado por estruturas. O armazenamento usado por essas estruturas de dados é conhecido como "espaço aumentado".

Quando uma estrutura é configurada com o SCM, uma pequena quantidade de armazenamento real é alocada do CF para a estrutura, conhecida como espaço fixo aumentado. Esta é alocada mesmo que a estrutura nunca realmente use qualquer SCM. Como os dados da estrutura são armazenados no SCM, espaço aumentado adicional dinâmico será alocado do armazenamento real sobressalente no CF.

Quando os dados são removidos do SCM, o espaço aumentado dinâmico é retornado para o CF. O espaço aumentado, seja fixo ou dinâmico, nunca é obtido do armazenamento real que é alocado para uma estrutura.

Além de armazenamento aumentado, quando uma estrutura está configurada para usar o SCM, a quantidade de armazenamento de controle usada por essa estrutura aumenta. Isso significa que uma estrutura de lista configurada com SCM pode conter menos entradas e elementos do que uma estrutura do mesmo tamanho sem SCM sendo configurado.

Para entender o impacto do SCM em estruturas novas ou existentes, use a ferramenta [CFSizer](#).

Um ponto importante final a ser observado é que depois que os dados são movidos da estrutura para o SCM e espaço aumentado dinâmico é utilizado, a estrutura não pode ser alterada seja manual ou automaticamente.

Ou seja, a quantidade de armazenamento alocado para a estrutura não pode ser aumentada ou diminuída, a proporção de entrada para elemento que é usada pela estrutura não pode ser alterada, e assim por diante. Para tornar alterável a estrutura novamente, ela não deve ter quaisquer dados armazenados no SCM e não deve usar armazenamento aumentado dinâmico.

Por que usar SCM

Armazenamento emergencial e desempenho melhorado são dois casos de uso do SCM com o IBM MQ for z/OS.

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

Esta seção apresenta a teoria por trás dos dois cenários possíveis. Para obter detalhes adicionais sobre como configurar os cenários, consulte:

- [“Armazenamento emergencial - configuração básica” na página 203](#)
- [“Melhor desempenho – configuração básica” na página 209](#)

Importante: O uso de SCM com estruturas de CF não é dependente de qualquer versão específica do IBM MQ. No entanto, o cenário de armazenamento emergencial funciona somente com o IBM WebSphere MQ 7.1 e mais recente, porque ele requer SMDS e as regras de transferência.

Armazenamento emergencial

SMDS e transferência de mensagem podem ser usados em conjunto com o SCM para reduzir a probabilidade de um código de razão MQRD_STORAGE_MEDIUM_FULL sendo retornado para um aplicativo IBM MQ durante uma indisponibilidade estendida.

Visão geral

Uma única fila compartilhada é configurada em uma estrutura de aplicativo. O aplicativo de colocação coloca mensagens na fila compartilhada; o aplicativo de obtenção obtém as mensagens da fila compartilhada.

Durante a execução normal, espera-se que a profundidade da fila seja próxima de zero, mas uma necessidade de negócios indica que o sistema deve ser capaz de tolerar uma indisponibilidade de duas horas do aplicativo de obtenção. Isso significa que a fila compartilhada deve ser capaz de conter duas horas de mensagens do aplicativo de colocação.

Atualmente, esse processo é obtido usando as regras de transferência padrão, e SMDS, de modo que o tamanho da estrutura é minimizado, reduzindo o custo de desempenho associado à transferência.

A taxa das mensagens sendo enviadas para a fila compartilhada deverá dobrar no curto a médio prazo. Embora o requisito de que o sistema seja capaz de tolerar uma indisponibilidade ainda exista, armazenamento real disponível no CF não é suficiente para dobrar o tamanho da estrutura.

Como o CF, que contém a estrutura do aplicativo, reside em uma máquina zEC12, existe a possibilidade de associar SCM suficiente à estrutura para armazenar mensagens suficientes a fim de que uma indisponibilidade de duas horas possa ser tolerada.

Considere o que acontece durante um período de tempo:

1. Inicialmente, o sistema está em seu estado estável. Os aplicativos de colocação e obtenção estão sendo executados normalmente e a profundidade da fila está próxima de, ou em, zero. O resultado é que a estrutura do aplicativo está praticamente vazia.
2. Em um determinado momento, o aplicativo de obtenção sofre uma falha inesperada e é interrompido. O aplicativo de colocação continua a colocar mensagens na fila e a estrutura do aplicativo começa a ser preenchida.
3. Após a estrutura ficar 70% cheia, as condições da primeira regra de transferência são atendidas e todas as mensagens com um tamanho maior ou igual a 32 KB são transferidas para o SMDS.

Consulte [“SMDS e regras de transferência” na página 197](#) para obter uma visão geral das regras de transferência.

4. Como as mensagens continuam a ser colocadas na fila compartilhada, a estrutura continua a ser preenchida (por causa dos dados da mensagem sendo armazenados na estrutura, ou como resultado dos ponteiros para as mensagens transferidas sendo armazenadas na estrutura).

Quando a estrutura fica 80% cheia, a segunda regra de transferência começa a ser aplicada e as mensagens com 4 KB ou maiores são transferidas para o SMDS.

5. Quando a estrutura está mais de 90% cheia, todas as mensagens são transferidas para o SMDS e somente os ponteiros das mensagens estão sendo colocados na estrutura.

A essa altura, o algoritmo de pré-migração começa a ser executado e a mover os dados da estrutura para o SCM. Supondo que todas as mensagens na fila tenham a mesma prioridade, as mensagens mais recentes são pré-migradas.

Como todas as mensagens estão sendo agora transferidas para o SMDS, os dados que estão sendo movidos para o SCM não são dados de mensagens reais; em vez disso, são ponteiros para as mensagens no SMDS.

Como resultado, o número de mensagens que podem ser armazenadas na combinação da estrutura, e no SCM e SMDS associados à estrutura, é muito grande.

Desempenho: Durante essa fase da indisponibilidade, o aplicativo de colocação pode sofrer um grau de degradação de desempenho devido a ter de gravar no SMDS. Nesse caso, o uso de SCM não deve ser um fator limitante no aplicativo de colocação em termos de desempenho. O SCM fornece espaço extra para evitar o preenchimento da estrutura.

6. Eventualmente, o aplicativo de obtenção está disponível novamente e a indisponibilidade é encerrada.

No entanto, o SCM ainda está sendo usado pela estrutura. O aplicativo de obtenção começa a ler as mensagens da fila, obtendo as mais antigas, de prioridade mais alta primeiro.

Como essas mensagens foram gravadas antes de a estrutura começar a ser preenchida, elas resultam totalmente da parte do armazenamento real da estrutura.

7. Conforme a estrutura começa a ficar vazia, ela fica abaixo do limite no qual a pré-migração está ativa; assim, a pré-migração é interrompida.
8. O uso da estrutura é reduzido abaixo do ponto no qual as regras de transferência entram em vigor, de modo que as mensagens não são mais transferidas para o SMDS, a menos que tenham mais de 63 KB.

A essa altura, o algoritmo de pré-migração começa a mover os dados do SCM para a estrutura. Como o aplicativo de obtenção obtém as mensagens da fila na ordem esperada pelo algoritmos SCM, as mensagens são colocadas antes que o aplicativo de obtenção precise delas.

O resultado é que o aplicativo de obtenção nunca precisa aguardar para que as mensagens sejam colocadas em sincronia a partir do SCM.

9. Conforme o aplicativo de obtenção continua a se mover para baixo na fila, ele começa a recuperar as mensagens que foram transferidas para o SMDS.
10. Finalmente, o sistema está em um estado estável novamente. Nenhuma mensagem é armazenada no SCM ou SMDS e a profundidade da fila fica próxima de zero.

Desempenho melhorado

Esse cenário descreve o uso do SCM para aumentar o número de mensagens que podem ser armazenadas em uma fila compartilhada sem incorrer no custo de desempenho do uso do SMDS.

Descrição

Para esse cenário, um aplicativo de colocação e obtenção se comunica por meio de uma fila compartilhada que é armazenada na estrutura do aplicativo.

O aplicativo de colocação tende a ser executado em bursts, quando ele coloca um grande número de mensagens em um curto período de tempo. Depois, em um período de tempo estendido, ele não produz nenhuma mensagem.

O aplicativo de obtenção processa sequencialmente cada mensagem e executa processamento complexo em cada um. Como resultado, a maior parte do tempo a profundidade da fila é zero, exceto quando o aplicativo de colocação começa a ser executado, onde a profundidade da fila começa a aumentar à medida que as mensagens vão sendo colocadas de forma mais rápida do que elas estão sendo obtidas.

A profundidade da fila aumenta até que o aplicativo de colocação é interrompido, e o aplicativo de obtenção tem tempo suficiente para processar todas as mensagens na fila.

Notas:

1. Nesse cenário, o fator-chave é o desempenho. As mensagens que estão sendo enviadas para a fila são sempre menores que 63 KB; assim, nunca precisam ser transferidas para SMDS.
2. A estrutura do aplicativo foi dimensionada para que seja grande o suficiente para conter todas as mensagens que serão colocadas nela pelo aplicativo de colocação em um único "burst".
3. As regras de transferência devem ser todas desativadas para que, mesmo quando a estrutura começar a ser preenchida, as mensagens não sejam transferidas para o SMDS. Isso ocorre porque os custos de desempenho que estão associados à gravação e leitura de mensagens no SMDS são considerados inaceitáveis.

Com o tempo, o número de mensagens que o aplicativo de colocação enviar em um burst deverá aumentar por várias ordens de magnitude. Como o aplicativo de obtenção deve processar cada mensagem sequencialmente, o número de mensagens na fila aumenta para o ponto em que a estrutura é preenchida.

Nesse ponto, o aplicativo de colocação recebe um código de razão (MQRC_STORAGE_MEDIUM_FULL) ao colocar uma mensagem e a operação de colocação falha. O aplicativo de colocação somente consegue tolerar brevemente períodos quando é incapaz de colocar mensagens na fila. Se o período for muito longo, o aplicativo será encerrado.

Supondo que você não tenha tempo, ou qualificações disponíveis, para regravar o aplicativo de colocação ou obtenção, esse problema terá três soluções possíveis:

1. Aumente o tamanho da estrutura do aplicativo.
2. Inclua regras de transferência na estrutura do aplicativo para que as mensagens sejam transferidas para o SMDS conforme a fila começa a ser preenchida.
3. Associe o SCM à estrutura.

A primeira solução é rápida de ser implementada, mas o armazenamento real disponível no CF não é suficiente.

A segunda solução também pode ser rápida de ser implementada, mas o impacto no desempenho de transferência para o SMDS é considerado muito significativo para usar essa opção.

A terceira solução, associando o SCM à estrutura, fornece um equilíbrio aceitável de custo e desempenho.

A associação do SCM a uma estrutura resulta em um uso maior do armazenamento real no CF devido ao armazenamento aumentado que as operações de obtenção usaram. No entanto, a quantidade real de armazenamento real será inferior à quantidade usada na primeira opção.

Outra consideração é o custo de SCM. No entanto, esse custo é muito mais barato do armazenamento real. Esses fatores se combinam para tornar a terceira opção mais barata do que a primeira.

Embora a terceira opção, potencialmente, possa não ser tão boa quanto a primeira, os algoritmos de pré-busca e pré-migração usados pelo CF podem ser combinados para tornar aceitáveis as diferenças em desempenho, ou negligenciáveis, em alguns casos.

Claro que o desempenho pode ser muito melhor do que usar SMDS para transferir mensagens.

Considere o que acontece durante um período de tempo:

1. Inicialmente, o aplicativo de obtenção está ativo e aguardando que as mensagens sejam entregues na fila compartilhada. O aplicativo de colocação não está ativo e a fila compartilhada está vazia.
2. Em um determinado momento, o aplicativo de colocação se torna ativo e começa a colocar um grande número de mensagens na fila compartilhada. O aplicativo de obtenção começa a obter as mensagens, mas a profundidade da fila começa a aumentar rapidamente, porque o aplicativo de obtenção é mais lento do que o aplicativo de colocação.

Como resultado, a estrutura do aplicativo começa a ser preenchida.

3. Conforme o tempo aumenta, o aplicativo de colocação ainda está ativo. A estrutura do aplicativo é preenchida até aproximadamente 90%.

Aqui é quando o algoritmo de pré-migração do SCM começa a mover as mensagens da estrutura para o SCM, liberando espaço na estrutura.

Como o aplicativo de obtenção obtém primeiro as mensagens mais antigas, de prioridade mais alta da fila, ele está sempre obtendo as mensagens da estrutura e não precisa esperar que as mensagens sejam colocadas de forma síncrona do SCM para a estrutura.

4. O aplicativo de colocação ainda está ativo e colocando mensagens na fila compartilhada. No entanto, o aplicativo nunca recebe um código de razão MQR_STORAGE_MEDIUM_FULL, porque existe espaço suficiente no SCM para armazenar todas as mensagens que não se ajustam à estrutura.
5. Eventualmente, o aplicativo de colocação é interrompido porque não tem mais nenhuma mensagem a ser colocada.

O algoritmo de pré-migração é interrompido porque a estrutura cai abaixo de 90% de uso e o aplicativo de obtenção continua processando as mensagens na fila.

6. Conforme o aplicativo de obtenção começa a liberar espaço na estrutura, o algoritmo de pré-busca começa a trazer as mensagens de volta do SCM para a estrutura.

Como o aplicativo de obtenção processa as mensagens na ordem esperada pelo algoritmo de pré-busca, o aplicativo de obtenção nunca é bloqueado enquanto aguarda que os dados da mensagem sejam colocados de forma síncrona do SCM para a estrutura.

7. Finalmente, o aplicativo de obtenção processa todas as mensagens na fila compartilhada e aguarda até que a próxima mensagem esteja disponível. A estrutura e o SCM ficam vazios de mensagens.

Armazenamento emergencial - configuração básica

Como você configura um cenário básico para armazenamento emergencial no IBM MQ.

Sobre esta tarefa

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

SMDS e transferência de mensagem podem ser usados em conjunto com o SCM para reduzir a probabilidade de um código de razão MQR_STORAGE_MEDIUM_FULL sendo retornado para um aplicativo IBM MQ durante uma indisponibilidade estendida.

Por exemplo, sua empresa tem um aplicativo que coloca mensagens na fila e um aplicativo que obtém mensagens da fila. Durante a execução normal, você espera que a profundidade da fila fique próxima de zero, mas uma necessidade de negócios indica que o sistema é capaz de tolerar uma indisponibilidade de duas horas do aplicativo que obtém as mensagens.

Isso significa que a fila compartilhada que está sendo usada deve ser capaz de conter duas horas de mensagens do aplicativo de colocação. Atualmente, você obtém isso usando as regras de transferência padrão e o SMDS.

Você espera que a taxa das mensagens que estão sendo enviadas para a fila compartilhada dobre no curto a médio prazo. Embora sua necessidade de que o sistema seja capaz de tolerar uma indisponibilidade de duas horas ainda exista, armazenamento real disponível no CF não é suficiente para dobrar o tamanho da estrutura. Como o CF que contém a estrutura do aplicativo reside em uma máquina zEC12, você tem a capacidade de associar SCM suficiente à estrutura para armazenar mensagens suficientes, de forma que uma indisponibilidade de duas horas possa ser tolerada.

Esse cenário inicial usa um:

- Grupo de filas compartilhadas, IBM1, que contém um único gerenciador de filas, CSQ3. Além da estrutura de administração, o grupo de filas compartilhadas definiu uma única estrutura de aplicativo, SCEN1.

- Recurso de acoplamento (CF) CF01, no qual a estrutura de aplicativo SCEN1 é armazenada como a estrutura IBM1SCEN1. Essa estrutura tem um tamanho máximo de 1 GB.
- Fila compartilhada única, SCEN1.Q, que a estrutura do aplicativo usa.

Essa configuração está ilustrada no [Figura 61](#) na página 204.

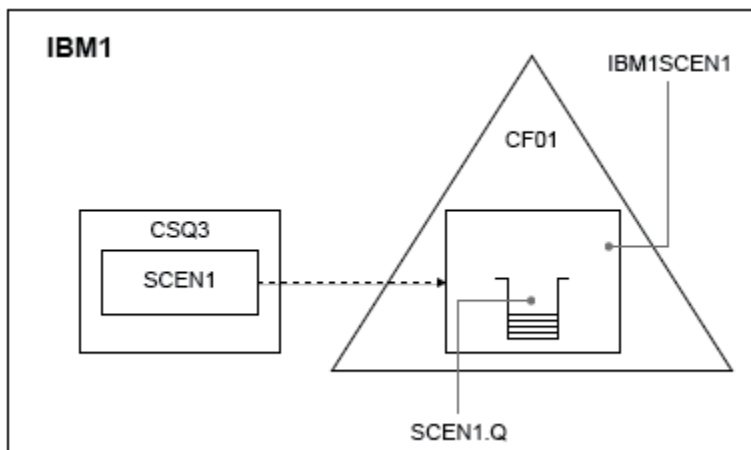


Figura 61. Configuração básica

Além disso, suponha que o gerenciador de filas CSQ3 já seja o único membro do grupo de filas compartilhadas IBM1.

Deve-se incluir a definição da estrutura IBM1SCEN1 na política do Coupling Facility Resource Manager (CFRM). Para simplificar, a estrutura é definida para que possa ser criada em apenas um único recurso de acoplamento, CF01, especificando PREFLIST (CF01).



Atenção: Para permitir alta disponibilidade em seu sistema de produção, é necessário incluir pelo menos dois CFs no PREFLIST para quaisquer estruturas que sejam usadas pelo IBM MQ.

Procedimento

1. Atualize a política CFRM usando o comando a seguir:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN1
```

Política de CFRM de amostra para a estrutura IBM1SCEN1:

```
STRUCTURE
NAME(IBM1SCEN1)
SIZE(1024M)
INITSIZE(512M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
```

2. Verifique se a estrutura foi criada corretamente, usando o comando a seguir:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Nesse ponto, sua estrutura não foi alocada ao grupo de compartilhamento de filas, conforme mostrado pela linha STATUS.

3. Configure o IBM MQ para usar a estrutura definida na política de CFRM.

- a. Use o comando `DEFINE CFSTRUCT`, com o nome da estrutura de SCEN1 para criar um objeto CFSTRUCT do IBM MQ:

```
DEFINE CFSTRUCT(SCEN1)
CFCONLOS(TOLERATE)
CFLEVEL(5)
DESCR('Structure for SCM scenario 1')
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFLD1SZ(64K) OFFLD1TH(70)
OFFLD2SZ(64K) OFFLD2TH(80)
OFFLD3SZ(64K) OFFLD3TH(90)
```

- b. Valide a estrutura, usando o comando `DISPLAY CFSTRUCT`.
- c. Defina a fila compartilhada SCEN1.Q, para usar a estrutura SCEN1, utilizando o comando MQSC a seguir:

```
DEFINE QLOCAL(SCEN1.Q) QSGDISP(SHARED) CFSTRUCT(SCEN1) MAXDEPTH(999999999)
```

4. Use o IBM MQ Explorer para colocar uma única mensagem na fila SCEN1.Q e tirar a mensagem novamente.
5. Emita o comando a seguir para verificar se a estrutura agora é alocada:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Faça check-in da saída do comando, que a linha STATUS mostra ALLOCATED.

Resultados

Você criou a configuração básica. Agora será possível ter uma ideia do desempenho de linha de base de sua configuração usando o método selecionado.

Como proceder a seguir

Incluir SMDS e SCM na estrutura inicial

Conceitos relacionados

[“Uso da memória de classe de armazenamento com filas compartilhadas” na página 195](#)

O uso da memória de classe de armazenamento (SCM) pode ser vantajoso quando usada com filas compartilhadas do IBM MQ for z/OS.

 *Incluindo SMDS e SCM na estrutura inicial*

Como você inclui SMDS e SCM para armazenamento emergencial no IBM MQ.

Sobre esta tarefa

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

Esta parte da tarefa usa a configuração básica descrita em [“Armazenamento emergencial - configuração básica” na página 203](#). O cenário descreve a inclusão de conjuntos de dados de mensagem compartilhada (SMDS) e, em seguida, de SCM na estrutura inicial.

Essa configuração final é ilustrada na [Figura 62 na página 206](#).

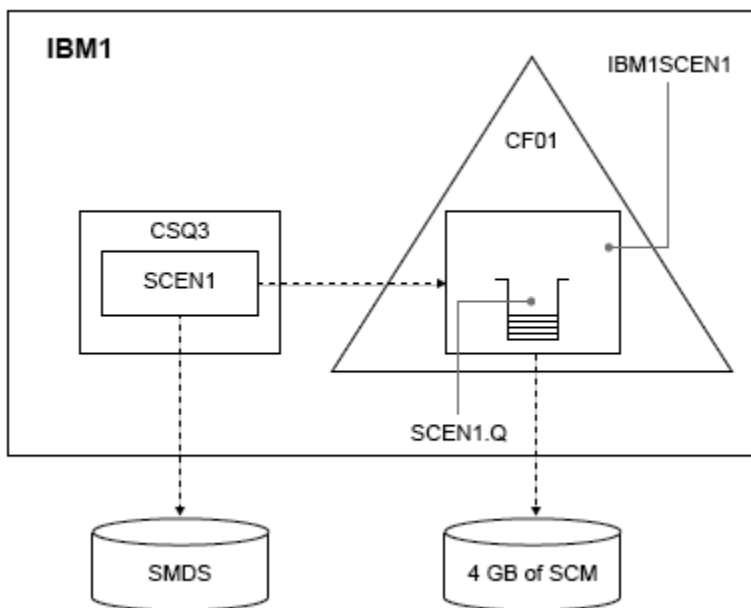


Figura 62. Configuração que inclui SMDS e SCM para armazenamento emergencial

Procedimento

1. Crie o conjunto de dados SMDS que a estrutura de aplicativo SCEN1 usa, editando a JCL de amostra **CSQ4SMDS**, conforme mostrado:

```
//CSQ4SMDS JOB NOTIFY=&SYSUID
//*
/* Allocate SMDS
/*
//DEFINE EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER          -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS) -
MEGABYTES(5000 3000) -
LINEAR                 -
SHAREOPTIONS(2 3) ) -
DATA                   -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS.DATA) )
/*
/*
/* Format the SMDS
/*
//FORM EXEC PGM=CSQJUFMT,COND=(0,NE),REGION=0M
//STEPLIB DD DSN=MQ800.SCSQANLE,DISP=SHR
// DD DSN=MQ800.SCSQAUTH,DISP=SHR
//SYSUT1 DD DISP=OLD,DSN=CSQSMDS.SCEN1.CSQ3.SMDS
//SYSPRINT DD SYSOUT=*
```

2. Emita o comando **ALTER CFSTRUCT** para mudar a estrutura do aplicativo SCEN1, para usar o SMDS para transferir e implementar as regras de transferência padrão:

```
ALTER CFSTRUCT(SCEN1) OFFLOAD(SMDS) OFFLD1SZ(32K) OFFLD2SZ(4K) OFFLD3SZ(0K)
DSGROUP('CSQSMDS.SCEN1.*.SMDS') DSBLOCK(1M)
```

Observe o seguinte :

- Como SCEN1.Q é a única fila compartilhada na estrutura de aplicativo SCEN1, o valor **DSBLOCK** foi configurado como 1M, o maior valor possível. Esta deve ser a configuração mais eficiente para nosso cenário.

- Como as mensagens enviadas pelo aplicativo de colocação são de 30 KB, transferência para SMDS não é iniciada até que a segunda regra de transferência seja atendida, quando a estrutura está 80% cheia.
3. Execute o aplicativo de teste novamente.
Observe o aumento do armazenamento de mensagens na fila.
 4. Inclua 4 GB de SCM na estrutura IBM1SCEN1, executando o procedimento a seguir:
 - a) Verifique quanto de SCM está instalado e alocado para CF01, emitindo o comando a seguir:

```
D CF,CFNAME=CF01
```

- b) Verifique as figuras STORAGE - CLASS MEMORY na seção STORAGE CONFIGURATION da saída exibida para ver o armazenamento disponível..
- c) Atualize a política de CFRM com as palavras-chave SCMMAXSIZE e SCMALGORITHM, conforme mostrado:

```
STRUCTURE
NAME(IBM1SCEN1)
SIZE(1024M)
INITSIZE(512M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
SCMMAXSIZE(4G)
SCMALGORITHM(KEYPRIORITY1)
```

5. Ative a política de CFRM emitindo o comando a seguir:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=polname
```

6. Reconstrua a estrutura IBM1SCEN1.

Deve-se executar este procedimento porque a estrutura foi alocada quando você fez as mudanças anteriores.

Emita o comando a seguir para reconstruir a estrutura:

```
SETXCF START,REBUILD,STRNM=IBM1SCEN1
```

Resultados

Você incluiu com êxito o SCM na sua configuração.

Como proceder a seguir

Otimize o desempenho do sistema. Consulte [“Otimizando o uso da memória de classe de armazenamento”](#) na página 207 para obter mais informações.

 *Otimizando o uso da memória de classe de armazenamento*

Como você melhora o uso da memória de classe de armazenamento (SCM).

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

Execute o seguinte comando:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Como a estrutura já estava cheia de dados de mensagens, devido aos testes anteriores, parte da reconstrução envolveu a pré-migração de algumas das mensagens da estrutura para SCM. Esse processo foi iniciado utilizando o comando anterior.

A saída desse comando produz, por exemplo:

```
ACTIVE STRUCTURE
-----
ALLOCATION TIME: 06/17/2014 09:28:50
CFNAME : CF01
COUPLING FACILITY: 002827.IBM.02.00000000B8D7
PARTITION: 3B CPCID: 00
STORAGE CONFIGURATION ALLOCATED MAXIMUM %
ACTUAL SIZE: 1024 M 1024 M 100
AUGMENTED SPACE: 3 M 142 M 2
STORAGE-CLASS MEMORY: 88 M 4096 M 2
ENTRIES: 120120 1089536 11
ELEMENTS: 240240 15664556 1
SPACE USAGE IN-USE TOTAL %
ENTRIES: 84921 219439 38
ELEMENTS: 2707678 3149050 85
EMCS: 2 282044 0
LOCKS: 1024
SCMHIGHTHRESHOLD : 90
SCMLOWTHRESHOLD : 70
ACTUAL SUBNOTIFYDELAY: 5000
PHYSICAL VERSION: CD5186A0 2BD8B85C
LOGICAL VERSION: CD515C50 CE2ED258
SYSTEM-MANAGED PROCESS LEVEL: 9
XCF GRPNAME : IXCL0053
DISPOSITION : KEEP
ACCESS TIME : NOLIMIT
MAX CONNECTIONS: 32
# CONNECTIONS : 1
CONNECTION NAME ID VERSION SYSNAME JOBNAME ASID STATE
-----
CSQEIBM1CSQ301 01 00010059 SC61 CSQ3MSTR 0091 ACTIVE
```

Observe o seguinte da saída do comando:

- Que `STORAGE_CLASS MEMORY` fornece a confirmação de que um **MAXIMUM** de 4096 MB de SCM foi incluído na estrutura.
- A figura `ALLOCATED` para a quantia de `STORAGE-CLASS MEMORY` usada para pré-preparação Agora há espaço livre na estrutura na qual não havia antes de o SCM ser incluído.
- A quantidade de `AUGMENTED SPACE` usada para controlar o uso de SCM.
- O ponto no qual o algoritmo de pré-migração começa a mover dados da estrutura para o SCM é quando a estrutura está 90% cheia. Isso é indicado pela propriedade **SCMHIGHTHRESHOLD** não configurável.
- O ponto abaixo do qual o algoritmo de pré-busca começa a mover dados do SCM para a estrutura é quando a estrutura está 70% cheia. Isso é indicado pela propriedade **SCMLOWTHRESHOLD** não configurável.

Agora é possível testar várias maneiras de otimizar o uso de SCM. Observe o seguinte :

- Depois que o SCM é usado para armazenar mensagens, não é possível alterar a estrutura até que você tenha removido todos os dados do SCM.

Nesse caso, isso significa que a proporção de entrada para elemento é congelada ao valor que estava em vigor quando SCM foi usado pela primeira vez. Deve-se cuidadosamente assegurar que a estrutura esteja no estado que desejar, antes que o algoritmo de pré-migração comece a mover dados para o SCM.

- O tamanho da estrutura atual está correto antes de usar o SCM?

Por exemplo, você aumentou **INITSIZE** de 512 MB para um `SIZE` de 1 GB?

Se você não fizer isso, é possível que, embora tenha ativado sua estrutura para auto-mudança, o algoritmo de pré-migração começará a mover dados para o SCM antes de a mudança ter a chance de ser iniciada. Como resultado, a estrutura é congelada usando 512 MB de armazenamento real.

- A proporção de entrada para elemento está correta antes de usar o SCM?

O objetivo desse cenário é aumentar o número de ponteiros de mensagem transferidos que podem ser armazenados na estrutura e no SCM como um todo, bem como manter o máximo possível de mensagens inteiramente no armazenamento de estrutura. Acessar essas mensagens é mais rápido do que acessar mensagens no SMDS.

Portanto, você precisa ter uma estrutura que comece com uma proporção de entrada para elemento ideal para armazenar mensagens, e depois fazer a transição para uma proporção ideal para armazenar ponteiros de mensagens antes que o algoritmo de pré-migração seja iniciado pela primeira vez. Essa transição pode ser alcançada, em parte, fazendo uso das regras de transferência do IBM MQ.

Mude as regras de transferência emitindo o comando a seguir:

```
ALTER CFSTRUCT(SCEN1) OFFLD1SZ(0K)
```

Pode ser necessário realizar várias execuções para otimizar as proporções de entrada para elemento.

A tabela a seguir mostra possíveis melhorias no número de mensagens colocadas na fila durante as diferentes fases do cenário de armazenamento emergencial.

Tabela 18. Comparação dos resultados para o cenário de armazenamento emergencial

Descrição do Teste	Número de mensagens	Tempo para preencher filas (segundos)
Configuração básica	27.850	3.2
SMDS com as regras de transferência padrão	205.000	158
SCM com as regras de transferência padrão	828.610	469
SCM com regras de transferência ajustadas	1.135.775	679

A última linha na tabela mostra que o ajuste das regras de transferência teve o efeito requerido.

Você precisa examinar seu sistema para ver se é possível melhorar esses números de alguma forma. Por exemplo, você poderá ficar sem armazenamento SMDS disponível. Se você pode alocar mais armazenamento SMDS, é necessário ser capaz de aumentar o número de mensagens na fila bastante significativamente.

Melhor desempenho – configuração básica

Como você configura um cenário básico para melhor desempenho usando filas compartilhadas no IBM MQ.

Sobre esta tarefa

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

Este cenário descreve o uso do SCM para aumentar o número de mensagens que podem ser armazenadas em uma fila compartilhada sem incorrer no custo de desempenho do uso do SMDS.

Esse cenário inicial é muito semelhante ao usado para armazenamento emergencial e usa um:

- Grupo de filas compartilhadas, IBM1, que contém um único gerenciador de filas, CSQ3. Além da estrutura de administração, o grupo de filas compartilhadas definiu uma única estrutura de aplicativo, SCEN2.
- Recurso de acoplamento (CF) CF01, no qual a estrutura de aplicativo SCEN2 é armazenada como a estrutura IBM1SCEN2. Essa estrutura tem um tamanho máximo de 2 GB.

- Fila compartilhada única, SCEN2.Q, que está configurada para usar a estrutura do aplicativo. Essa configuração está ilustrada no [Figura 63](#) na página 210.

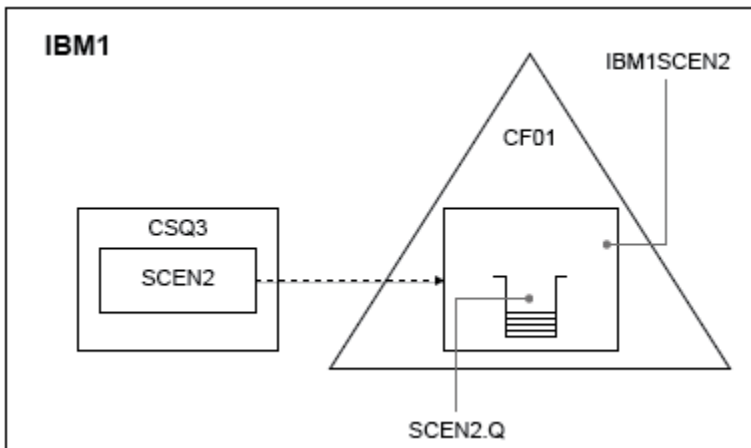


Figura 63. Configuração básica

Além disso, suponha que o gerenciador de filas CSQ3 já seja o único membro do grupo de filas compartilhadas IBM1.

Deve-se incluir a definição da estrutura IBM1SCEN2 na política do Coupling Facility Resource Manager (CFRM). Para simplificar, a estrutura é definida para que possa ser criada em apenas um único recurso de acoplamento, CF01, especificando PREFLIST (CF01).

Política CFRM de amostra para a estrutura IBM1SCEN2:

```
STRUCTURE
NAME (IBM1SCEN2)
SIZE (2048M)
INITSIZE (2048M)
ALLOWAUTOALT (YES)
FULLTHRESHOLD (85)
PREFLIST (CF01)
ALLOWREALLOCATE (YES)
DUPLEX (DISABLED)
ENFORCEORDER (NO)
```

As palavras-chave **INITSIZE** e **SIZE** têm o valor 2048M, de modo que a estrutura não pode ser redimensionada.

Procedimento

1. Atualize a política CFRM usando o comando a seguir:

```
SETXCF START ,POLICY ,TYPE=CFRM ,POLNAME=IBM1SCEN2
```

2. Verifique se a estrutura foi criada corretamente, usando o comando a seguir:

```
D XCF ,STR ,STRNAME=IBM1SCEN2
```

A emissão do comando anterior fornece a saída a seguir:

```
RESPONSE=SC61
IXC360I 07.58.51 DISPLAY XCF 581
STRNAME: IBM1SCEN2
STATUS: NOT ALLOCATED
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
```

```
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY
```

```
EVENT MANAGEMENT: MESSAGE-BASED   MANAGER SYSTEM NAME: SC53
MANAGEMENT LEVEL : 01050107
```

Nesse ponto, sua estrutura não foi alocada ao grupo de compartilhamento de filas, conforme mostrado pela linha STATUS.

3. Configure o IBM MQ para usar a estrutura definida na política de CFRM.
 - a. Use o comando `DEFINE CFSTRUCT`, com o nome da estrutura de SCEN2 para criar um objeto CFSTRUCT do IBM MQ.

```
DEFINE CFSTRUCT(SCEN2)
CFCONLOS(TOLERATE)
CFLEVEL(5)
DESCR('Structure for SCM scenario 2')
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFFLD1SZ(64K) OFFFLD1TH(70)
OFFFLD2SZ(64K) OFFFLD2TH(80)
OFFFLD3SZ(64K) OFFFLD3TH(90)
```

- b. Verifique a estrutura, usando o comando `DISPLAY CFSTRUCT`.
 - c. Defina a fila compartilhada SCEN2.Q para usar a estrutura SCEN2, utilizando o comando MQSC a seguir:

```
DEFINE QLOCAL(SCEN2.Q) QSGDISP(SHARED) CFSTRUCT(SCEN2) MAXDEPTH(999999999)
```

4. Use o IBM MQ Explorer para colocar uma única mensagem na fila SCEN2.Q e desativar a mensagem novamente.
5. Emita o comando a seguir para verificar se a estrutura agora é alocada:

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Revise a saída do comando, uma parte da é mostrada, e assegure que a linha STATUS mostre ALLOCATED.

```
RESPONSE=SC61
IXC360I 08.31.27 DISPLAY XCF 703
STRNAME: IBM1SCEN2
STATUS: ALLOCATED
EVENT MANAGEMENT: MESSAGE-BASED
TYPE: SERIALIZED LIST
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY
```

Além disso, anote os valores dos campos na seção SPACE USAGE:

- ENTRADAS

- ELEMENTS
- EMCS
- LOCKS

Segue um exemplo dos valores:

SPACE USAGE	IN-USE	TOTAL	%
ENTRIES:	344686	345242	99
ELEMENTS:	6548455	6548467	99
EMCS:	2	780318	0
LOCKS:	1024		

Resultados

Você criou a configuração básica. Agora será possível ter uma ideia do desempenho de linha de base de sua configuração usando o método selecionado.

Como proceder a seguir

É necessário testar o cenário básico. Como exemplo, é possível usar os três aplicativos a seguir, iniciando os aplicativos na ordem mostrada e executando-os simultaneamente.

1. Use um aplicativo PCF para solicitar o valor de profundidade atual (**CURDEPTH**) para SCEN2 . Q a cada cinco segundos. A saída pode ser usada para criar o gráfico da profundidade da fila com o passar do tempo.
2. Um aplicativo de obtenção de encadeamento único obtém repetidamente mensagens de SCEN2 . Q, usando uma obtenção com uma espera infinita. Para simular o processamento das mensagens que foram removidas, o aplicativo de obtenção é pausado por quatro milissegundos para cada dez mensagens removidas.
3. Um aplicativo de colocação de encadeamento único coloca um total de um milhão de mensagens não persistentes de 4 KB em SCEN2 . Q. Esse aplicativo não é pausado entre a colocação de cada mensagem, de forma que as mensagens são colocadas em SCEN2 . Q mais rápido do que o aplicativo de obtenção pode obtê-las.

Como resultado, quando o aplicativo de colocação está em execução, a profundidade de SCEN2 . Q aumenta.

Quando a estrutura IBM1SCEN2 é preenchida e o aplicativo de colocação recebe um código de razão MQRC_STORAGE_MEDIUM_FULL, o aplicativo de colocação é suspenso por cinco segundos antes de tentar colocar a próxima mensagem na fila.

É possível criar o gráfico da aplicação CURDEPTH durante um período de tempo. Você obtém uma forma de saída onda saw-tooth conforme o aplicativo de colocação é pausado para permitir que a fila fique parcialmente vazia.

Acesse [“Incluindo SCM na estrutura inicial”](#) na página 212.

Conceitos relacionados

[“Uso da memória de classe de armazenamento com filas compartilhadas”](#) na página 195

O uso da memória de classe de armazenamento (SCM) pode ser vantajoso quando usada com filas compartilhadas do IBM MQ for z/OS.

 [Incluindo SCM na estrutura inicial](#)

Como você inclui SCM para melhor desempenho no IBM MQ.

Sobre esta tarefa

Importante: O IBM z16 está planejado para ser a última geração do IBM Z[®] para suportar o uso da Memória Flash Virtual (também conhecida como Storage Class Memory ou SCM) para imagens do

Recurso de Acoplamento. Para obter mais informações, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Instruções de Direção](#).

Como alternativa, você deve usar estruturas maiores ou transferir mensagens para SMDS.

Esta parte da tarefa usa a configuração básica descrita em “Melhor desempenho – configuração básica” na página 209. O cenário de descreve a inclusão de SCM na estrutura inicial.

Essa configuração final é ilustrada na [Figura 64](#) na página 213.

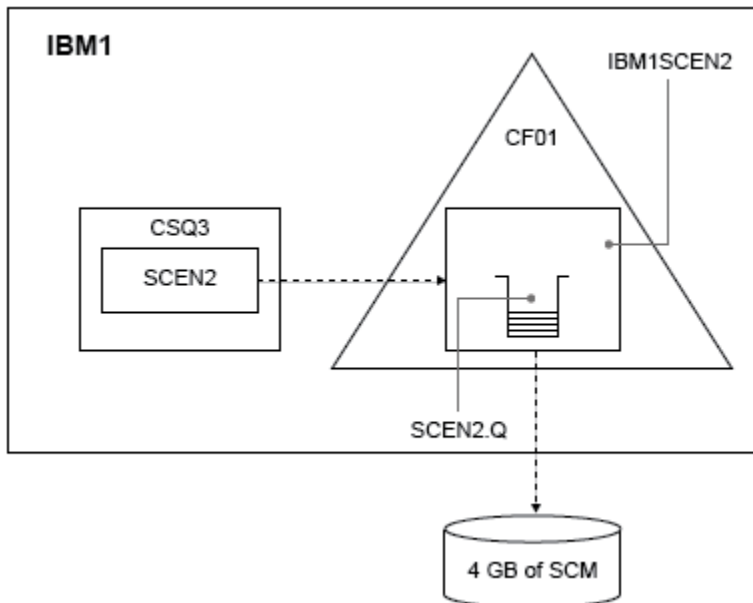


Figura 64. Configuração que inclui SCM para melhor desempenho

Procedimento

1. Inclua 4 GB de SCM na estrutura IBM1SCEN2, executando o procedimento a seguir:
 - a) Verifique quanto de SCM está instalado e alocado para CF01, emitindo o comando a seguir:

```
D CF,CFNAME=CF01
```

- b) Verifique as figuras STORAGE-CLASS MEMORY na seção STORAGE CONFIGURATION da saída exibida para ver o armazenamento disponível..
- c) Atualize a política de CFRM com as palavras-chave SCMMAXSIZE e SCMALGORITHM, conforme mostrado:

```
STRUCTURE  
NAME(IBM1SCEN2)  
SIZE(2048M)  
INITSIZE(2048M)  
ALLOWAUTOALT(YES)  
FULLTHRESHOLD(85)  
PREFLIST(CF01)  
ALLOWREALLOCATE(YES)  
DUPLEX(DISABLED)  
ENFORCEORDER(NO)  
SCMMAXSIZE(4G)  
SCMALGORITHM(KEYPRIORITY1)
```

2. Ative a política de CFRM emitindo o comando a seguir:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN2
```

3. Reconstrua a estrutura IBM1SCEN2.

Deve-se executar este procedimento porque a estrutura foi alocada quando você fez as mudanças anteriores.

Emita o comando a seguir para reconstruir a estrutura:

```
SETXCF START,REBUILD,STRNM=IBM1SCEN2
```

4. Emita o comando a seguir para confirmar a nova configuração da estrutura:

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Revise a saída do comando, uma parte da qual é descrita a seguir:

SPACE USAGE	IN-USE	TOTAL	%
ENTRIES:	33	342684	0
ELEMENTS:	48	6503697	0
EMCS:	2	575600	0
LOCKS:		1024	

Resultados

Calcule a mudança no uso de armazenamento real pelo aumento do armazenamento de controle para usar SCM.

- Antes da inclusão de SCM na estrutura, a estrutura tem estes totais, conforme mostrado em [“Melhor desempenho – configuração básica”](#) na página 209:
 - 345.242 entradas
 - 6.548.467 elementos
 - 780.318 EMCS
- Depois que SCM é incluído na estrutura, a estrutura tem estes totais:
 - 342.684 entradas
 - 6.503.697 elementos
 - 575.600 EMCS

Usando esses números, depois que o SCM foi incluído, a estrutura diminui de tamanho em:

- 2558 entradas
- 44.770 elementos
- 204.718 EMCS

A quantidade de armazenamento de estrutura que é usada para gerenciar o SCM é conforme a seguir para uma estrutura de 2 GB com 4 GB de SCM alocado:

$$(2558 + 44,770 + 204,718) * 256 = 61.5 \text{ MB}$$

Observe que a inclusão de mais SCM provavelmente obtém apenas uma redução marginal do tamanho da estrutura, porque a quantidade de armazenamento de controle usada para controlar o SCM aumenta, conforme o tamanho da estrutura e a quantidade de SCM alocado aumentam.

Como proceder a seguir

Repita os testes descritos na seção final de [“Melhor desempenho – configuração básica”](#) na página 209.

É possível criar o gráfico dos resultados da aplicação revisada durante um período de tempo. Comparando o gráfico com aquele obtido anteriormente, agora você obtém uma saída sem uma onda saw-tooth, uma vez que o aplicativo de colocação não mais precisa aguardar a fila ficar parcialmente vazia.

Para obter mais informações, consulte [MP16: WebSphere MQ for z/OS -Planejamento de capacidade e ajuste](#).

z/OS Enfileiramento distribuído e grupos de filas compartilhadas

O enfileiramento distribuído e os grupos de filas compartilhadas são duas técnicas que você pode usar para aumentar a disponibilidade dos sistemas do aplicativo. Use este tópico para localizar informações adicionais sobre essas técnicas.

Para complementar a alta disponibilidade de mensagens nas filas compartilhadas, o componente de enfileiramento distribuído de IBM MQ tem funções adicionais para fornecer o seguinte:

- Mais alta disponibilidade na rede.
- Capacidade aumentada para conexões de rede de entrada para o grupo de filas compartilhadas.

A [Figura 65 na página 215](#) ilustra o enfileiramento distribuído e grupos de filas compartilhadas. Ele mostra dois gerenciadores de filas em um sysplex, ambos pertencentes ao mesmo grupo de filas compartilhadas. Eles podem acessar a fila compartilhada SQ1. Gerenciadores de fila na rede (no AIX e Windows por exemplo) podem colocar as mensagens nesta fila por meio do inicializador de canais de um gerenciador de filas. Os aplicativos clonados em ambos os gerenciadores de filas atendem à fila.

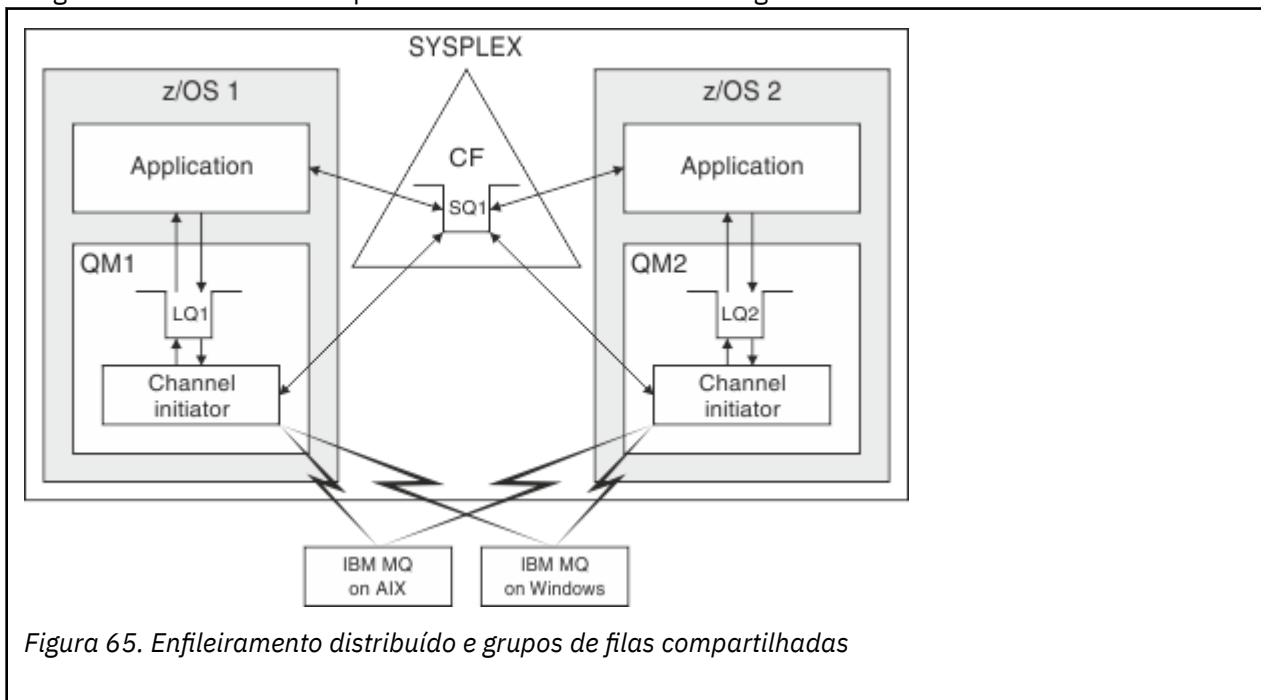


Figura 65. Enfileiramento distribuído e grupos de filas compartilhadas

Conceitos relacionados

[“Canais Compartilhados” na página 215](#)

Use este tópico para entender os conceitos de canais compartilhados e o seu uso com o IBM MQ for z/OS.

[“Fila do intra-grupo” na página 220](#)

Esta seção descreve o enfileiramento intragrupo, uma função exclusiva do IBM MQ for z/OS para a plataforma z/OS. Essa função está disponível apenas para gerenciadores de filas definidos para um grupo de filas compartilhadas.

[“Clusters e grupos de filas compartilhadas” na página 218](#)

Use este tópico para entender como é possível usar grupos de filas compartilhadas com clusters.

z/OS Canais Compartilhados

Use este tópico para entender os conceitos de canais compartilhados e o seu uso com o IBM MQ for z/OS.

Inúmeros produtos em rede fornecem um mecanismo para ocultar as falhas da rede ou para equilibrar as solicitações de rede de entrada em um conjunto de servidores elegíveis. Os produtos em rede tornam

uma *porta genérica* disponível para solicitações de conexão de entrada e a solicitação de entrada pode ser satisfeita conectando-se a um dos servidores elegíveis.

Esses produtos de rede incluem:

- recursos genéricos VTAM
- SYSPLEX Distributor

O inicializador de canais tira proveito desses produtos para usar as capacidades das filas compartilhadas

Existem dois tipos de canais compartilhados, *canal de entrada compartilhado* e *canal de saída compartilhado*.

- [Canais de entrada compartilhados](#)
- [Canais de saída compartilhados](#)

Para obter informações adicionais sobre os canais, consulte

- [Resumo do canal compartilhado](#)
- [Status do canal compartilhado](#)

Canais de Entrada Compartilhada

Cada inicializador de canais no grupo de filas compartilhadas inicia uma tarefa de listener adicional para atender em uma *porta genérica*. Esta porta genérica é disponibilizada para a rede por uma das tecnologias de suporte (VTAM, TCP/IP). As solicitações de conexão de rede de entrada para a porta genérica são despachadas pela tecnologia de rede, para qualquer um dos listeners no grupo de filas compartilhadas (QSG) que estão atendendo na porta genérica.

É possível iniciar um canal no inicializador de canais para o qual a conexão de entrada está direcionada se o inicializador de canais tiver acesso a uma definição de canal para um canal com esse nome. É possível definir uma definição de canal para ser privada para um gerenciador de filas ou armazenada no repositório compartilhado, com isso ficando disponível em qualquer lugar (uma definição global). Isso significa que é possível tornar uma definição de canal disponível em qualquer inicializador de canais no grupo de filas compartilhadas definindo-a como uma definição global.

Há uma diferença adicional ao iniciar um canal por meio da porta genérica: a sincronização do canal é com o grupo de filas compartilhadas e não com um gerenciador de filas individual. Por exemplo, considere um gerenciador de filas remotas iniciando um canal por meio da porta genérica. Quando o canal é iniciado pela primeira vez, ele pode ser iniciado no gerenciador de filas QM1 e no fluxo de mensagens. Se o canal parar e for reiniciado no gerenciador de filas QM2, as informações sobre o número de mensagens transmitidas ainda estarão corretas porque a sincronização é com o grupo de filas compartilhadas.

É possível usar um canal de entrada iniciado por meio da porta genérica para colocar as mensagens em qualquer fila. O gerenciador de filas remotas não sabe se a fila de destino é compartilhada ou não. Se a fila de destino for uma fila compartilhada, o gerenciador de filas remotas se conectará por meio de qualquer inicializador de canais disponível de uma maneira de carregamento balanceado e as mensagens são colocadas na fila compartilhada.

Se a fila de destino for uma fila privada, as mensagens serão colocadas na fila privada de propriedade de qualquer gerenciador de filas à qual a instância atual do canal esteja conectada. Neste ambiente, conhecido como *filas locais replicadas*, cada gerenciador de filas deve ter o mesmo conjunto de filas privadas definido.

Configurando os canais SVRCONN para um grupo de filas compartilhadas

A configuração ideal para os canais SVRCONN em um grupo de filas compartilhadas é configurar os listeners privados em cada CHINIT que usa um número de porta diferente a partir dos canais ponto a ponto. Essas portas do listener são usadas como recursos 'back-end' para um novo mecanismo de distribuição de carga de trabalho como o Sysplex Distributor que usa os endereços IP Virtual (VIPAs). O

endereço VIPA externo é então usado como o endereço de destino para as definições CLNTCONN na rede. O canal SVRCONN pode ser definido com QSGDISP(GROUP) de maneira que a mesma definição fique disponível a todos os gerenciadores de fila no QSG. Essa configuração evita o uso de um listener compartilhado e, portanto, reduz o efeito de desempenho do grupo de filas compartilhadas que mantém o estado do canal compartilhado, que não é necessário para os canais cliente / servidor.

Canais de Saída Compartilhados

Um canal de saída será considerado um canal compartilhado se estiver recebendo as mensagens de uma fila de transmissão compartilhada. Se ele for compartilhado, reterá as informações de sincronização no grupo de filas compartilhadas nível. Isso significa que o canal pode ser reiniciado em um gerenciador de filas e instância do inicializador de canais diferentes dentro do grupo de filas compartilhadas se o subsistema de comunicações, o inicializador de canais ou o gerenciador de filas falhar. Reiniciar os canais com falha dessa maneira é um recurso de canais compartilhados denominado *recuperação de canal peer*.

Balanceamento de carga de trabalho para canais de saída compartilhados

Um canal compartilhado de saída será elegível para ser iniciado em qualquer inicializador de canais dentro do grupo de filas compartilhadas se você não especificou que deseja que ele seja iniciado em um determinado inicializador de canais. O inicializador de canais selecionado pelo IBM MQ é determinado usando os seguintes critérios:

- É o subsistema de comunicações necessário atualmente disponível para o inicializador de canais?
- É uma conexão Db2 disponível para o inicializador de canais?
- Qual inicializador de canais tem a menor carga de trabalho atual? A carga de trabalho inclui canais que estejam ativos e sendo tentados novamente.

Resumo de Canal Compartilhado

Os canais compartilhados são diferentes dos canais privados das seguintes maneiras:

Canal Privado

Ligado a um único inicializador de canais.

- O canal de saída usa uma fila de transmissão local.
- O canal de entrada iniciou por meio de uma porta local.
- Informações de sincronização retidas na fila SYSTEM.CHANNEL.SYNCQ.

Canal Compartilhada

Carga de trabalho balanceada com alta disponibilidade.

- O canal de saída usa uma fila de transmissão compartilhada.
- O canal de entrada iniciou por meio de uma porta genérica.
- Informações de sincronização retidas na fila SYSTEM.QSG.CHANNEL.SYNCQ.

Especifique se um canal é privado ou compartilhado quando você inicia o canal usando as opções CHLDISP com o comando `START CHANNEL`. Um canal compartilhado pode ser iniciado acionando da mesma maneira que um canal privado. No entanto, quando um canal compartilhado é iniciado, o IBM MQ executa o balanceamento de carga de trabalho e inicia o canal no inicializador de canais mais apropriado dentro do grupo de filas compartilhadas. (Se necessário, é possível especificar que um canal compartilhado seja iniciado em um inicializador de canais específico.)

Status de Canal Compartilhado

Os inicializadores de canais em um grupo de filas compartilhadas mantêm uma tabela de status de canal compartilhado no Db2. Isso registra quais canais estão ativos em quais inicializadores de canais. A tabela de status de canal compartilhado será usada se ocorrer uma falha no sistema de comunicações

ou inicializador de canais. Ele indica quais canais precisam ser reiniciados em um inicializador de canais diferente no grupo de filas compartilhadas.

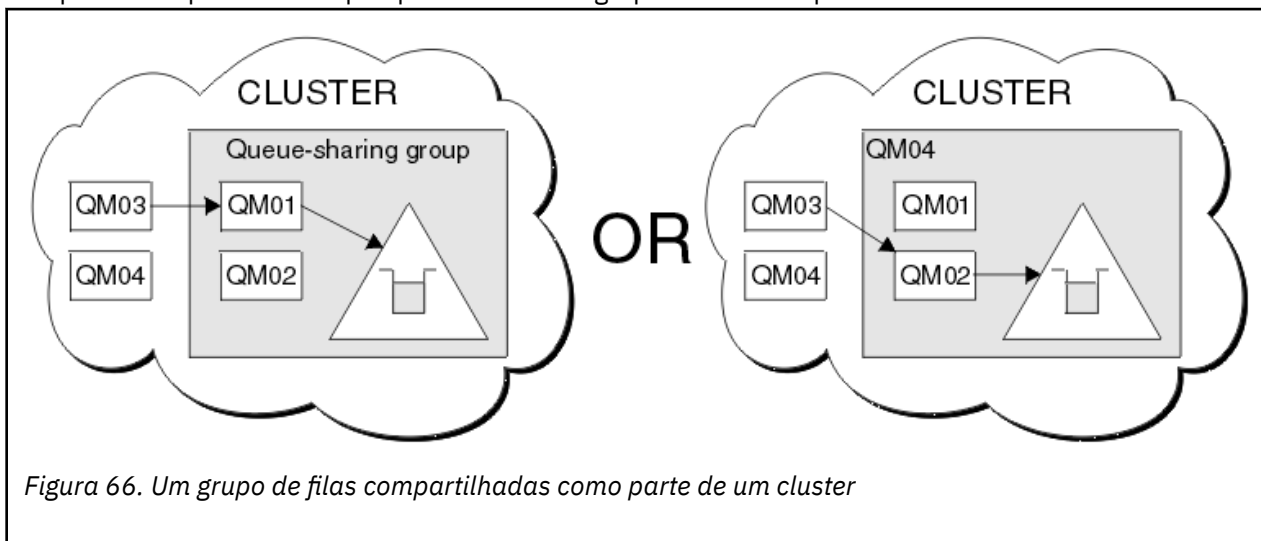
z/OS Clusters e grupos de filas compartilhadas

Use este tópico para entender como é possível usar grupos de filas compartilhadas com clusters.

É possível disponibilizar suas filas compartilhadas disponíveis para um cluster em uma única definição. Para isso, especifique o nome do cluster quando definir a fila compartilhada.

Os usuários na rede veem a fila compartilhada como sendo hospedada por cada gerenciador de filas dentro do grupo de filas compartilhadas (a fila compartilhada não é anunciada como sendo hospedada pelo grupo de filas compartilhadas). Os clientes podem iniciar as sessões com qualquer membro do grupo de filas compartilhadas para colocar mensagens na mesma fila compartilhada.

A [Figura 66 na página 218](#) mostra como os membros de um cluster podem acessar uma fila compartilhada por meio de qualquer membro do grupo de filas compartilhadas.



z/OS Influenciando a Distribuição de Carga de Trabalho com Filas Compartilhadas

Use este tópico para entender os fatores que afetam a distribuição de carga de trabalho com filas compartilhadas em um grupo de filas compartilhadas.

IBM MQ não fornece balanceamento de carga de trabalho para filas compartilhadas. No entanto, a distribuição de carga de trabalho em um grupo de filas compartilhadas (QSG) pode ser influenciada em *um modo baseado em pull*. A escolha de qual gerenciador de filas atenderá uma fila (receber uma mensagem gravada em uma fila compartilhada) é afetada pela capacidade de processamento disponível de cada gerenciador de filas no grupo de filas compartilhadas e as metas de gerenciamento de carga de trabalho definidas através do sysplex.

No entanto, é importante compreender que o gerenciador de filas que executa o MQPUT de uma mensagem também pode ter uma grande influência na decisão de qual gerenciador de filas recebe a mensagem.

É mais provável que um gerenciador de filas locais execute o MQGET

Para um aplicativo que executa um MQPUT, o gerenciador de filas locais é considerado o gerenciador de fila ao qual o aplicativo está conectado.

Exatamente qual gerenciador de fila atende um MQPUT de uma mensagem executando um MQGET em nome de um aplicativo de obtenção é influenciado pelas seguintes considerações.

Quando uma mensagem é colocada em uma fila compartilhada vazia, o gerenciador de filas locais geralmente é postado antes de qualquer um dos outros gerenciadores de filas no grupo de filas

compartilhadas seja notificado. Se o gerenciador de filas locais estiver em uma posição para processar a mensagem, ele receberá uma notificação de transição da lista do recurso de acoplamento (CF) antes de qualquer outro gerenciador de filas locais no QSG. (Uma lista de notificação de transição de lista é uma notificação de que a fila compartilhada alterou o estado de vazio para não vazio.

Os possíveis cenários, neste caso, são os seguintes:

1. MQPUT de mensagem não persistente fora do ponto de sincronização e *fast put to waiting getter*.

Se houver um aplicativo com um *MQGET com espera* no gerenciador de filas locais para a fila, o MQPUT da mensagem é passado diretamente para o buffer do aplicativo de obtenção e não gravado para a fila. Isso é válido para filas compartilhadas e não compartilhadas. Este recurso geralmente é denominado mecanismo *fast put to a waiting getter*. No caso de filas compartilhadas, nenhum outro gerenciador de filas no QSG é notificado porque não há transição de vazio para não vazio da fila. Isso significa, por exemplo, que, desde que esse gerenciador de filas possa atender a todas as colocações desse aplicativo e presumindo que nenhum outro aplicativo esteja colocando mensagens na fila, nenhum outro gerenciador de filas no grupo de filas compartilhadas ajudará na liberação dessa fila. Se, no entanto, não houver *MQGET com espera* no gerenciador de filas locais e uma mensagem for colocada na fila compartilhada, o CF notificará outros gerenciadores de filas no grupo de filas compartilhadas de acordo com suas regras para notificações de transições de lista.

2. MQPUT de uma mensagem persistente ou mensagem no ponto de sincronização.

Nesse caso, se houver um aplicativo com um *MQGET com espera* no gerenciador de filas locais, então, a mensagem será colocada na fila compartilhada e o CF notificará outros gerenciadores de filas no grupo de filas compartilhadas de acordo com suas regras para notificações de lista transições. No entanto, o gerenciador de filas locais não espera uma notificação de transição de CF, mas primeiro honra qualquer *MQGET com espera* local e geralmente executa a obtenção dessa mensagem em nome do aplicativo antes que qualquer outro gerenciador de filas no grupo de filas compartilhadas possa responder a uma notificação de CF. Isso depende da forma como o gerenciador de filas locais está ocupado. Caso contrário, qualquer gerenciador de filas notificado pelo CF devido à chegada da mensagem na fila vazia tentará atender a obtenção primeiro. O primeiro gerenciador de filas a responder processa a nova mensagem.

3. Finalmente, se a fila não for o dreno de mensagens, em que CF tenha enviado uma notificação de uma mudança de estado de vazio para não vazio para a fila, todos os gerenciadores de filas conectados terão uma oportunidade de ajudar no processamento da fila. Neste caso, a carga de trabalho será considerada *baseada em pull*.

Esse design permite o melhor desempenho sobre uma distribuição de carga de trabalho baseada puramente em pull. O objetivo é obter vantagem da alta disponibilidade oferecida pelas filas retidas no CF enquanto permite que o gerenciador de filas execute, onde possível, o *MQGET* sem precisar fazer referência ao CF e assim processar a carga de trabalho de mensagem a forma mais eficiente possível.

As abordagens alternativas podem ser adotadas onde a ênfase no balanceamento da carga de trabalho for mais importante do que os aprimoramentos de desempenho descritos anteriormente. Por exemplo, assegurando que nenhum dos aplicativos de obtenção seja conectado ao mesmo gerenciador de filas ao qual o aplicativo de inserção está conectado. Ao usar este design, todas as mensagens são colocadas na fila e todos os gerenciadores de fila no QSG são notificados quando a fila move de vazia para não vazia, de acordo com o algoritmo CF para tratar dessas transições. Além disso, o mecanismo *fast put to waiting getter* não é aplicável.

Onde localizar mais informações sobre filas compartilhadas e grupos de filas compartilhadas?

Use a tabela neste tópico para localizar mais informações sobre como o IBM MQ for z/OS usa as filas compartilhadas e os grupos de filas compartilhadas.

Tabela 19. Onde localizar mais informações sobre filas compartilhadas e grupos de filas compartilhadas?

Tópico	Onde Localizar
Recuperação do grupo de filas compartilhadas	“Recuperação e reinicialização no z/OS” na página 261
Segurança do grupo de filas compartilhadas	“Conceitos de segurança no IBM MQ for z/OS” na página 280
Definições de objeto global e privado Direcionando Comandos para diferentes filas gerenciadores	Origens das quais é possível emitir comandos no z/OS
Planejando seu Recurso de Acoplamento ambiente	Definindo Recursos de Acoplamento
Planejando seu ambiente do SMDS	Planejando o Ambiente do Shared Message Data Set (SMDS)
Planejando seu Db2 ambiente	Planejando seu ambiente do Db2
Configurando filas compartilhadas Parâmetros do sistema	“Filas compartilhadas e grupos de filas compartilhadas” na página 172
Programas utilitário Migrando filas	IBM MQ utilitários em z/OS referência
Mensagens do console	Mensagens para o IBM MQ for z/OS
Comandos MQSC	Comandos do MQSC
Clusters do IBM MQ	Configurando um cluster do gerenciador de filas
enfileiramento distribuído do IBM MQ Nomes de canal	Introdução ao Gerenciamento de Filas Distribuído
Gravando aplicativos	Visão geral do design do aplicativo
chamada MQCONN	MQCONN

z/OS

Fila do intra-grupo

Esta seção descreve o enfileiramento intragrupo, uma função exclusiva do IBM MQ for z/OS para a plataforma z/OS. Essa função está disponível apenas para gerenciadores de filas definidos para um grupo de filas compartilhadas.

Para obter informações sobre grupos de filas compartilhadas, consulte [“Filas compartilhadas e grupos de filas compartilhadas” na página 172](#).

Conceitos de enfileiramento intragrupo

É possível executar a transferência rápida de mensagens entre os gerenciadores de filas em um grupo de filas compartilhadas sem definir os canais. Isso usa uma fila do sistema denominada SYSTEM.QSG.TRANSMIT.QUEUE, que é uma fila de transmissão compartilhada. Cada gerenciador de filas no grupo de filas compartilhadas inicia uma tarefa chamada de agente de enfileiramento intragrupo, que aguarda a chegada nessa fila das mensagens que são destinadas para seu gerenciador de filas. Quando essa mensagem é detectada, ela é removida da fila e colocada na fila de destino correta.

As regras de resolução de nome padrão são usadas, mas, se o enfileiramento intragrupo (IGQ) estiver ativado e o gerenciador de filas de destino estiver dentro do grupo de filas compartilhadas, o SYSTEM.QSG.TRANSMIT.QUEUE será usado para transferir a mensagem para o gerenciador de filas de destino correto em vez de usar uma fila de transmissão e canal.

Ative o enfileiramento intragrupo por meio de um atributo do gerenciador de filas. O enfileiramento intragrupo move as mensagens não persistentes fora do ponto de sincronização e as mensagens persistentes dentro do ponto de sincronização. Se localizar um problema na entrega de mensagens para a fila de destino, o enfileiramento intragrupo tentará colocá-las na fila de devoluções. Se a fila de devoluções estiver cheia ou indefinida, as mensagens não persistentes serão descartadas, mas as mensagens persistentes serão restauradas e retornadas ao SYSTEM.QSG.TRANSMIT.QUEUE e o agente IGQ tentará entregar as mensagens até que sejam bem-sucedidas.

Um canal compartilhado de entrada que recebe uma mensagem destinada para uma fila em um gerenciador de filas diferente no grupo de filas compartilhadas pode usar o enfileiramento intragrupo para *saltar* a mensagem para o destino correto.

Pode haver vezes em que você queira que o gerenciador de filas locais coloque uma mensagem diretamente na fila de destino, se a fila de destino for uma fila compartilhada e não da mensagem sendo transferida primeiro para o gerenciador de filas de destino. É possível usar o atributo do gerenciador de filas SQQMNAME para controlar isso. Se você configurar o valor de SQQMNAME para USE, o comando MQOPEN será executado no gerenciador de filas especificado pelo **ObjectQMgrName**.

No entanto, se a fila de destino for uma fila compartilhada e você configurar o valor de SQQMNAME para IGNORE e o **ObjectQMgrName** for aquele de outro gerenciador de fila no grupo de filas compartilhadas, a fila compartilhada será aberta no gerenciador de fila local. Se o gerenciador de filas locais não puder abrir a fila de destino ou colocar uma mensagem na fila, a mensagem será transferido para o **ObjectQMgrName** especificado por meio de IGQ ou de um canal IBM MQ .

O enfileiramento intragrupo pode ser usado para entregar mensagens pequenas de forma mais eficiente para as filas que residem em gerenciadores de filas remotas dentro de um grupo de filas compartilhadas. O enfileiramento intragrupo também suporta mensagens grandes, sendo a maior de 100 MB *menos* o comprimento do cabeçalho da fila de transmissão

Nota: Se você usar esse recurso, os usuários deverão ter o mesmo acesso às filas em cada gerenciador de filas no grupo de filas compartilhadas.

O diagrama a seguir mostra um exemplo típico do enfileiramento intragrupo.

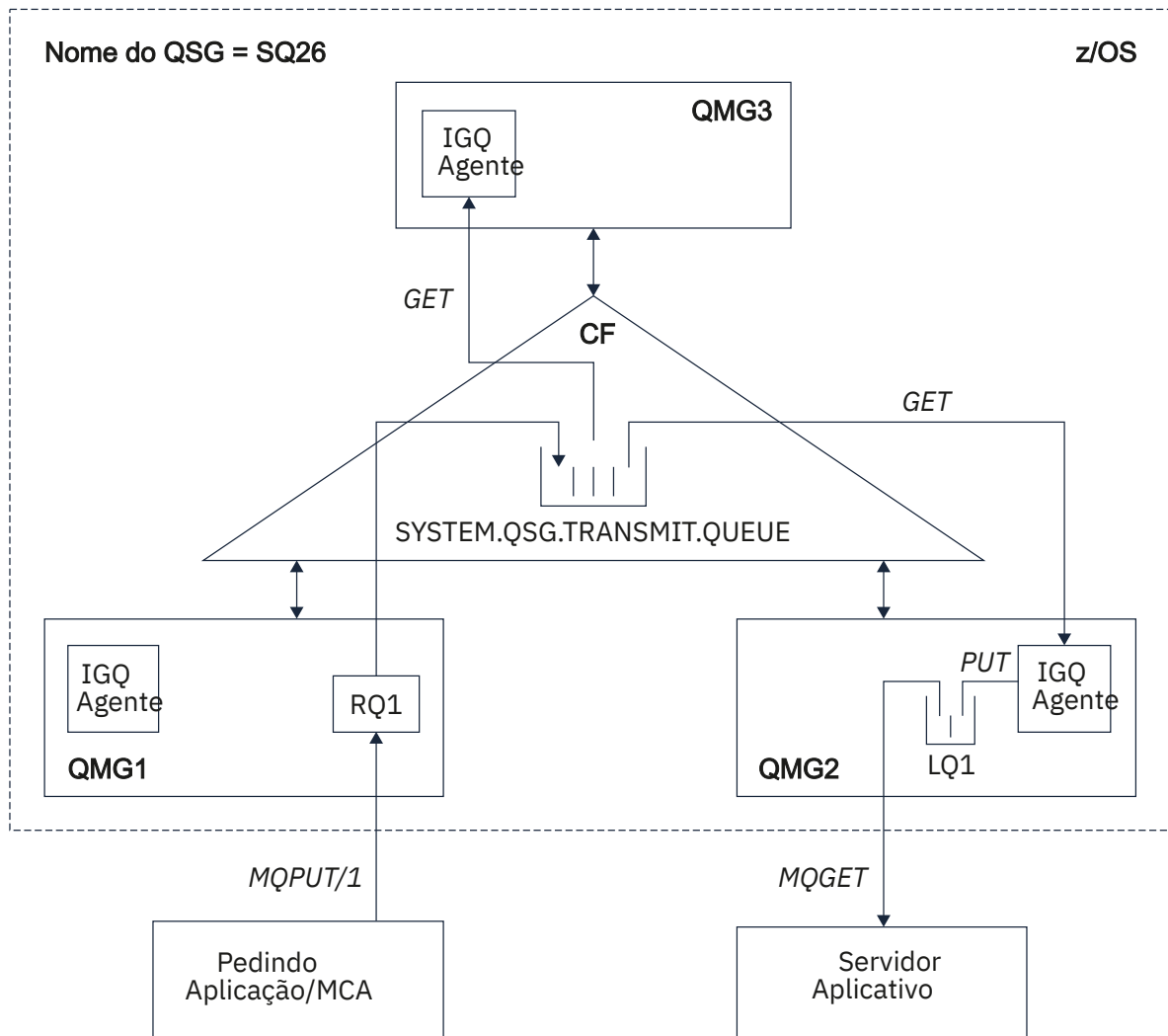


Figura 67. Um Exemplo de Enfileiramento Intragrupo

O diagrama mostra:

- Agentes IGQ em execução em três gerenciadores de filas (QMG1, QMG2 e QMG3) que são definidos para um grupo de filas compartilhadas chamado SQ26.
- A fila de transmissão compartilhada SYSTEM.QSG.TRANSMIT.QUEUE definida no recurso de acoplamento (CF).
- Uma definição de fila remota que é definida no gerenciador de filas QMG1.
- Uma fila local que é definida no gerenciador de filas QMG2.
- Um aplicativo de solicitação (este aplicativo poderia ser um Agente do Canal de Mensagem (MCA)) que é conectado do gerenciador de filas QMG1.
- Um aplicativo do servidor que é conectado ao gerenciador de filas QMG2.
- Uma mensagem de solicitação sendo colocada no SYSTEM.QSG.TRANSMIT.QUEUE.

Enfileiramento Intragrupo e Agente de Enfileiramento Intragrupo

Um agente IGQ é iniciado durante a inicialização do gerenciador de filas. Quando os aplicativos abrem e colocam as mensagens nas filas remotas, o gerenciador de filas locais determina se o enfileiramento intragrupo é usado para transferência de mensagem. Se o enfileiramento intragrupo tiver que ser usado,

o gerenciador de filas locais colocará a mensagem no SYSTEM.QSG.TRANSMIT.QUEUE. O agente IGQ no gerenciador de filas remotas de destino recupera a mensagem e a coloca na fila de destino.

Terminologia de enfileiramento intragrupo

Explicações da terminologia: enfileiramento intragrupo, fila de transmissão compartilhada para uso pelo enfileiramento intragrupo e agente de enfileiramento intragrupo.

Fila intragrupo

O enfileiramento intragrupo pode afetar potencialmente a transferência de mensagens rápida e menos dispendiosa entre os gerenciadores de filas em um grupo de filas compartilhadas, sem a necessidade de definir canais.

Fila de Transmissão Compartilhada para Uso pelo Enfileiramento Intragrupo

Cada grupo de filas compartilhadas tem uma fila de transmissão compartilhada denominada SYSTEM.QSG.TRANSMIT.QUEUE para uso pelo enfileiramento intragrupo. Se o enfileiramento intragrupo estiver ativado, SYSTEM.QSG.TRANSMIT.QUEUE aparecerá no caminho de resolução do nome ao abrir as filas remotas. Quando os aplicativos (incluindo Agentes do Canal de Mensagem (MCAs)) colocarem as mensagens em uma fila remota, o gerenciador de filas locais determinará a elegibilidade de mensagens para transferência rápida e as colocará no SYSTEM.QSG.TRANSMIT.QUEUE.

Agente de Enfileiramento Intragrupo

O agente IGQ é a tarefa, iniciada na inicialização do gerenciador de filas, que aguarda que as mensagens adequadas cheguem no SYSTEM.QSG.TRANSMIT.QUEUE. O agente IGQ recupera as mensagens adequadas desta fila e as entrega para as filas de destino.

O agente IGQ para cada gerenciador de filas sempre é iniciado porque o enfileiramento intragrupo é usado pelo próprio gerenciador de filas para seu próprio processamento interno.

Benefícios de enfileiramento intragrupo

Os benefícios do enfileiramento intragrupo são: definições de sistema reduzidas, administração de sistema reduzidas, desempenho melhorado, suporta migração e entrega de mensagens quando executar multi-hop entre os gerenciadores de filas em um grupo de filas compartilhadas.

Os benefícios do enfileiramento intragrupo são:

Definições do sistema reduzidas

O enfileiramento intragrupo remove a necessidade de definir canais entre os gerenciadores de filas em um grupo de filas compartilhadas.

Administração do sistema reduzida

Como não há canais definidos entre os gerenciadores de filas em um grupo de filas compartilhadas, não há nenhum requisito para administração de canal.

Desempenho melhorado

Como há apenas um agente IGQ necessário para a entrega de uma mensagem para uma fila de destino (em vez de dois agentes intermediários emissor e receptor), a entrega de mensagens que usam o enfileiramento intragrupo pode ser menos oneroso do que a entrega de mensagens que usam os canais. No enfileiramento intragrupo existe apenas um componente de recebimento, porque a necessidade para o componente de envio foi removida. Este salvamento ocorre porque a mensagem está disponível para o agente IGQ no gerenciador de fila de destino para entrega na fila de destino assim que a operação de inserção no gerenciador de filas locais tiver sido concluída e, no caso das mensagens colocadas no escopo do ponto de sincronização, confirmada.

Suporte de migração

Os aplicativos externos a um grupo de filas compartilhadas podem entregar mensagens para uma fila que reside em qualquer gerenciador de filas no grupo de filas compartilhadas, enquanto estiver

sendo conectado apenas a um gerenciador de filas específico no grupo de filas compartilhadas. Isso ocorre porque as mensagens que chegam em um canal receptor, destinado a uma fila em um gerenciador de filas remotas, podem ser enviadas de forma transparente para a fila de destino que usa o enfileiramento intragrupo. Esse recurso permite que os aplicativos sejam implementados entre o grupo de filas compartilhadas sem a necessidade de mudar qualquer sistema que seja externo ao grupo de filas compartilhadas.

Uma configuração típica é ilustrada pelo seguinte diagrama, em que:

- Um aplicativo de solicitação conectado ao gerenciador de filas QMG1 precisa enviar uma mensagem a uma fila local no gerenciador de filas QMG3.
- O gerenciador de filas QMG1 é conectado apenas ao gerenciador de filas QMG2.
- Os gerenciadores de filas QMG2 e QMG3, que foram anteriormente conectados usando canais, agora são membros do grupo de filas compartilhadas SQ26.

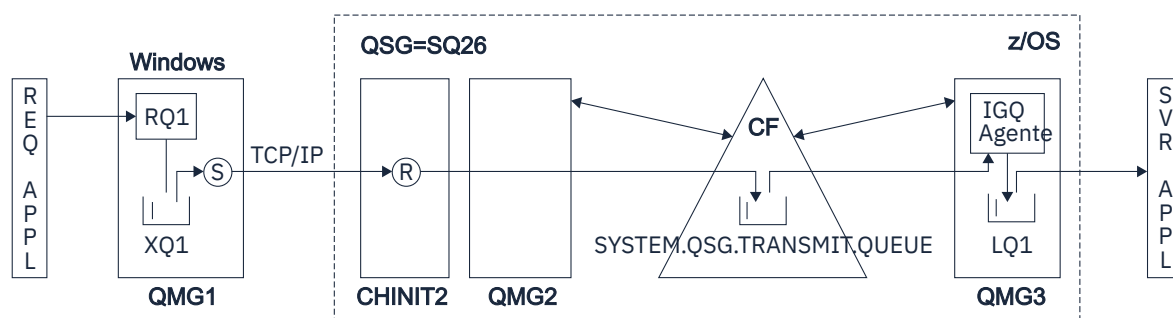


Figura 68. Um exemplo de suporte de migração

O fluxo de operações é o seguinte:

1. O aplicativo de solicitação coloca uma mensagem, destinada para a fila local LQ1 no gerenciador de filas remotas QMG3, na definição de fila remota RQ1.
2. O gerenciador de filas QMG1, em execução em uma estação de trabalho do Windows NT, coloca as mensagens na fila de transmissão XQ1.
3. O MCA (S) emissor no QM1 transmite a mensagem, usando TCP/IP, para o MCA (R) receptor no inicializador de canais CHINIT2.
4. O MCA (R) receptor no inicializador de canais CHINIT2 coloca as mensagens na fila de transmissão compartilhada SYSTEM.QSG.TRANSMIT.QUEUE.
5. O agente IGQ no gerenciador de filas QMG3 recupera a mensagem do SYSTEM.QSG.TRANSMIT.QUEUE e a coloca na fila local de destino LQ1.
6. O aplicativo do servidor recupera a mensagem da fila local de destino e a processa.

Entrega de mensagens ao executar múltiplos saltos entre os gerenciadores de filas em um grupo de filas compartilhadas

O diagrama anterior em [Suporta migração](#) também ilustra a entrega de mensagens executando múltiplos saltos entre os gerenciadores de filas em um grupo de filas compartilhadas. As mensagens que chegam em um gerenciador de filas dentro do grupo de filas compartilhadas, mas destinadas a uma fila em outro gerenciador de filas no grupo de filas compartilhadas, podem ser facilmente transmitidas para a fila de destino no gerenciador de filas de destino, usando o enfileiramento intragrupo.

Limitações de enfileiramento intragrupo

As limitações do enfileiramento intragrupo são: mensagens elegíveis para transferência usando o enfileiramento intragrupo, número de agente de enfileiramento intragrupo por gerenciador de filas e iniciando e parando o agente de enfileiramento intragrupo.

Este tópico descreve as limitações do enfileiramento intragrupo.

As mensagens elegíveis para transferência usando o enfileiramento intragrupo

Como o enfileiramento intragrupo usa uma fila de transmissão compartilhada que é definida no recurso de acoplamento (CF), o enfileiramento intragrupo é limitado a entregar mensagens com o comprimento máximo de mensagens suportado para as filas compartilhadas, menos o comprimento de um cabeçalho de fila de transmissão (MQXQH).

Número de agentes do enfileiramento intragrupo por gerenciador de filas

Apenas um agente IGQ é iniciado por gerenciador de filas em um grupo de filas compartilhadas.

Iniciando e parando o agente de enfileiramento intragrupo

O agente IGQ é iniciado durante a inicialização do gerenciador de filas e terminado durante o encerramento do gerenciador de filas. Ele é projetado para ser uma tarefa de longa execução, de autorrecuperação (no caso de finalização anormal). Se ocorrer um erro com a definição do SYSTEM.QSG.TRANSMIT.QUEUE (por exemplo, se esta fila for Get inibido), o agente IGQ continuará tentando novamente. Se o agente IGQ encontrar um erro que resulte no término normal do agente enquanto o gerenciador de filas ainda estiver ativo, ele poderá ser reiniciado emitindo um comando ALTER QMGR IGQ(ENABLED). Este comando evita a necessidade de reciclar o gerenciador de filas.

Configurando o atributo do gerenciador de filas IGQ como ENABLED ou DISABLED

Se o atributo do gerenciador de filas IGQ for configurado como ENABLED ou DISABLED, manipulações de objetos existentes poderão ser invalidadas com código de razão MQRC_OBJECT_CHANGED. Veja [Introdução ao enfileiramento intragrupo](#) para obter mais informações.

Introdução ao enfileiramento intragrupo

É possível ativar, desativar e usar o enfileiramento intragrupo conforme descrito neste tópico.

Ativando o enfileiramento intragrupo

Para ativar o enfileiramento intragrupo em seus gerenciadores de filas, você precisa fazer o seguinte:

- Definir uma fila de transmissão compartilhada chamada SYSTEM.QSG.TRANSMIT.QUEUE. A definição dessa fila pode ser localizada em thlqual.SCSQPROC (CSQ4INSS), a amostra CSQINP2 para objetos SYSTEM para os grupos de filas compartilhadas. Esta fila deve ser definida com os atributos corretos, conforme indicado em thlqual.SCSQPROC (CSQ4INSS), para que o enfileiramento entre grupos funcione corretamente.
- Como o agente IGQ sempre é iniciado na inicialização do gerenciador de filas, o enfileiramento intragrupo sempre fica disponível para o processamento da mensagem de entrada. O agente IGQ processa quaisquer mensagens que forem colocadas no SYSTEM.QSG.TRANSMIT.QUEUE. No entanto, para ativar o enfileiramento intragrupo para processamento de saída, o atributo do gerenciador de filas IGQ deve ser configurado como ENABLED.

Importante: Se o atributo do gerenciador de filas IGQ for configurado como ENABLED, manipulações de objetos existentes poderão ser invalidadas com código de razão MQRC_OBJECT_CHANGED. Consulte [“Propriedades específicas de enfileiramento intragrupo”](#) na [página 233](#) para obter mais informações. Conforme descrito na seção 'Resposta do programador' para este código de razão, os aplicativos precisam ser codificados para lidar com esta situação (consulte 2041 (07F9) (RC2041): MQRC_OBJECT_CHANGED para obter mais detalhes).

Além disso, como o IGQ é projetado como uma tarefa de longa execução e recuperação automática, que inicia durante a inicialização e finaliza com o encerramento, veja [“Limitações de enfileiramento intragrupo”](#) na [página 224](#) para obter informações adicionais.

Desativando o enfileiramento intragrupo

Para desativar o enfileiramento intragrupo para transferência da mensagem de saída, configure o atributo do gerenciador de filas IGQ para DISABLED. Se o enfileiramento intragrupo estiver desativado para um determinado gerenciador de filas, o agente IGQ nesse gerenciador de filas ainda poderá processar as mensagens de entrada que foram colocadas no SYSTEM.QSG.TRANSMIT.QUEUE por um gerenciador de filas que tenha o enfileiramento intragrupo ativado para transferência de saída.

Importante: Se o atributo do gerenciador de filas IGQ for configurado como ENABLED, manipulações de objetos existentes poderão ser invalidadas com código de razão MQR_OBJECT_CHANGED. Consulte “[Propriedades específicas de enfileiramento intragrupo](#)” na página 233 para obter mais informações. Conforme descrito na seção 'Resposta do programador' para este código de razão, os aplicativos precisam ser codificados para lidar com esta situação (consulte [2041 \(07F9\) \(RC2041\)](#)): [MQR_OBJECT_CHANGED](#) para obter mais detalhes).

Além disso, como o IGQ é projetado como uma tarefa de longa execução e recuperação automática, que inicia durante a inicialização e finaliza com o encerramento, veja “[Limitações de enfileiramento intragrupo](#)” na página 224 para obter informações adicionais.

Usando o enfileiramento intragrupo

Assim que o enfileiramento intragrupo é ativado, ele fica disponível para uso e um gerenciador de filas usa-o sempre que possível. Ou seja, quando um aplicativo colocar uma mensagem em uma definição de fila remota, para uma fila remota completa ou para uma fila de cluster, o gerenciador de filas determinará se a mensagem é elegível para ser entregue usando o enfileiramento intragrupo e, se for, coloca a mensagem no SYSTEM.QSG.TRANSMIT.QUEUE. Não há necessidade de alterar os aplicativos do usuário ou filas do aplicativo, porque, para as mensagens elegíveis, o gerenciador de filas usa o SYSTEM.QSG.TRANSMIT.QUEUE, na preferência para qualquer outra fila de transmissão.

Configurações de Enfileiramento Intragrupo

Além da configuração típica de enfileiramento intragrupo outras configurações são possíveis.

O “[Conceitos de enfileiramento intragrupo](#)” na página 221 descreve a configuração típica.

Conceitos relacionados

“[Enfileiramento Distribuído com Enfileiramento Intragrupo \(diversos caminhos de entrega\)](#)” na página 226

Para aplicativos que processam mensagens curtas, pode ser viável configurar o enfileiramento intragrupo apenas para entregar mensagens entre os gerenciadores de filas em um grupo de filas compartilhadas.

“[Armazenamento em Cluster com Enfileiramento Intragrupo \(diversos caminhos de entrega\)](#)” na página 228

É possível configurar os gerenciadores de filas para que eles estejam em um cluster, bem como em um grupo de filas compartilhadas.

“[Armazenamento em Cluster, Enfileiramento Intragrupo e Enfileiramento Distribuído](#)” na página 231

É possível configurar um gerenciador de filas que é um membro de um cluster, bem como um grupo de filas compartilhadas e está conectado a um gerenciador de filas distribuído usando um par de canais emissor/receptor.

Enfileiramento Distribuído com Enfileiramento Intragrupo (diversos caminhos de entrega)

Para aplicativos que processam mensagens curtas, pode ser viável configurar o enfileiramento intragrupo apenas para entregar mensagens entre os gerenciadores de filas em um grupo de filas compartilhadas.

A opção de enfileiramento dentro do grupo sobre canais de comunicação pode ser controlada pelo nível do tipo CFSTRUCT. (3 em vez de 4 ou 5). O comprimento máximo da mensagem, conforme configurado no SYSTEM.QSQ.TRANSMIT.QUEUE.

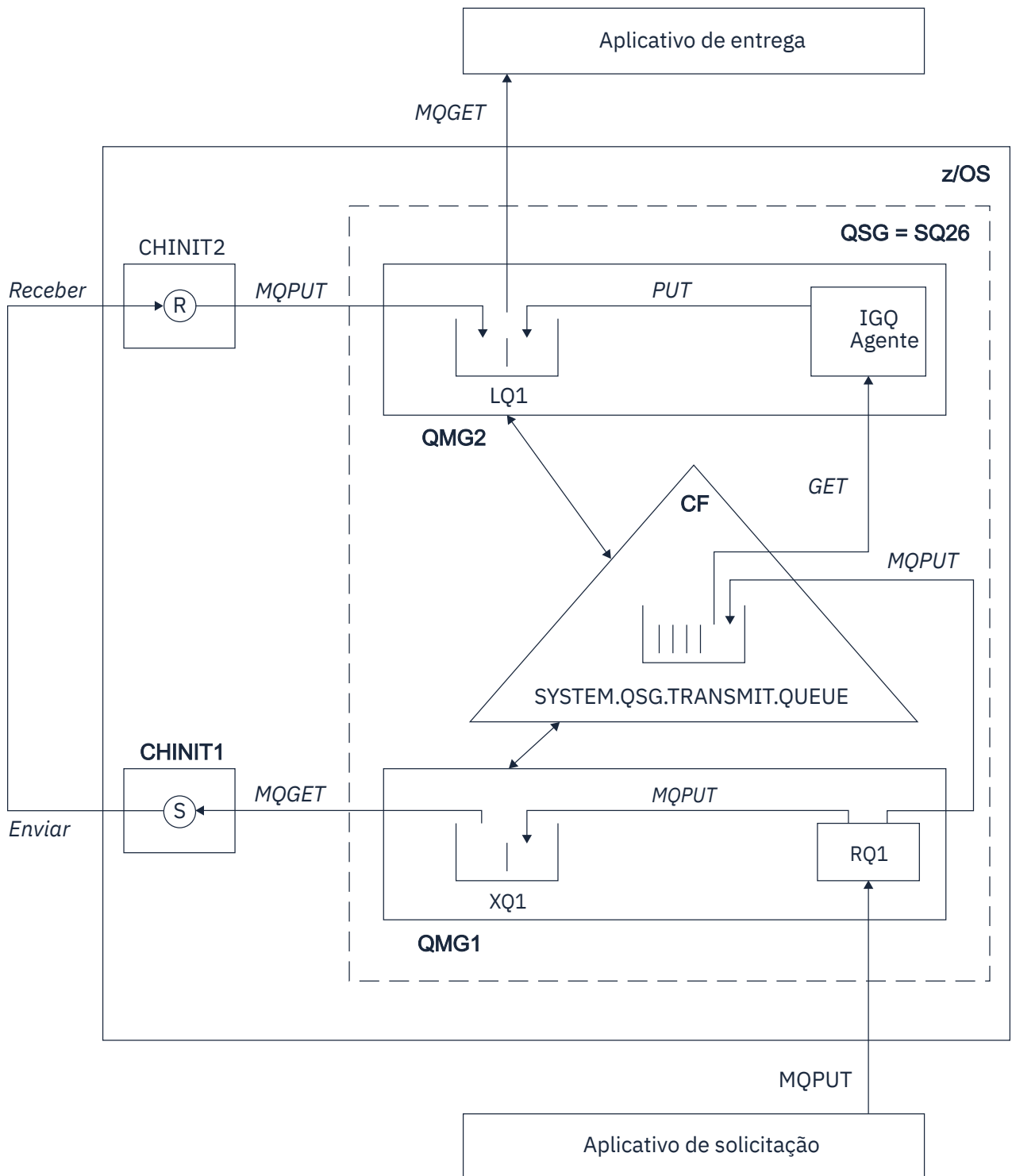


Figura 69. Uma Configuração de Exemplo

Processamento de Abertura/Inserção

1. É importante observar que quando o aplicativo de solicitação abre a fila remota RQ1, a resolução de nome ocorre para a fila de transmissão não compartilhada XQ1 e a fila de transmissão compartilhada SYSTEM.QSG.TRANSMIT.QUEUE.
2. Quando o aplicativo de solicitação coloca uma mensagem na fila remota, considerando se o enfileiramento intragrupo está ativado para a transferência de saída no gerenciador de filas e nas características da mensagem, a mensagem é colocada na fila de transmissão XQ1 ou

na fila de transmissão SYSTEM.QSG.TRANSMIT.QUEUE. O gerenciador de filas coloca todas as mensagens na fila de transmissão XQ1, e todas as mensagens pequenas na fila de transmissão SYSTEM.QSG.TRANSMIT.QUEUE.

3. Se a fila de transmissão XQ1 estiver cheia ou não estiver disponível, as solicitações de inserção para grandes mensagens falharão sincronicamente com um retorno e código de razão adequados. No entanto, as solicitações de inserção para mensagens pequenas continuarão bem-sucedidas e serão colocadas na fila de transmissão SYSTEM.QSG.TRANSMIT.QUEUE.
4. Se a fila de transmissão SYSTEM.QSG.TRANSMIT.QUEUE estiver cheia ou não puder receber inserção, as solicitações de inserção para mensagens pequenas falharão sincronicamente com um retorno e código de razão adequados. No entanto, as solicitações de inserção para mensagens grandes continuarão bem-sucedidas e serão colocadas na fila de transmissão XQ1. Neste caso, nenhuma tentativa será feita para colocar as mensagens pequenas em uma fila de transmissão.

Fluxo para Mensagens Grandes

1. O aplicativo de solicitação coloca as mensagens grandes na fila remota RQ1.
2. O gerenciador de filas QMG1 coloca as mensagens na fila de transmissão XQ1.
3. O MCA (S) emissor no gerenciador de filas QMG1 recupera as mensagens da fila de transmissão XQ1 e as envia para o gerenciador de filas QMG2.
4. O MCA (R) receptor no gerenciador de fila QMG2 recebe as mensagens e as coloca na fila de destino LQ1.
5. O aplicativo de entrega recupera e, em seguida, processa as mensagens da fila LQ1.

Fluxo para Mensagens Pequenas

1. O aplicativo de solicitação coloca as mensagens pequenas na fila remota RQ1.
2. O gerenciador de filas QMG1 coloca as mensagens na fila de transmissão SYSTEM.QSG.TRANSMIT.QUEUE.
3. IGQ no gerenciador de filas QMG2 recupera as mensagens e as coloca na fila de destino LQ1.
4. O aplicativo de entrega recupera as mensagens da fila LQ1.

Pontos a Observar

1. O aplicativo de solicitação não precisa estar ciente do mecanismo subjacente usado para a entrega de mensagens.
2. Um mecanismo de entrega de mensagens potencialmente mais rápido pode ser alcançado para mensagens pequenas.
3. Diversos caminhos estão disponíveis para entrega de mensagem (ou seja, a rota de canal normal e a rota de enfileiramento intragrupo).
4. A rota de enfileiramento intragrupo, sendo potencialmente mais rápida, é selecionada na preferência para a rota de canal normal. Dependendo das características da mensagem, a entrega da mensagem pode ser dividida em dois caminhos. Portanto, as mensagens podem ser entregues fora da sequência (embora esta entrega também seja possível se as mensagens forem entregues usando apenas a rota de canal normal).
5. Quando uma rota tiver sido selecionada e as mensagens tiverem sido colocadas nas filas de transmissão, apenas a rota selecionada será usada para a entrega da mensagem. Qualquer mensagem não processada no SYSTEM.QSG.TRANSMIT.QUEUE não será desviada para a fila de transmissão XQ1.

Armazenamento em Cluster com Enfileiramento Intragrupo (diversos caminhos de entrega)

É possível configurar os gerenciadores de filas para que eles estejam em um cluster, bem como em um grupo de filas compartilhadas.

Quando as mensagens são enviadas para uma fila de clusters e os gerenciadores de filas de destino local e remoto estão no mesmo grupo de filas compartilhadas, o enfileiramento intragrupo é usado para a entrega de mensagens pequenas (usando o SYSTEM.QSG.TRANSMIT.QUEUE) e a entrega de mensagens grandes, se o enfileiramento intragrupo suporta o tamanho da mensagem. Além disso, o SYSTEM.CLUSTER.TRANSMIT.QUEUE é usado para a entrega de mensagens para qualquer gerenciador de filas que esteja no cluster, mas fora do grupo de filas compartilhadas. O diagrama a seguir ilustra esta configuração (os inicializadores de canais não são mostrados).

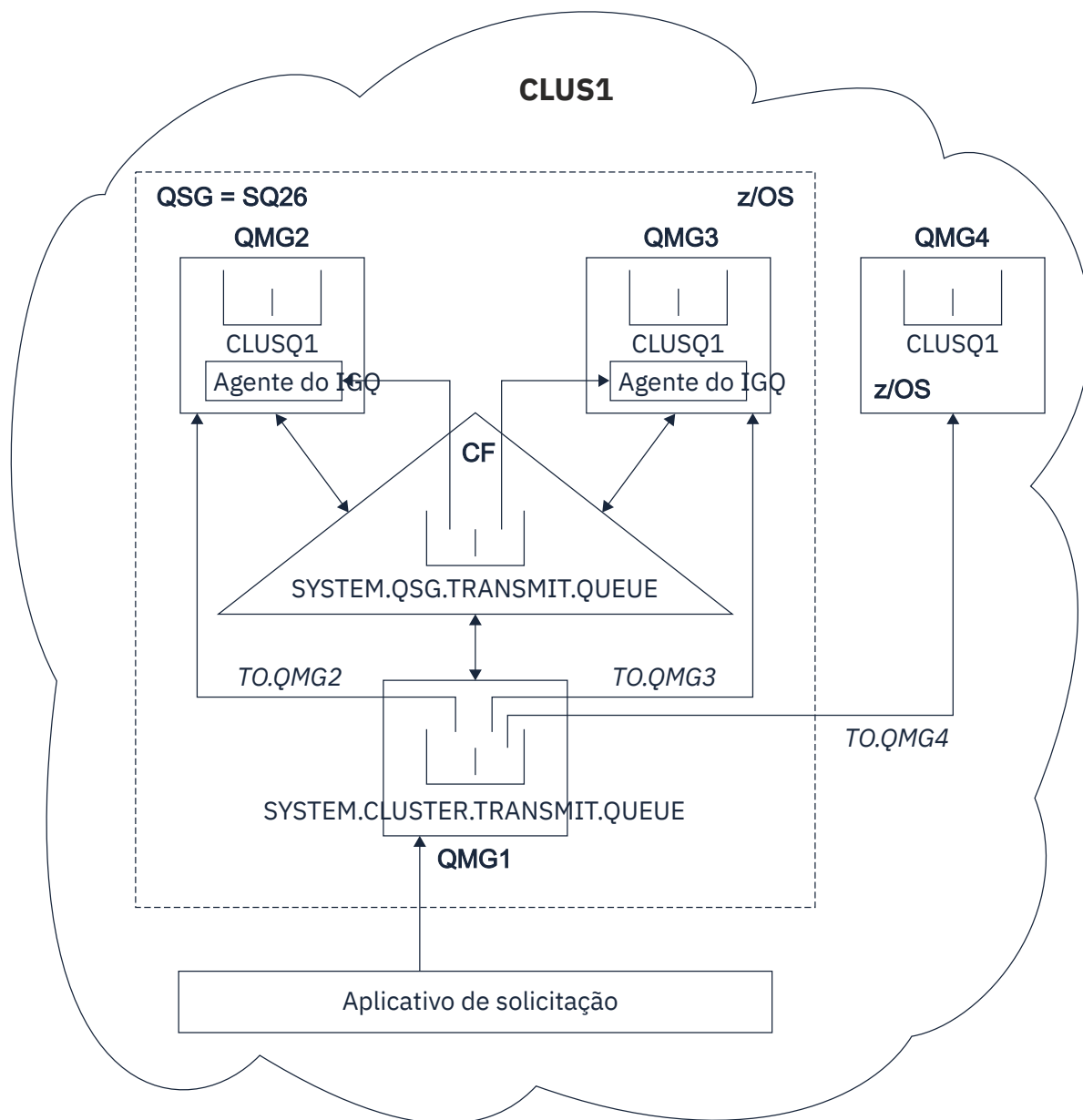


Figura 70. Um Exemplo de Armazenamento em Cluster com Enfileiramento Intragrupo

O diagrama mostra:

- Quatro gerenciadores de filas do z/OS QMG1, QMG2, QMG3 e QMG4 configurados em um cluster CLUS1.
- Gerenciadores de filas QMG1, QMG2 e QMG3 configurados em um grupo de filas compartilhadas SQ26.
- Agentes IGQ em execução nos gerenciadores de filas QMG2 e QMG3.
- O SYSTEM.CLUSTER.TRANSMIT.QUEUE local definido em QMG1.

Nota: Para esclarecimento, o SYSTEM.CLUSTER.TRANSMIT.QUEUE não é mostrado em outros gerenciadores de filas.

- O sistema SYSTEM.QSG.TRANSMIT.QUEUE definido no CF, que está em uma estrutura IBM MQ configurada com o atributo CFLEVEL (3) RECOVER (YES).
- Os canais de cluster TO.QMG2 (conectando QMG1 ao QMG2), TO.QMG3 (conectando QMG1 ao QMG3), e TO.QMG4 (conectando QMG1 ao QMG4).
- A fila de clusters CLUSQ1 sendo hospedada nos gerenciadores de filas QMG2, QMG3 e QMG4.

Assuma que o aplicativo de solicitação abra a fila de clusters com a opção MQOO_BIND_NOT_FIXED; de modo que o gerenciador de filas de destino para a fila de clusters seja selecionada no momento da inserção.

Se o gerenciador de filas de destino selecionado for QMG2:

- Todas as mensagens grandes colocadas pelo aplicativo solicitante são:
 - Colocadas em SYSTEM.CLUSTER.TRANSMIT.QUEUE em QMG1, pois SYSTEM.QSG.TRANSMIT.QUEUE está em uma estrutura CFLEVEL(3); portanto, apenas mensagens de até 63 KB são suportadas.
 - Transferidas para a fila de clusters CLUSQ1 em QMG2 usando o canal de cluster TO.QMG2
- Todas as mensagens pequenas colocadas pelo aplicativo de solicitação são
 - Colocadas na fila de transmissão compartilhada SYSTEM.QSG.TRANSMIT.QUEUE. Essa fila está em uma estrutura configurada com o atributo RECOVER(YES), portanto, ela é usada para mensagens persistentes, não persistentes e pequenas.
 - Recuperada pelo agente IGQ no QMG2
 - Colocadas na fila de clusters CLUSQ1 em QMG2

Se o gerenciador de filas de destino selecionado for QMG4:

- Como o QMG4 não é um membro do grupo de filas compartilhadas SQ26, todas as mensagens colocadas pelo aplicativo de solicitação são
 - Colocadas no SYSTEM.CLUSTER.TRANSMIT.QUEUE no QMG1
 - Transferidas para a fila de clusters CLUSQ1 em QMG4 usando o canal de cluster TO.QMG4

Pontos a Observar

- O aplicativo de solicitação não precisa estar ciente do mecanismo subjacente usado para a entrega de mensagens.
- Um mecanismo de entrega potencialmente mais rápido é alcançado para a transferência de pequenas mensagens não persistentes entre os gerenciadores de filas em um grupo de filas compartilhadas (até se os mesmos gerenciadores de filas estiverem em um cluster).
- Diversos caminhos estão disponíveis para entrega de mensagem (ou seja, a rota de cluster e a rota de enfileiramento intragrupo).
- A rota de enfileiramento intragrupo, sendo potencialmente mais rápida, é selecionada na preferência para a rota de cluster. Dependendo das características da mensagem, a entrega da mensagem pode ser dividida em dois caminhos. Portanto, as mensagens podem ser entregues fora da sequência fora da sequência. É importante observar que esta entrega é possível sem considerar a opção MQOO_BIND_* especificada pelo aplicativo. O enfileiramento intragrupo distribui as mensagens da mesma maneira que o armazenamento em cluster, considerando se MQOO_BIND_NOT_FIXED, MQOO_BIND_ON_OPEN, MQOO_BIND_ON_GROUP ou MQOO_BIND_AS_Q_DEF é especificado na abertura.
- Quando uma rota tiver sido selecionada e as mensagens tiverem sido colocadas nas filas de transmissão, apenas a rota selecionada será usada para a entrega da mensagem. Qualquer mensagem não processada no SYSTEM.QSG.TRANSMIT.QUEUE não será desviada para SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Armazenamento em Cluster, Enfileiramento Intragrupo e Enfileiramento Distribuído

É possível configurar um gerenciador de filas que é um membro de um cluster, bem como um grupo de filas compartilhadas e está conectado a um gerenciador de filas distribuído usando um par de canais emissor/receptor.

Esta configuração é uma combinação de enfileiramento distribuído com o enfileiramento intragrupo e o armazenamento em cluster com enfileiramento intragrupo.

O enfileiramento intragrupo está descrito em “[Enfileiramento Distribuído com Enfileiramento Intragrupo \(diversos caminhos de entrega\)](#)” na página 226.

O armazenamento em cluster com o enfileiramento intragrupo está descrito em “[Armazenamento em Cluster com Enfileiramento Intragrupo \(diversos caminhos de entrega\)](#)” na página 228.

Mensagens de Enfileiramento Intragrupo

Esta seção descreve as mensagens colocadas no SYSTEM.QSG.TRANSMIT.QUEUE.

Estrutura da Mensagem

Como todas as outras mensagens que são colocadas nas filas de transmissão, as mensagens que são colocadas no SYSTEM.QSG.TRANSMIT.QUEUE recebem o prefixo do cabeçalho da fila de transmissão (MQXQH).

Persistência de mensagem

No IBM WebSphere MQ 5.3 e acima, as filas compartilhadas suportam mensagens persistentes e não persistentes.

Se o gerenciador de filas terminar enquanto o agente IGQ estiver processando mensagens não persistentes ou se o agente IGQ terminar de maneira anormal enquanto estiver no meio do processamento das mensagens, as mensagens não persistentes sendo processadas podem ser perdidas. Os aplicativos devem se organizar para a recuperação das mensagens não persistentes, se sua recuperação for necessária.

Se uma solicitação de inserção para uma mensagem não persistente, emitida pelo agente IGQ, falhar inesperadamente, a mensagem sendo processada será perdida.

Entrega de Mensagens

O agente IGQ recupera e entrega todas as mensagens não persistentes fora do escopo do ponto de sincronização e todas as mensagens persistentes no escopo do ponto de sincronização. Neste caso, o agente IGQ age como o coordenador do ponto de sincronização. Portanto, o agente IGQ processa mensagens não persistentes assim como mensagens não persistentes de via rápida são processadas em um canal de mensagens. Consulte [Mensagens Rápidas, não Persistentes](#).

Envio em Lote de Mensagens

O agente IGQ usa um tamanho do lote fixo de 50 mensagens. Qualquer mensagem persistente recuperada em um lote é confirmada em intervalos de 50 mensagens. O agente confirma um lote que consiste nas mensagens persistentes quando não houver mais mensagens disponíveis para recuperação no SYSTEM.QSG.TRANSMIT.QUEUE.

Tamanho de Mensagem

O tamanho máximo da mensagem que pode ser colocado no SYSTEM.QSG.TRANSMIT.QUEUE é o comprimento máximo de mensagem suportado para as filas compartilhadas menos o comprimento de um cabeçalho de fila de transmissão (MQXQH).

Persistência de mensagem padrão e prioridade da mensagem padrão

Se o SYSTEM.QSG.TRANSMIT.QUEUE estiver no caminho de resolução de nome da fila estabelecido no momento da abertura, então para as mensagens que são colocadas com a persistência padrão e a prioridade padrão (ou com a persistência padrão ou a prioridade padrão), as regras normais serão aplicadas na seleção da fila que tem valores padrões de prioridade e persistência que são usados. (Consulte a seção [Mensagens do IBM MQ](#), para obter mais informações sobre as regras de seleção de fila).

Conceitos relacionados

“Mensagens não Entregues/Não Processadas” na página 232

Este tópico descreve o que acontece nas mensagens não entregues e não processadas no SYSTEM.QSG.TRANSMIT.QUEUE.

“Mensagens de relatório - enfileiramento entre grupos” na página 232

Este tópico descreve as mensagens de relatório: Confirmação de chegada, confirmação de entrega, relatório de expiração e relatório de exceção.

Mensagens não Entregues/Não Processadas

Este tópico descreve o que acontece nas mensagens não entregues e não processadas no SYSTEM.QSG.TRANSMIT.QUEUE.

Se um agente IGQ não puder entregar uma mensagem à fila de destino, o agente IGQ:

- Honrará a opção de relatório MQRO_DISCARD_MSG (se o campo de opções do Relatório do MQMD para a mensagem não entregue indica que deve) e descartará a mensagem não entregue.
- Tentará colocar a mensagem não entregue na fila de devoluções para o gerenciador de filas de destino, se a mensagem ainda não tiver sido descartada. O agente IGQ coloca o prefixo na mensagem com um cabeçalho de fila de devoluções (MQDLH).

Se uma fila de devoluções não estiver definida ou se uma mensagem não entregue não puder ser colocada na fila de devoluções e se a mensagem não entregue for:

- persistente, o agente IGQ vai restaurar o lote atual de mensagens persistentes que está processando e vai inserir um estado de nova tentativa. Para obter informações adicionais, consulte [“Propriedades específicas de enfileiramento intragrupo”](#) na página 233.
- não persistente, o agente IGQ vai descartar a mensagem e continuará processando a próxima mensagem.

Se um gerenciador de filas em um grupo de filas compartilhadas for finalizado antes de seu agente IGQ associado ter tido tempo de processar todas as suas mensagens, as mensagens não processadas permanecerão no SYSTEM.QSG.TRANSMIT.QUEUE até que o gerenciador de filas seja iniciado na próxima vez. O agente IGQ então recupera e entrega as mensagens às filas de destino.

Se o recurso de acoplamento falhar antes que todas as mensagens no SYSTEM.QSG.TRANSMIT.QUEUE tenham sido processadas, quaisquer mensagens não persistentes não processadas serão perdidas.

IBM recomenda que os aplicativos não coloquem as mensagens diretamente nas filas de transmissão. Se um aplicativo colocar as mensagens diretamente no SYSTEM.QSG.TRANSMIT.QUEUE, o agente IGQ talvez não consiga processar essas mensagens e elas permanecerão no SYSTEM.QSG.TRANSMIT.QUEUE. Em seguida, os usuários precisam usar seus próprios métodos para lidar com essas mensagens não processadas.

Mensagens de relatório - enfileiramento entre grupos

Este tópico descreve as mensagens de relatório: Confirmação de chegada, confirmação de entrega, relatório de expiração e relatório de exceção.

Mensagens de relatório de confirmação de chegada (COA)/confirmação de entrega (COD)

As mensagens COA e COD são geradas pelo gerenciador de filas, quando o enfileiramento intragrupo for usado.

Mensagens de relatório de expiração

Mensagens de relatório de expiração são geradas pelo gerenciador de filas.

Mensagens do Relatório de Exceção

Dependendo da opção do relatório MQRO_EXCEPTION_* especificada no campo *Opções de Relatório* do descritor de mensagens para a mensagem não entregue, o agente IGQ gera o relatório de exceção necessária e o coloca na fila de resposta especificada. O enfileiramento intragrupo pode ser usado para entregar o relatório de exceções para a fila de resposta de destino.

A persistência da mensagem de relatório é a mesma que a persistência da mensagem não entregue. Se o agente IGQ falhar ao resolver o nome da fila de resposta de destino ou se falhar ao colocar a mensagem de resposta em uma fila de transmissão (para transferência subsequente para a fila de resposta de destino), ele tentará colocar o relatório de exceção na fila de devoluções do gerenciador de filas no qual a mensagem de relatório é gerada. Se não for possível, então caso a mensagem não entregue seja:

- persistente, o agente IGQ descartará o relatório de exceções, restaurará o lote atual de mensagens e inserirá um estado de nova tentativa. Para obter informações adicionais, consulte [“Propriedades específicas de enfileiramento intragrupo”](#) na página 233.
- não persistente, o agente IGQ descartará o relatório de exceções e continuará o processamento da próxima mensagem no SYSTEM.QSG.TRANSMIT.QUEUE.

Segurança para enfileiramento intragrupo

Este tópico descreve os acordos de segurança para enfileiramento intragrupo.

Os atributos do gerenciador de filas IGQAUT (autoridade IGQ) e IGQUSER (ID do usuário agente IGQ) podem ser configurados para controlar o nível de verificação de segurança que é executado quando o agente IGQ abre as filas de destino.

Autoridade de Enfileiramento Intragrupo (IGQAUT)

O atributo IGQAUT pode ser configurado para indicar o tipo de verificações de segurança a ser executado e, portanto, para determinar os IDs de usuário a serem usados pelo agente IGQ quando estabelecer a autoridade para colocar as mensagens na fila de destino.

O atributo IGQAUT é análogo ao atributo PUTAUT disponível nas definições de canal.

Identificador de usuário de enfileiramento entre grupos (IGQUSER)

O atributo IGQUSER pode ser usado para nomear um ID do usuário a ser usado pelo agente IGQ quando estabelecer a autoridade para colocar as mensagens em uma fila de destino.

O atributo IGQUSER é análogo ao atributo MCAUSER disponível nas definições de canal.

Propriedades específicas de enfileiramento intragrupo

Esta seção descreve as propriedades específicas de enfileiramento intragrupo, incluindo invalidação de manipulações de objetos, recurso de recuperação automática e nova tentativa do agente de enfileiramento intragrupo e o agente de enfileiramento intragrupo e serialização.

Invalidação de Manipulações de Objetos (MQRC_OBJECT_CHANGED)

Se for descoberto que os atributos de um objeto foram mudados depois que o objeto foi aberto, o gerenciador de filas invalidará a manipulação de objetos com MQRC_OBJECT_CHANGED em seu próximo uso.

O enfileiramento intragrupo apresenta as regras a seguir para invalidação de manipulação de objetos:

- Se o SYSTEM.QSG.TRANSMIT.QUEUE foi incluído no caminho de resolução do nome durante o processamento aberto porque o enfileiramento intragrupo estava ENABLED no momento da abertura, mas o enfileiramento intragrupo era considerado DISABLED no momento da inserção, então o gerenciador de filas invalidará a manipulação de objetos e falhará a solicitação de inserção com MQRC_OBJECT_CHANGED.
- Se o SYSTEM.QSG.TRANSMIT.QUEUE não foi incluído no caminho de resolução do nome durante o processamento aberto porque o enfileiramento intragrupo estava DISABLED no momento da abertura, mas o enfileiramento intragrupo era considerado ENABLED no momento da inserção, então o gerenciador de filas invalidará a manipulação de objetos e falhará a solicitação de inserção com MQRC_OBJECT_CHANGED.
- Se SYSTEM.QSG.TRANSMIT.QUEUE foi incluído no caminho de resolução do nome durante o processamento aberto porque o enfileiramento intragrupo estava ativado no momento da abertura, mas

for descoberto que a definição SYSTEM.QSG.TRANSMIT.QUEUE foi alterada pelo momento da inserção, então o gerenciador de filas invalidará a manipulação de objetos e falhará na solicitação de inserção com MQRC_OBJECT_CHANGED.

Auto Recuperação do Agente de Enfileiramento Intragrupo

Se o agente IGQ terminar de maneira anormal, a mensagem CSQM067E será emitida e o agente IGQ será iniciado novamente.

Capacidade de Nova Tentativa do Agente de Enfileiramento Intragrupo

Se o agente IGQ encontrar um problema ao acessar o SYSTEM.QSG.TRANSMIT.QUEUE (porque não está definido, por exemplo ou estiver definido com os atributos incorretos ou estiver inibido para Gets ou por alguma outra razão), o agente IGQ não entrará no estado de nova tentativa.

O agente IGQ observa as contagens de repetições longas e curtas. Os valores para essas contagens e intervalos, que não podem ser mudados, são os seguintes:

<i>Tabela 20. Valores de intervalos e contagens de repetições longas e curtas</i>	
Constant	Value
Contagem de tentar novamente curta	10
Intervalo de tentar novamente curta	60 segundos = 1 min
Contagem de tentar novamente longa	999,999,999
Intervalo de tentar novamente longo	1200 seconds = 20 min

O agente de enfileiramento intragrupo e serialização

Uma tentativa, pelo agente IGQ para serializar o acesso às filas compartilhadas embora a recuperação peer-to-peer ainda esteja em andamento possa falhar.

Se houver uma falha de um gerenciador de filas em um grupo de filas compartilhadas enquanto o agente IGQ estiver lidando com mensagens não confirmadas em uma fila ou em filas compartilhadas, o agente IGQ será finalizado e a recuperação peer-to-peer compartilhada ocorrerá para o gerenciador de filas com falha. Como a recuperação peer-to-peer compartilhada é uma atividade assíncrona, ela deixa a possibilidade para o gerenciador de filas com falha e também o agente IGQ para esse gerenciador de filas, reiniciar antes que a recuperação peer-to-peer de fila compartilhada seja concluída. O que, por sua vez, deixa a possibilidade para que qualquer mensagem seja processada antes e fora da sequência com as mensagens ainda sendo recuperadas. Para assegurar se as mensagens não são processadas fora da sequência, o agente IGQ serializa o acesso às filas compartilhadas, emitindo a chamada de API MQCONN.

Uma tentativa, pelo agente IGQ para serializar o acesso às filas compartilhadas embora a recuperação peer-to-peer ainda esteja em andamento possa falhar. Uma mensagem de erro é emitida e o agente IGQ é colocado no estado de nova tentativa. Quando a recuperação peer-to-peer do gerenciador de filas for concluída, por exemplo, no momento da próxima nova tentativa, o agente IGQ poderá ser iniciado.

z/OS

Gerenciamento de armazenamento no z/OS

IBM MQ for z/OS requer estruturas de dados temporários e permanentes e usa os conjuntos de páginas e os buffers de memória para armazenar esses dados. Estes tópicos fornecem detalhes adicionais sobre como o IBM MQ usa os conjuntos de páginas e os buffers.

Conceitos relacionados

[“Conjuntos de páginas para o IBM MQ for z/OS” na página 235](#)

Use este tópico para entender como o IBM MQ for z/OS usa conjuntos de páginas para armazenar mensagens.

[“Classes de armazenamento para o IBM MQ for z/OS” na página 236](#)

Uma classe de armazenamento é um conceito do IBM MQ for z/OS que permite que o gerenciador de filas mapeie filas para conjuntos de páginas. É possível usar as classes de armazenamento para controlar quais conjuntos de dados são usados por quais filas.

[“Buffers e buffer pools para o IBM MQ for z/OS” na página 237](#)

IBM MQ for z/OS usa buffers e buffer pools para dados em cache temporariamente. Use este tópico para entender melhor como os buffers são organizados e usados.

Referências relacionadas

[“Em que local localizar mais informações sobre o gerenciamento de armazenamento para o IBM MQ for z/OS” na página 239](#)

Use este tópico como uma referência para localizar informações adicionais sobre o gerenciamento de armazenamento para IBM MQ for z/OS.

Conjuntos de páginas para o IBM MQ for z/OS

Use este tópico para entender como o IBM MQ for z/OS usa conjuntos de páginas para armazenar mensagens.

Um *conjunto de páginas* é um conjunto de dados linear VSAM que foi formatado especialmente para ser usado pelo IBM MQ. Os conjuntos de páginas são usados para armazenar a maioria das mensagens e definições de objeto.

As exceções a isso são as definições globais, que são armazenadas em um repositório compartilhado no Db2 e as mensagens nas filas compartilhadas. Elas não são armazenadas nos conjuntos de páginas do gerenciador de filas. Para obter mais informações sobre filas compartilhadas, consulte [“Filas compartilhadas e grupos de filas compartilhadas” na página 172](#), e para obter mais informações sobre definições globais, consulte [Definições privadas e globais](#).

Conjuntos de páginas do IBM MQ podem ter até 64 GB de tamanho. Cada conjunto de páginas é identificado por um identificador do conjunto de páginas (PSID), um número inteiro no intervalo de 00 a 99. Cada gerenciador de filas deve ter seu próprio conjunto de páginas.

O IBM MQ usa o conjunto de páginas zero (PSID=00) para armazenar as definições de objeto e outras importantes informações relevantes ao gerenciador de filas. Para a operação normal do IBM MQ é essencial que o conjunto de páginas não fique cheio; portanto, não use-o para armazenar as mensagens.

Para melhorar o desempenho do seu sistema, você também deve separar as mensagens de vida útil curta das mensagens de vida útil longa, colocando-as em diferentes conjuntos de páginas.

Você deve formatar conjuntos de páginas e o IBM MQ fornece um utilitário FORMAT para isto; consulte [Formatando conjuntos de páginas \(FORMAT\)](#). Conjuntos de páginas também devem ser definidos para o subsistema IBM MQ.

O IBM MQ for z/OS pode ser configurado para expandir um conjunto de páginas dinamicamente se ficar cheio. O IBM MQ continua a expandir o conjunto de páginas se necessário até que existam 123 extensões lógicas, se houver espaço de armazenamento em disco suficiente disponível. As extensões podem expandir os volumes se o conjunto de dados linear estiver definido desta maneira; no entanto, o IBM MQ não pode expandir os conjuntos de páginas além de 64 GB.

Não é possível usar os conjuntos de páginas de um gerenciador de filas do IBM MQ em um gerenciador de filas do IBM MQ diferente ou alterar o nome do gerenciador de filas. Se você deseja transferir os dados de um gerenciador de filas para outro, deve descarregar todos os objetos e as mensagens do primeiro gerenciador de filas e recarregá-los para outro.

Não é possível usar os conjuntos de páginas com mais de 4 GB em um gerenciador de filas que executa uma liberação anterior à V6. Durante o período de migração, quando é provável que você precise executar fallback para uma liberação anterior do código:

- Não altere o conjunto de páginas 0 para que tenha mais de 4 GB.
- Outros conjuntos de páginas com mais de 4 GB serão deixados offline ao reiniciar um gerenciador de filas com uma liberação anterior.

Para obter informações adicionais sobre a migração dos conjuntos de páginas existentes capazes de expandir além de 4 GB, consulte [Definindo um Conjunto de Páginas para que Tenha Mais de 4 GB](#).

É possível para um administrador incluir dinamicamente os conjuntos de páginas em um gerenciador de filas em execução ou remover os conjuntos de páginas de um gerenciador de filas em execução (exceto para o conjunto de páginas zero). O comando DEFINE PSID pode ser executado depois que a reinicialização do gerenciador de filas for concluída, apenas se o comando contiver a palavra-chave DSN.

Classes de armazenamento para o IBM MQ for z/OS

Uma classe de armazenamento é um conceito do IBM MQ for z/OS que permite que o gerenciador de filas mapeie filas para conjuntos de páginas. É possível usar as classes de armazenamento para controlar quais conjuntos de dados são usados por quais filas.

Introduzindo Classes de Armazenamento

Uma *classe de armazenamento* mapeia uma ou mais filas para um conjunto de páginas. Isso significa que as mensagens para essa fila são armazenadas nesse conjunto de páginas.

As classes de armazenamento permitem controlar onde os dados de mensagem não compartilhada são armazenados para propósitos de isolamento de aplicativos ou gerenciamento de carregamento e espaço do conjunto de dados administrativo. Também é possível usar as classes de armazenamento para definir o grupo XCF e o nome do membro de uma região IMS se você estiver utilizando a ponte IMS (descrito em [“IBM MQ e IMS” na página 295](#)).

As filas compartilhadas não usam as classes de armazenamento para obter um mapeamento do conjunto de páginas porque as mensagens contidas nelas não são armazenadas nos conjuntos de páginas.

Como as Classes de Armazenamento Trabalham

- Defina uma classe de armazenamento, usando o comando DEFINE STGCLASS, especificando um identificador do conjunto de páginas (PSID).
- Ao definir uma fila, você especifica a classe de armazenamento no atributo STGCLASS.

No exemplo a seguir, a fila local QE5 é mapeada para o conjunto de páginas 21 por meio da classe de armazenamento ARC2.

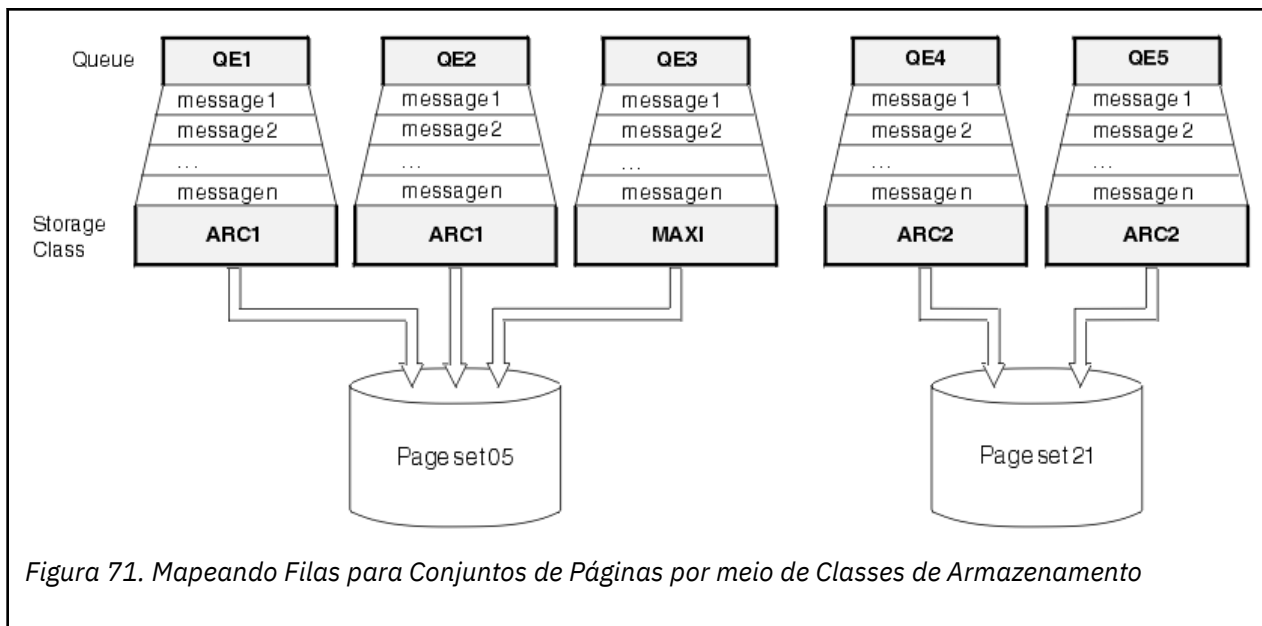
```
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE QLOCAL(QE5) STGCLASS(ARC2)
```

Isso significa que as mensagens colocadas na fila QE5 são armazenadas no conjunto de páginas 21 (se permanecerem na fila por tempo suficiente para serem gravadas para DASD).

Mais de uma fila pode usar a mesma classe de armazenamento e é possível definir quantas classes de armazenamento desejar. Por exemplo, é possível estender o exemplo anterior para incluir mais definições de fila e classe de armazenamento, da seguinte maneira:

```
DEFINE STGCLASS(ARC1) PSID(05)
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE STGCLASS(MAXI) PSID(05)
DEFINE QLOCAL(QE1) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE2) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE3) STGCLASS(MAXI) ...
DEFINE QLOCAL(QE4) STGCLASS(ARC2) ...
DEFINE QLOCAL(QE5) STGCLASS(ARC2) ...
```

No Figura 71 na página 237, as classes de armazenamento ARC1 e MAXI são associadas ao conjunto de páginas 05. Portanto, as filas QE1, QE2 e QE3 são mapeadas para o conjunto de páginas 05. De maneira semelhante, a classe de armazenamento ARC2 associa as filas QE4 e QE5 ao conjunto de páginas 21.



Se você definir uma fila sem especificar uma classe de armazenamento, o IBM MQ usará uma classe de armazenamento padrão.

Se uma mensagem for colocada em uma fila que nomeie uma classe de armazenamento não existente, o aplicativo receberá um erro. Você deve alterar a definição de fila para fornecer a ela um nome de classe de armazenamento existente ou criar a classe de armazenamento nomeada pela fila.

É possível alterar a classe de armazenamento apenas quando:

- Todas as filas que usam essa classe de armazenamento estiverem vazias e não tiverem atividade não confirmada.
- Todas as filas que usam essa classe de armazenamento estiverem fechadas.

z/OS Buffers e buffer pools para o IBM MQ for z/OS

IBM MQ for z/OS usa buffers e buffer pools para dados em cache temporariamente. Use este tópico para entender melhor como os buffers são organizados e usados.

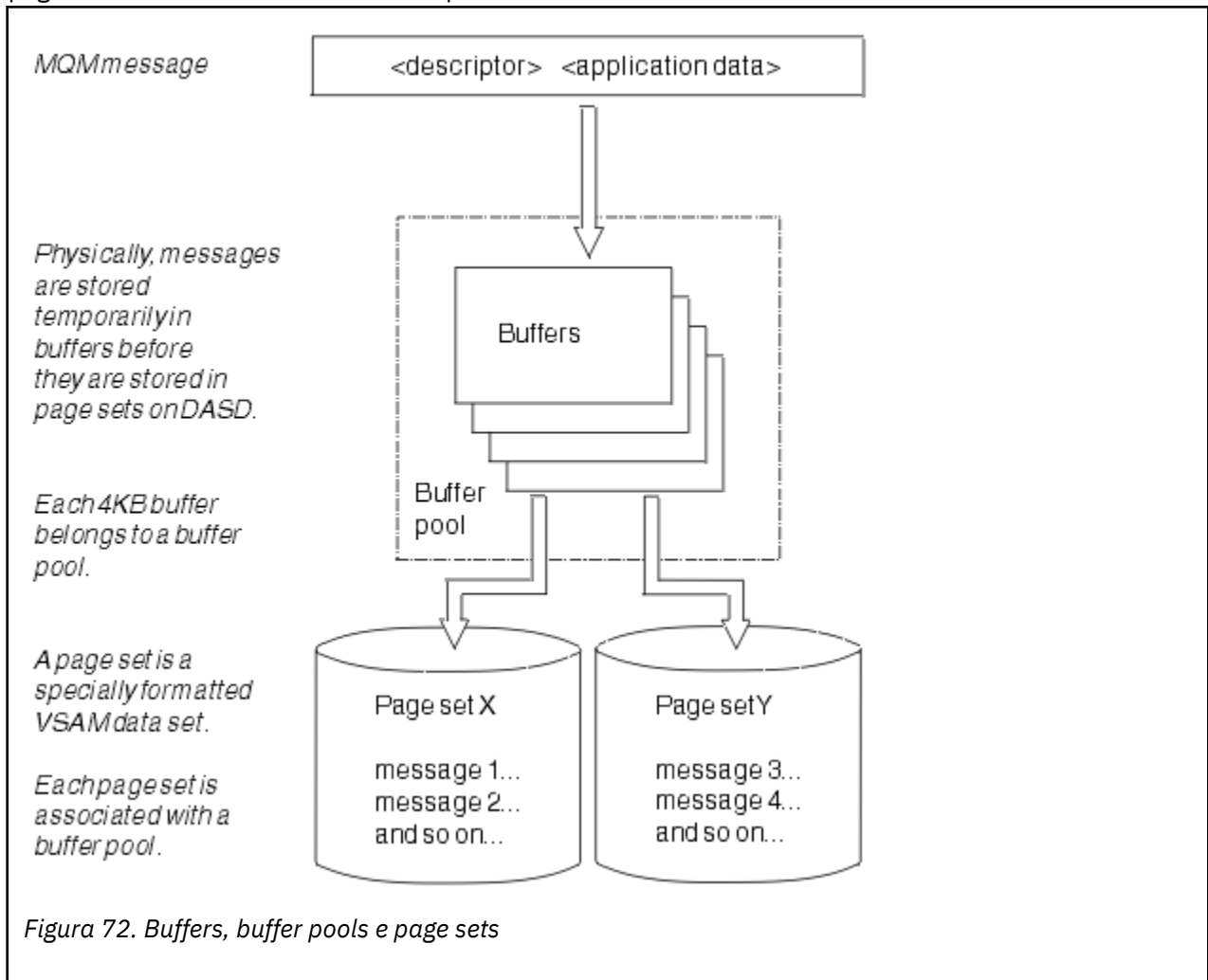
Para eficiência, o IBM MQ usa uma forma de armazenamento em cache em que as mensagens (e definições de objeto) são armazenadas temporariamente em buffers antes de serem armazenadas em conjuntos de páginas no DASD. As mensagens de vida útil curta ou sejam as mensagens que são recuperadas de uma fila pouco depois de serem recebidas, só podem ser armazenadas em buffers. Essa atividade de armazenamento em cache é controlada por um gerenciador de buffer, que é um componente do IBM MQ.

Os buffers estão organizados em *buffer pools*. É possível definir até 100 buffer pools (de 0 a 99) para cada gerenciador de filas.

Recomenda-se utilizar o número mínimo de conjuntos de buffer consistentes com o objeto e a separação do tipo de mensagem delineada em [Figura 72 na página 238](#) e quaisquer requisitos de isolamento de dados que seu aplicativo possa ter. Cada buffer tem 4 KB de comprimento. Os buffer pools usam armazenamento de 31 bits por padrão; neste modo, o número máximo de buffers é determinado pela quantidade de armazenamento de 31 bits disponível no espaço de endereço do gerenciador de filas; não use mais de cerca de 70% para buffers. Como alternativa, a alocação de armazenamento do buffer pool pode ser feita a partir do armazenamento de 64 bits (use o atributo LOCATION do comando **DEFINE**

BUFFPOOL . O uso de LOCATION(ABOVE) para que o armazenamento de 64 bits seja usado tem dois benefícios. Primeiro, há muito mais armazenamento de 64 bits disponível, que os buffer pools podem ficar muito maiores e, segundo, o armazenamento de 31 bits é disponibilizado para uso por outras funções. Geralmente, quanto mais buffers você tiver, mais eficiente será o armazenamento em buffer e melhor o desempenho do IBM MQ.

O Figura 72 na página 238 mostra o relacionamento entre as mensagens, os buffers, os buffer pools e os conjuntos de páginas. Um buffer pool é associado a um ou mais conjuntos de páginas; cada conjunto de páginas é associado a um único buffer pool.



É possível emitir dinamicamente comandos para modificar o tamanho do buffer pool e o local, usando o comando **ALTER BUFFPOOL**. Os conjuntos de páginas podem ser incluídos dinamicamente usando o comando **DEFINE PSID** ou excluídos usando o comando **DELETE PSID**.

Se um buffer pool for muito pequeno, o IBM MQ emitirá a mensagem **CSQP020E**. Em seguida, é possível incluir dinamicamente mais buffers no buffer pool afetado (observe que você pode ter que remover buffers de outros buffer pools para fazer isso).

Especifique o número de buffers em um conjunto com o comando **DEFINE BUFFPOOL** e é possível redimensionar dinamicamente os buffer pools com o comando **ALTER BUFFPOOL**. Determine o número atual de buffers em um conjunto dinamicamente, exibindo um conjunto de páginas que usa o buffer pool, usando o comando **DISPLAY USAGE**.

Por motivo de desempenho, não coloque as definições de objeto e mensagens no mesmo buffer pool. Use um buffer pool (ou seja, número zero) exclusivamente para o conjunto de páginas zero, onde as definições de objeto são mantidas. De maneira semelhante, mantenha as mensagens de vida útil curta e as mensagens de vida útil longa em diferentes buffer pools e, portanto, em diferentes conjuntos de páginas e em diferentes filas.

O comando **DEFINE BUFFPOOL** não pode ser usado após a reinicialização para criar um novo buffer pool. Em vez disso, se um comando **DEFINE PSID** usar a palavra-chave DSN, ele poderá identificar explicitamente um buffer pool que não está atualmente definido. Em seguida, o novo buffer pool será criado.

Em que local localizar mais informações sobre o gerenciamento de armazenamento para o IBM MQ for z/OS

Use este tópico como uma referência para localizar informações adicionais sobre o gerenciamento de armazenamento para IBM MQ for z/OS.

É possível localizar informações adicionais sobre os tópicos nesta seção a partir das seguintes origens:

Tópico	Onde Localizar
Quanto armazenamento você precisa	Planejando seus requisitos de armazenamento e desempenho no z/OS
Qual tamanho tornar seus conjuntos de páginas e buffer pools	Planejar Conjuntos de Páginas e Buffer Pools
Gerenciando os Conjuntos de Páginas	Gerenciando os Conjuntos de Páginas
Comandos MQSC	Os Comandos MQSC

Efetuando login no IBM MQ for z/OS

IBM MQ mantém os *logs* das mudanças de dados e eventos significativos conforme eles ocorrem. Esses logs podem ser usados para recuperar os dados para um estado anterior, se necessário.

O *conjunto de dados de autoinicialização* (BSDS) armazena as informações sobre os conjuntos de dados que contêm os logs.

O log não contém informações para estatísticas, rastreios ou avaliação de desempenho. Para obter detalhes adicionais sobre as informações estatísticas e de monitoramento que o IBM MQ coleta, consulte [Monitoramento e Estatísticas](#).

Para obter informações adicionais sobre a criação de log, consulte os seguintes tópicos:

- [“Arquivos de Log no IBM MQ for z/OS” na página 240](#)
- [“Como o Log Está Estruturado” na página 244](#)
- [“Como os logs do IBM MQ for z/OS são gravados” na página 244](#)
- [“Endereço de bytes relativo de log maior” na página 247](#)
- [“O Conjunto de Dados de Autoinicialização” na página 248](#)

Tarefas relacionadas

[Planejando seu Ambiente de Criação de Log](#)

[Configurando logs usando o módulo de parâmetro do sistema](#)

 [Administrando z/OS](#)

Referências relacionadas

 [Mensagens do IBM MQ for z/OS](#)

Os arquivos de log contêm as informações necessárias para a recuperação de transação. Os arquivos de log ativo podem ser arquivados para que você possa manter os dados de log por um longo período.

O que é Um Arquivo de Log

O IBM MQ registra todos os eventos significantes conforme eles ocorrem em um *log ativo*. O log contém as informações necessárias para recuperar:

- Mensagens persistentes
- Objetos do IBM MQ, tais como filas
- O gerenciador de filas do IBM MQ

O log ativo consiste em uma coleção de conjuntos de dados (até 310) que são usados de maneira cíclica.

É possível ativar o arquivamento de log para que quando um log ativo ficar cheio, uma cópia seja feita em um conjunto de dados do archive. Usar o archive permite manter os dados de log por um período estendido. Se você não usar o archive, os logs serão quebrados e os dados anteriores serão sobrescritos. Para recuperar um conjunto de páginas ou recuperar os dados em uma estrutura CF, você precisa registrar em log os dados de quando foi feito backup do conjunto de páginas ou da estrutura. Um log de archive pode ser criado no disco ou na fita.

Arquivando

Como o log ativo tem um tamanho fixo, o IBM MQ copia o conteúdo de cada conjunto de dados do log periodicamente em um *log de archive*, o qual normalmente é um conjunto de dados em um dispositivo de armazenamento de acesso direto (DASD) ou uma fita magnética. Se houver uma falha de subsistema ou transação, o IBM MQ usará o log ativo e, se necessário, o log de archive para recuperação.

O log de archive pode conter até 1000 conjuntos de dados sequenciais. É possível catalogar cada conjunto de dados usando o recurso de catálogo integrado (ICF) do z/OS.

Archive é um componente essencial de recuperação do IBM MQ. Se uma unidade de recuperação for de longa execução, os registros de log nessa unidade de recuperação podem ser localizados no log de archive. Neste caso, a recuperação requer dados do log de archive. No entanto, se o arquivamento for desativado, o log ativo com novos registros de log será agrupado, sobrescrevendo os registros de log anteriores. Isto significa que o IBM MQ pode não conseguir voltar a unidade de recuperação e as mensagens podem ser perdidas. O gerenciador de filas então finaliza de maneira anormal.

Portanto, em um ambiente de produção, **nunca desative o archive**. Se fizer isso, você corre o risco de perder os dados após uma falha do sistema ou da transação. Apenas se você estiver executando em um ambiente de teste poderá considerar a desativação do archive. Se precisar fazer isso, use a macro CSQ6LOGP, que é descrita em [Usando CSQ6LOGP](#).

Para ajudar a evitar problemas com unidades de trabalho de execução longa não planejadas, o IBM MQ emite uma mensagem (CSQJ160I ou CSQJ161I) se uma unidade de trabalho de execução longa é detectada durante o processamento de transferência de log ativo.

Duas Criações de Log

Nas duas criações de log, cada registro de log é gravado para dois conjuntos de dados de log ativos diferentes para minimizar a probabilidade de problemas de perda de dados durante a reinicialização.

É possível configurar o IBM MQ para executar com *criação de log única* ou *duas criações de log*. Com uma criação de log, os registros de log são gravados uma vez para um conjunto de dados de log ativo. Cada conjunto de dados de log ativo é um conjunto de dados linear (LDS) VSAM de extensão única. Com duas criações de log, cada registro de log é gravado em dois conjuntos de dados de log ativos diferentes. Duas criações de log minimizam a probabilidade de problema de perda de dados durante a reinicialização.

Desvio de Log

O desvio de log faz com que os registros de log para algumas unidades de trabalho sejam gravados mais adiante no log. Isso reduz a quantidade de dados do log que devem ser lidos na reinicialização do gerenciador de filas ou na volta, para unidade de trabalho em dúvida de longo prazo ou de longa execução.

Quando uma unidade de trabalho é considerada longa, uma representação de cada registro de log é gravada mais adiante no log. Essa técnica é conhecida como *desvio*. Quando a unidade de trabalho inteira tiver sido processada, a unidade de trabalho estará em um estado de *desvio*. Qualquer atividade de reinicialização ou volta relacionada à unidade de trabalho de desvio pode usar os registros de log de desvio em vez de usar os registros de log originais da unidade de trabalho.

A detecção de uma unidade de trabalho de longa execução é uma função do processo de ponto de verificação. No momento do ponto de verificação, cada unidade de trabalho ativa é verificada para estabelecer se precisa ser desviada. Se a unidade de trabalho ocorreu entre dois pontos de verificação anteriores desde quando foi criada ou desde quando foi enviada por último, a unidade de trabalho estará adequada para ser desviada. Isso significa que uma única unidade de trabalho pode ter sido desviada mais de uma vez. Isso é conhecido como uma unidade de trabalho de *vários desvios*.

Uma unidade de trabalho é desviada a cada três pontos de verificação. No entanto, o ponto de verificação é executado de forma assíncrona para o comutador de log (ou a composição do registro de log que fez com que LOGLOAD excedesse).

Há somente um único ponto de verificação ocorrendo de cada vez, portanto, pode haver múltiplos comutadores de log antes da conclusão de um ponto de verificação.

Isso significa que, se não houver logs ativos o suficiente ou se forem muito pequenos, o desvio de uma grande unidade de trabalho poderá não ser concluído antes que todos os logs sejam preenchidos.

A mensagem [CSQR027I](#) resultará se o desvio não puder ser concluído.

Se o arquivamento de log estiver desativado, ocorrerá ABEND 5C6 com o motivo 00D1032A, se houver uma tentativa de voltar a unidade de trabalho da qual o desvio falhou. Para evitar esse problema, é necessário usar OFFLOAD=YES.

O desvio de log é sempre ativo e é executado independentemente de o archive de log ser ativado ou não.

Nota: Embora todos os registros de log para uma unidade de trabalho sejam desviados, o conteúdo inteiro de cada registro não é desviado, apenas a parte que é necessária para volta. Isso significa que a quantidade de dados de log gravados é mantida no mínimo e que os registros desviados não podem ser usados se ocorrer uma falha no conjunto de páginas. Uma unidade de trabalho de longa execução é aquela que ficou em execução por mais de três pontos de verificação do gerenciador de filas.

Para obter mais informações sobre o desvio de log, veja [Gerenciando os logs](#).

Compactação de Log

É possível configurar o IBM MQ for z/OS para compactar e descompactar registros de log conforme eles são gravados e lidos a partir do conjunto de dados do log.

A compactação de log pode ser usada para reduzir a quantidade de dados gravados no log para mensagens persistentes em filas privadas. A quantidade de compactação que é atingida depende do tipo de dados contidos nas mensagens. Por exemplo Executar Codificação de Comprimento (RLE) trabalha compactando as instâncias repetidas de bytes que podem fornecer bons resultados de maneira eficiente para os dados orientados ao registro ou estruturados.



Atenção: Mensagens persistentes que estão sendo colocadas em uma fila compartilhada não estão sujeitas a compactação de log.

É possível usar os campos na seção do Gerenciador de Log dos registros de Recurso de Gerenciamento do Sistema 115 (SMF) para monitorar quanta compactação de dados é atingida. Para obter mais informações sobre o SMF, consulte [Usando o Recurso de Gerenciamento do Sistema](#) e [Mensagens de contabilidade e de estatísticas](#).

A compactação de log aumenta a utilização do processador do sistema. Você apenas deve considerar usar a compactação se o rendimento do seu gerenciador de filas for restrito pela largura da banda de E/S para os conjuntos de dados de log ou se estiver restrito pelo armazenamento em disco necessário para reter os conjuntos de dados de log. Se você estiver usando filas compartilhadas, as restrições de largura da banda de E/S podem ser aliviadas incluindo gerenciadores de filas adicionais no grupo de filas compartilhadas e distribuindo a carga de trabalho em mais gerenciadores de filas.

A opção de compactação de log pode ser ativada e desativada conforme necessário, sem precisar parar e reiniciar o gerenciador de filas. O gerenciador de filas pode ler qualquer registro de log compactado independente da configuração de compactação de log atual.

O gerenciador de filas suporta 3 configurações para a compactação de log.

Nenhum

Nenhuma compactação de dados de log é usada. Esse é o valor-padrão.

RLE

A compactação de dados de log é executada usando a codificação RLE.

QUALQUER

Ative o gerenciador de filas para selecionar o algoritmo de compactação que fornece o mais alto grau de compactação de registro de log. Esta opção resulta na compactação de RLE.

É possível controlar a compactação de registros de log usando um dos seguintes:

- Os comandos SET e DISPLAY LOG no MQSC; consulte [SET LOG](#) e [DISPLAY LOG](#)
- As funções Configurar log e Consultar log na interface PCF; consulte [Configurar Log](#) e [Consultar Log](#)
- A macro CSQ6LOGP no módulo de parâmetro do sistema; consulte [Usando CSQ6LOGP](#)

Além da Impressão de Log, o utilitário CSQ1LOGP tem suporte para expandir quaisquer registros de log compactados.

Dados do Log

O log pode conter até 18 milhões (1.8×10^{19}) de bytes. Cada byte pode ser endereçado por seu deslocamento do início do log e esse deslocamento é conhecido como seu *endereço de bytes relativo* (RBA).

O RBA é referenciado por um campo de 6 bytes ou 8 bytes fornecendo um intervalo total endereçável de 2^{48} bytes ou 2^{64} bytes, dependendo de se RBAs de log de 6 bytes ou 8 bytes estão em uso.

No entanto, quando o IBM MQ detecta que o intervalo usado está além de F00000000000 (se RBAs de 6 bytes estão em uso) ou FFFF800000000000 (se RBAs de log de 8 bytes estão em uso), as mensagens [CSQI045](#), [CSQI046](#), [CSQI047](#) e [CSQJ032](#) são emitidas, avisando para reconfigurar o RBA de log.

Se o valor de RBA atingir FFF800000000 (se RBAs de log de 6 bytes estão em uso) ou FFFFFFFC0000000000 (se RBAs de log de 8 bytes estão em uso), o gerenciador de filas será finalizado com o código de razão [00D10257](#).

Quando as mensagens de aviso sobre o intervalo de logs usado estiverem sendo emitidas, você deverá planejar uma indisponibilidade do gerenciador de filas durante a qual o gerenciador de filas pode ser convertido para usar RBAs de log de 8 bytes ou o log poderá ser reconfigurado. O procedimento para reconfigurar o log está documentado em [Reconfigurando o log do gerenciador de filas](#).

Se o gerenciador de filas estiver usando RBAs de log de 6 bytes, considere converter o gerenciador de filas para usar RBAs de log de 8 bytes em vez de reconfigurar o log do gerenciador de filas, seguindo o procedimento documentado em [Implementando o maior Relative Byte Address de log](#).

O log consiste em *registros de log*, cada um dos quais é um conjunto de dados de log tratado como uma única unidade. Um registro de log é identificado pelo RBA do primeiro byte de seu cabeçalho ou por seu número de sequência do registro de log (LRSN). O RBA ou LRSN identifica exclusivamente um registro que é iniciado em um determinado ponto no log.

Se você usa o RBA ou LRSN para identificar os pontos de log, depende de se você está usando grupos de filas compartilhadas. Em um ambiente de compartilhamento de fila, não é possível usar o endereço de byte relativo para identificar exclusivamente um ponto de log, porque diversos gerenciadores de filas podem atualizar a mesma fila ao mesmo tempo e cada um tem seu próprio log. Para resolver isso, o número de sequência do registro de log é derivado a partir de um valor de registro de data e hora e não representa necessariamente o deslocamento físico do registro de log no log.

Cada registro de log possui um cabeçalho que fornece seu tipo, o subcomponente do IBM MQ que fez o registro e, para registros da unidade de recuperação, um identificador da unidade de recuperação.

Existem quatro tipos de registro de log, descritos nos seguintes títulos:

- [Registros de log da unidade de recuperação](#)
- [Registros de ponto de verificação](#)
- [Registros de controle do conjunto de páginas](#)
- [Registros de backup da estrutura de CF](#)

Registros de Log da Unidade de Recuperação

A maioria dos registros de log descreve mudanças nas filas do IBM MQ. Todas essas mudanças são feitas nas unidades de recuperação.

O IBM MQ usa técnicas de criação de log especiais que envolvem *desfazer/refazer* e *registros de log de compensação* para reduzir os tempos de reinicialização e melhorar a disponibilidade do sistema.

Um efeito disso é que o tempo de reinicialização é limitado. Se ocorrer uma falha durante a reinicialização para que o gerenciador de filas tenha que ser reiniciado uma segunda vez, toda a atividade de recuperação que foi concluída no ponto de falha na primeira reinicialização não precisará ser reaplicada durante uma segunda reinicialização. Isso significa que sucessivas reinicializações não demoram progressivamente mais tempo para serem concluídas.

Registros de ponto de verificação

Para reduzir o tempo de reinicialização, o IBM MQ executa pontos de verificação periódicos durante a operação normal. Eles ocorrem da seguinte maneira:

- Quando um número predefinido de registros de logs tiver sido gravado. Esse número é definido pelo operando de frequência de ponto de verificação denominada LOGLOAD da macro do parâmetro do sistema CSQ6SYSP, descrita em [Usando o CSQ6SYSP](#).
- No fim de uma reinicialização bem-sucedida.
- No término normal.
- Sempre que o IBM MQ alterna para o próximo conjunto de dados de log ativo no ciclo.

No momento que um ponto de verificação é obtido, o IBM MQ emite o comando DISPLAY CONN (descrito em [DISPLAY CONN](#)) internamente para que uma lista de conexões atualmente em dúvida sejam gravadas no log do console do z/OS.

Registros de Controle do Conjunto de Páginas

Estes registros registram os conjuntos de páginas e buffer pools conhecidos no gerenciador de filas do IBM MQ em cada ponto de verificação, e registram informações sobre os intervalos de log requeridos para executar a recuperação de mídia do conjunto de páginas no momento do ponto de verificação.

Certas mudanças dinâmicas nos conjuntos de páginas e buffer pools também são gravadas como registros de controle do conjuntos de páginas, de modo que as mudanças possam ser recuperadas e automaticamente restabelecidas na próxima reinicialização do gerenciador de filas.

registros de backup de estrutura CF

Esses registros retêm dados lidos de uma estrutura de lista de recursos de acoplamento em resposta a um comando BACKUP CFSTRUCT. No improvável evento de falha da estrutura do recurso de acoplamento, esses registros são usados, junto com os registros da unidade de recuperação, pelo comando RECOVER CFSTRUCT para executar uma recuperação de mídia da estrutura do recurso de acoplamento para o ponto da falha.

Tarefas relacionadas

[Implementando o maior Endereço de Bytes Relativo de log](#)

Como o Log Está Estruturado

Use este tópico para entender a terminologia usada para descrever os registros de log.

Cada conjunto de dados de log ativo deve ser um conjunto de dados linear (LDS) VSAM. A unidade de saída física gravada no conjunto de dados de log ativo é um intervalo de controle (CI) de 4 KB. Cada CI contém um registro VSAM.

Registros de Log Físico e Lógico

Um VSAM CI é um registro *físico*. As informações registradas em log em um determinado horário formam um registro *lógico*, com um comprimento que varia independentemente do espaço disponível no CI. Portanto, um registro físico pode conter:

- Diversos registros lógicos
- Um ou mais registros lógicos e parte de outro registro lógico
- Parte de um registro lógico apenas

O termo *registro de log* se refere ao registro *lógico*, independentemente de quantos registros *físicos* são necessários para armazená-lo.

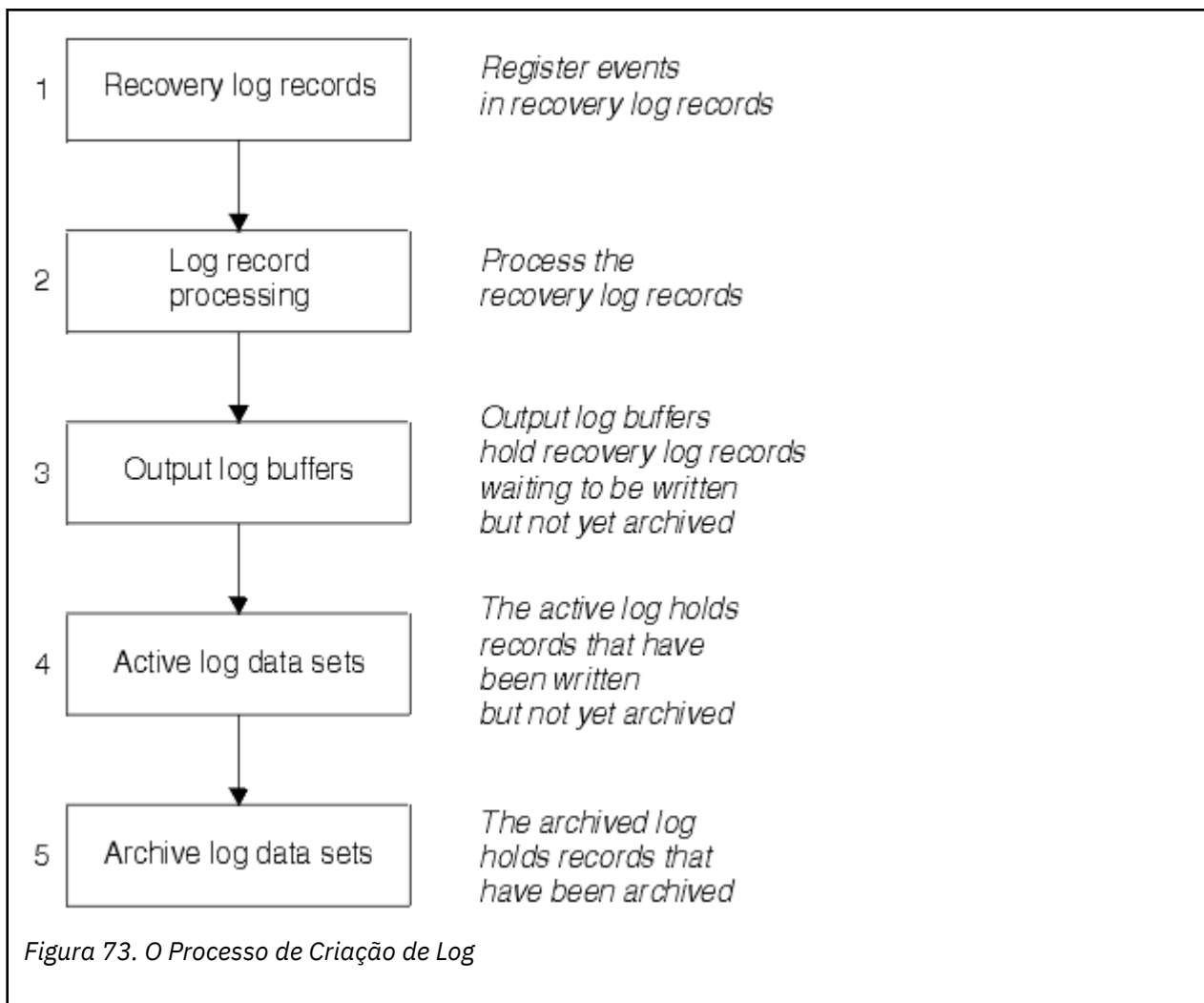
Como os logs do IBM MQ for z/OS são gravados

Use este tópico para entender como o IBM MQ processa registros de arquivos de log.

IBM MQ grava cada registro de log para um conjunto de dados DASD denominado *log ativo*. Quando o log ativo estiver cheio, o IBM MQ copiará seu conteúdo para um DASD ou conjunto de dados de fita denominado *log de archive*. Esse processo é denominado *transferência*.

[Figura 73 na página 245](#) ilustra o processo de criação de log. Os registros de log geralmente atravessam o seguinte ciclo:

1. IBM MQ observa mudanças nos dados e eventos significativos nos registros do log de recuperação.
2. IBM MQ processa os registros de log de recuperação e os divide em segmentos, se necessário.
3. Os registros de log são colocados sequencialmente nos *buffers do log de saída*, que são formatados como Intervalos de Controles (CI) VSAM. Cada registro de log é identificado por um endereço de bytes relativo no intervalo de zero a $2^{64}-1$.
4. Os CIs são gravados em um conjunto de dados de log ativo DASD predefinido, que é usado sequencialmente e reciclado
5. Se o arquivamento estiver ativo, cada conjunto de dados de log ativo ficará cheio, seu conteúdo é automaticamente transferido para um novo conjunto de dados do log de archive.



Quando o Log Ativo é Gravado

Os buffers de log em armazenamento são gravados para um conjunto de dados de log ativo sempre que ocorrer algum dos seguintes:

- Os buffers de log ficarem cheios.
- O limite de gravação for atingido (conforme especificado na macro CSQ6LOGP).
- Ocorrer certos eventos significativos, como um ponto de confirmação ou quando um comando IBM MQ BACKUP CFSTRUCT é emitido.

Quando o gerenciador de filas é inicializado, os conjuntos de dados de log ativo nomeados no BSDS são alocados dinamicamente para uso exclusivo do gerenciador de filas e permanecem alocados exclusivamente para IBM MQ até que o gerenciador de filas seja finalizado.

Incluindo Dinamicamente os Conjuntos de Dados do Log

É possível definir dinamicamente novos conjunto de dados de log ativo enquanto o gerenciador de filas estiver em execução. Este recurso alivia o problema de uma interrupção do gerenciador de filas quando o arquivamento não conseguir transferir os logs ativos devido a um problema transitório. Consulte o comando [DEFINE LOG](#) para obter mais informações.

Nota: Para redefinir ou remover os logs ativos, você deve finalizar e reiniciar o gerenciador de filas.

IBM MQ e Storage Management Subsystem

Os parâmetros do IBM MQ permitem especificar classes de armazenamento do Storage Management Subsystem (MVS/DFP SMS) ao alocar dinamicamente os conjuntos de dados de log de archive do IBM MQ. IBM MQ inicia os conjuntos de dados do log de archive, mas é possível usar SMS para executar a alocação do conjunto de dados de archive.

Referências relacionadas

[“Quando o log de archive do IBM MQ for z/OS é gravado”](#) na página 246

Use este tópico para entender o processo de copiar os logs ativos para os logs de archive e quando o processo ocorre.

z/OS Quando o log de archive do IBM MQ for z/OS é gravado

Use este tópico para entender o processo de copiar os logs ativos para os logs de archive e quando o processo ocorre.

O processo de copiar os logs ativos para arquivar os logs é denominado *transferência*. A relação da transferência para outros eventos de criação de log é mostrada esquematicamente em [Figura 74](#) na página 246.

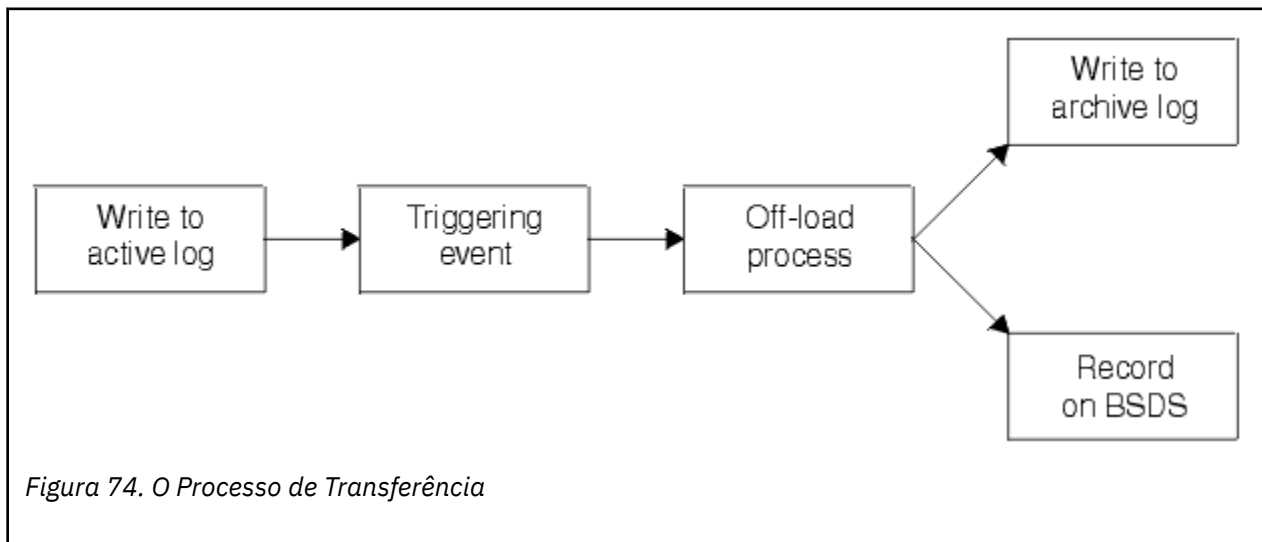


Figura 74. O Processo de Transferência

Acionando o Processo de Transferência

O processo de transferência de um log ativo para um log de archive pode ser acionado por diversos eventos. Por exemplo:

- Preenchendo um conjunto de dados de log ativo.
- Usando o comando MQSC ARCHIVE LOG.
- Um erro que ocorre ao gravar para um conjunto de dados de log ativo.

O conjunto de dados é truncado antes do ponto de falha e o registro que não foi gravado se torna o primeiro registro do novo conjunto de dados. A transferência é acionada para o conjunto de dados truncado como para um conjunto de dados do log integral normal. Se houver logs ativos duplos, ambas as cópias serão truncadas para que as duas cópias permaneçam sincronizadas.

A mensagem CSQJ110E é emitida quando o último log disponível ativo estiver 5% cheio e em incrementos de 5% a partir desse momento, indicando a porcentagem da capacidade de log que está

em uso. Se todos os logs ativos ficarem cheios, o IBM MQ para o processamento, até que ocorra a transferência e emite essa mensagem:

```
CSQJ111A +CSQ1 OUT OF SPACE IN ACTIVE LOG DATA SETS
```

O Processo de Transferência

Quando todos os logs ativos ficarem cheios, o IBM MQ executa o processo de transferência e para o processamento até que o processo de transferência tenha sido concluído. Se o processamento de transferência falhar quando os logs ativos ficarem cheios, o IBM MQ encerra de forma anormal.

Quando um log ativo estiver pronto para ser transferido, uma solicitação será enviada para o operador de console do z/OS para montar uma fita ou preparar uma unidade DASD. O valor da opção de criação de log ARCWTOR (para obter informações adicionais, consulte [Usando CSQ6ARVP](#)) determina se a solicitação foi recebida. Se você estiver usando a fila para a transferência, especifique ARCWTOR=YES. Se o valor for YES, a solicitação será precedida por um WTOR (número de mensagem CSQJ008E) informando ao operador para preparar um conjunto de dados do log de archive para ser alocado.

O operador não precisa responder a essa mensagem imediatamente. No entanto, atrasar a resposta atrasa o processo de transferência. Isso não afeta o desempenho do IBM MQ, a menos que o operador atrase a resposta enquanto o IBM MQ é executado fora dos logs ativos.

O operador pode responder cancelando o processo de transferência. Neste caso, se a alocação for para a primeira cópia de conjuntos de dados do archive dual, o processo de transferência será simplesmente atrasado até que o próximo conjunto de dados de log ativo fique cheio. Se a alocação for para a segunda cópia, o processo de archive alterará para o modo de cópias simples, mas apenas para este conjunto de dados.

Interrupções e Erro durante a Transferência

Uma solicitação para parar o gerenciador de filas não entra em vigor até que o processamento de transferência seja concluído. Se o IBM MQ falhar enquanto a transferência estiver em andamento, a transferência iniciará novamente quando o gerenciador de filas for reiniciado.

Mensagens durante o Processamento de Transferência

As mensagens transferidas são enviadas para o console do z/OS pelo IBM MQ e o processo de transferência. É possível usar essas mensagens para localizar os intervalos de RBA nos vários conjuntos de dados de log.

Endereço de bytes relativo de log maior

Essa função melhora a disponibilidade do gerenciador de filas, aumentando o período de tempo antes de você precisar reconfigurar o log.

Os dados de recuperação são gravados no log para que mensagens persistentes estejam disponíveis quando o gerenciador de filas for reiniciado. O termo Relative Byte Address de log (RBA de log) é utilizado para fazer referência ao local dos dados como um deslocamento do início do log.

Antes da IBM MQ 8.0, o RBA de log de 6 bytes podia tratar até 256 terabytes de dados. Antes desta quantidade de registros de log ser gravada, você precisa reconfigurar o log do gerenciador de filas, seguindo o procedimento documentado em [Reconfigurando o log do gerenciador de filas](#).

A reconfiguração dos logs de gerenciadores de filas não é um processo rápido e pode requerer uma indisponibilidade estendida, devido à necessidade de reconfigurar os conjuntos de páginas como parte do processo. Para um gerenciador de filas de alto uso, esta operação geralmente pode ser feita uma vez por ano.

No IBM MQ 8.0, o RBA de log pode ter 8 bytes de comprimento e o gerenciador de filas pode agora tratar mais de 64.000 vezes a quantidade de dados (16 exabytes) antes de o RBA de log precisar ser

reconfigurado. O impacto do uso do RBA de log maior é que o tamanho dos dados do log gravados aumenta em alguns bytes.

Quando essa função está ativada?

V 9.3.0 Os gerenciadores de filas criados no IBM MQ 9.3.0 ou mais recente já possuem essa função ativada.

Se o RBA de log atual estiver se aproximando do final do intervalo de RBA de log, considere converter o gerenciador de filas para usar um RBA de log de 8 bytes em vez de reconfigurar o log do gerenciador de filas. A conversão de um gerenciador de filas para usar RBAs de log de 8 bytes requer uma indisponibilidade mais curta do que reconfigurar o log, e aumenta significativamente o período de tempo antes que você precise reconfigurar o log.

A mensagem [CSQJ034I](#), emitida durante a inicialização do gerenciador de filas, indica o final do intervalo de RBA do log do gerenciador de filas, conforme configurado, e pode ser usada para determinar se RBAs de log de 6 ou 8 bytes estão em uso.

Como esta função é ativada?

O RBA de log de 8 bytes é ativado ao iniciar o gerenciador de filas com um BSDS no formato da versão 2. Em resumo, ela é obtida:

1. Assegurando que todos os gerenciadores de filas no grupo de filas compartilhadas atendam aos requisitos para ativar o RBA de log de 8 bytes
2. Encerrando o gerenciador de filas corretamente
3. Executando o utilitário de conversão BSDS para criar uma cópia do BSDS no formato da versão 2.
4. Reiniciando o gerenciador de filas com o BSDS convertido.

Após um gerenciador de filas ter sido convertido para usar RBAs de log de 8 bytes, ele não pode voltar a usar o RBA de log de 6 bytes.

Consulte [Implementando o maior endereço de bytes relativo de log](#) para o procedimento detalhado sobre como ativar RBAs de log de 8 bytes.

Tarefas relacionadas

[Planejando para aumentar o intervalo de log endereçável máximo](#)

Referências relacionadas

[O utilitário de conversão conjunto de dados de autoinicialização \(CSQJUCNV\)](#)

z/OS O Conjunto de Dados de Autoinicialização

O conjunto de dados de autoinicialização é requerido pelo IBM MQ como um mecanismo para referenciar conjuntos de dados do log e registros de log. Essas informações são necessárias durante o processamento normal e recuperação de reinicialização.

Qual a Função do Conjunto de Dados de Autoinicialização

O *conjunto de dados de autoinicialização* (BSDS) é um conjunto de dados de chave em sequência VSAM (KSDS) que retém as informações necessárias pelo IBM MQ. Ele contém o seguinte:

- Um inventário de todos os conjuntos de dados de logs ativos e arquivados conhecidos para o IBM MQ. IBM MQ usa este inventário para:
 - Rastrear os conjuntos de dados do log de archive e ativo
 - Localizar os registros de log para que possam satisfazer solicitações de leitura de log durante o processamento normal
 - Localizar os registros de log que possam manipular o processamento de reinicialização

IBM MQ armazena as informações no inventário sempre que um conjunto de dados do log de archive for definido ou um conjunto de dados de log ativo for reutilizado. Para logs ativos, o inventário mostra quais estão cheios e quais estão disponíveis para reutilização. O inventário retém o endereço de bytes relativo (RBA) de cada parte do log retido no conjunto de dados.

- Um inventário *wrap-around* de toda atividade recente do IBM MQ. Isso será necessário, se você tiver que reiniciar o gerenciador de filas.

O BSDS será necessário se o gerenciador de filas tiver um erro e você tiver que reiniciá-lo. IBM MQ **deve** ter um BSDS. Para minimizar a probabilidade de problemas durante a reinicialização, é possível configurar o IBM MQ com BSDSs dual, cada um gravando as mesmas informações. O uso de BSDSs duais é conhecido pela execução no *modo dual*. Se possível, coloque as cópias em volumes separados. Isso reduz o risco de serem perdidos se o volume for corrompido ou destruído. Use BSDSs duais em vez da gravação dual para DASD.

O BSDS é configurado quando o IBM MQ for customizado e você puder gerenciar o inventário usando o utilitário de inventário de log de mudanças (CSQJU003). Para obter informações adicionais sobre este utilitário, consulte [Administrando IBM MQ for z/OS](#). Isso é referenciado por uma instrução DD no procedimento de inicialização do gerenciador de filas.

Normalmente, o IBM MQ mantém cópias duplicadas do BSDS. Se ocorrer um erro de E/S, ele desalocará a cópia com falha e continuará com um único BSDS. É possível restaurar a operação de modo dual, isso é descrito no [Administrando IBM MQ for z/OS](#).

Os logs ativos são registrados primeiro no BSDS quando o IBM MQ está instalado. Não é possível substituir os logs ativos sem terminar e reiniciar o gerenciador de filas.

Os conjuntos de dados do log de archive são alocados dinamicamente. Quando um é alocado, o nome do conjunto de dados é registrado no BSDS. A lista de conjuntos de dados do log de archive é expandida conforme os archives são incluídos e agrupada quando um número determinado pelo usuário de entradas tiver sido atingido. O número máximo de entradas é 1000 para uma única criação de log de archive e 2000 para duas criações de log.

É possível usar um sistema de gerenciamento de fita para excluir os conjuntos de dados do log de archive (IBM MQ não tem um método automatizado). Portanto, as informações sobre um conjunto de dados do log de archive podem estar no BSDS muito depois que o conjunto de dados do log de archive tiver sido excluído pelo administrador do sistema.

Por outro lado, o número máximo de conjuntos de dados de log de archive poderia ter sido excedido e os dados do BSDS eliminados bem antes de o conjunto de dados ter atingido sua data de expiração.

É possível usar o seguinte comando MQSC para determinar a extensão do log e o nome do conjunto de dados do log de archive ou ativo retendo o mais antigo log RBA, necessário para diversos tipos de mídia ou recuperação do gerenciador de filas:

```
DISPLAY USAGE TYPE(DATASET)
```

Se o módulo de parâmetro do sistema especificar que os conjuntos de dados do log de archive são catalogados quando alocados, o BSDS apontará para o catálogo do recurso de catálogo integrado (ICF) para as informações necessárias para as alocações mais recentes. Caso contrário, as entradas BSDS para cada volume registram o número de série do volume e as informações de unidade necessárias para as alocações mais recentes.

A versão de BSDS

O formato do BSDS varia de acordo com a sua versão. Aumentar a versão do BSDS permite que novos recursos sejam usados. As seguintes versões de BSDS são suportadas pelo IBM MQ:

Versão 1

Suportada por todas as liberações do IBM MQ. Um BSDS versão 1 suporta valores do RBA de log de 6 bytes.

Versão 2

Suportado pelo IBM MQ 8.0 e mais recente Um BSDS versão 2 permite valores do RBA de log de 8 bytes e até 310 conjuntos de dados em cada cópia de log ativo.

V 9.3.0 Ativado por padrão para gerenciadores de filas criados em IBM MQ 9.3.0 e posterior.

Versão 3

Suportado pelo IBM MQ 8.0 e mais recente O BSDS é convertido automaticamente na versão 3, a partir da versão 2, quando mais de 31 conjuntos de dados são incluídos na cópia de log ativa.

É possível determinar a versão de um BSDS executando o utilitário de mapa de log de impressão (CSQJU004). Para converter um BSDS da Versão 1 para a Versão 2, execute o utilitário de conversão BSDS ([CSQJUCNV](#)).

Consulte “Endereço de bytes relativo de log maior” na [página 247](#) para obter mais informações sobre RBAs de log de 6 bytes e 8 bytes.

Conjuntos de Dados do Log de Archive e Cópias BSDS

Sempre que um novo conjunto de dados do log de archive é criado, uma cópia do BSDS também é criada. Se o log de archive estiver na fita, o BSDS será o primeiro conjunto de dados no primeiro volume de saída. Se o log de archive estiver no DASD, o BSDS será um conjunto de dados separado.

Os nomes do conjunto de dados do log de archive e da cópia BSDS são os mesmos, exceto se o qualificador de menor nível do nome de log de archive começar com A e a cópia BSDS começar com B, por exemplo:

Nome do Log de Archive

CSQ.ARCHLOG1.E00186.T2336229. A 0000001

Nome da Cópia BSDS

CSQ.ARCHLOG1.E00186.T2336229. B 0000001

Se ocorrer um erro de leitura durante a cópia do BSDS, a cópia não será criada, a mensagem [CSQJ125E](#) será emitida e a transferência para o novo conjunto de dados do log de archive continuará sem a cópia BSDS.

z/OS

Definição do sistema em z/OS

IBM MQ for z/OS usa muitas definições de objeto padrão, e fornece JCL de amostra para criar esses objetos padrão. Use este tópico para entender esses objetos padrão e o JCL de amostra.

Configurando Parâmetros do Sistema

No IBM MQ for z/OS, um módulo de parâmetro do sistema controla a criação de, archive, rastreo e ambientes de conexão que o IBM MQ usa em sua operação. Os parâmetros do sistema são especificados por três macros assembler, da seguinte maneira:

CSQ6SYSP

Os parâmetros do sistema, incluindo a configuração da conexão e ambiente de rastreo.

CSQ6LOGP

Parâmetros de criação de log.

CSQ6ARVP

Parâmetros de archive de log.

Os módulos de parâmetro padrão são fornecidos com o IBM MQ for z/OS. Se eles não contiverem os valores que você deseja usar, é possível criar seus próprios módulos de parâmetro usando a amostra fornecida com IBM MQ. A amostra é `thlqua1.SCSQPROC (CSQ4ZPRM)`.

É possível alterar alguns parâmetros do sistema enquanto um gerenciador de filas estiver em execução. Consulte os comandos SET SYSTEM, SET LOG e SET ARCHIVE nos [Comandos MQSC](#).

Para obter informações adicionais sobre a definição, consulte os seguintes tópicos:

- [“Definindo objetos do sistema para o IBM MQ for z/OS” na página 251](#)
- [“Ajustando o seu gerenciador de filas no IBM MQ for z/OS” na página 255](#)
- [“Definições de amostra fornecidas com o IBM MQ for z/OS” na página 257](#)

Conceitos relacionados

[Customize os conjuntos de dados de entrada de inicialização de amostra](#)

[Origens das quais é possível emitir comandos MQSC e PCF no IBM MQ for z/OS](#)

Tarefas relacionadas

[Administrando z/OS](#)

[Configurando os clusters](#)

[Monitoramento IBM MQ](#)

z/OS Definindo objetos do sistema para o IBM MQ for z/OS

IBM MQ for z/OS requer objetos predefinidos adicionais para aplicativos de publicação/assinatura, cluster e controle de canal, além de outras funções de administração do sistema.

Os objetos do sistema requeridos pelo IBM MQ for z/OS podem ser divididos nas seguintes categorias:

- [Objetos de Publicação/assinatura](#)
- [Objetos padrão do sistema](#)
- [Objetos do comando do sistema](#)
- [Objetos de administração do sistema](#)
- [Filas de canais](#)
- [Filas de clusters](#)
- [Filas de grupos de filas compartilhadas](#)
- [Classes de armazenamento](#)
- [Definindo a fila de mensagens não entregues de objeto do sistema](#)
- [Fila de transmissão padrão](#)
- [Filas internas](#)
- [“Fila de Autenticação de Canal” na página 255](#)

Objetos de Publicação/Assinatura

Há vários objetos do sistema que você precisa definir antes de poder usar aplicativos de publicação/assinatura com o IBM MQ for z/OS. Definições de amostra são fornecidas com o IBM MQ para ajudá-lo a definir esses objetos. Essas amostras estão descritas em [CSQ4INSG](#).

Para usar a publicação/assinatura você precisa definir os seguintes objetos:

- Uma fila local denominada SYSTEM.RETAINED.PUB.QUEUE, que é usada para reter uma cópia de cada publicação retida no gerenciador de filas. Cada nome de tópico integral poderia ter até uma publicação retida armazenada nesta fila. Se os seus aplicativos usarem as publicações retidas em vários tópicos diferentes ou se as suas mensagens de publicação retidas forem mensagens grandes, os requisitos para o armazenamento dessa fila devem ser planejados cuidadosamente, incluindo sua designação para seu próprio conjunto de páginas se os requisitos de armazenamento para ele forem grandes. Para melhorar o desempenho, você deve definir essa fila com um tipo de índice de MSGID (conforme mostrado na definição de fila de amostra fornecida).
- Uma fila local denominada SYSTEM.DURABLE.SUBSCRIBER.QUEUE, que é usada para reter uma cópia persistente das descrições duráveis no gerenciador de filas. Para melhorar o desempenho, você deve definir essa fila com um tipo de índice de CORRELID (conforme mostrado na definição de fila de amostra fornecida).

- Uma fila local denominada SYSTEM.DURABLE.MODEL.QUEUE, que é usada como um modelo para as assinaturas gerenciadas duráveis.
- Uma fila local chamada SYSTEM.NDURABLE.MODEL.QUEUE, que é usada como um modelo para assinaturas não duráveis gerenciadas.
- Uma lista de nomes denominada SYSTEM.QPUBSUB.QUEUE.NAMELIST, que contém uma lista de nomes de fila monitorados pela interface publicar/assinar enfileirada.
- Uma lista de nomes denominada SYSTEM.QPUBSUB.SUBPOINT.NAMELIST, que contém uma lista objetos de tópico usados pela interface publicar/assinar enfileirada para corresponder os objetos de tópico com os pontos de assinatura.
- Um tópico denominado SYSTEM.BASE.TOPIC, que é usado como um tópico base para resolver os atributos.
- Um tópico denominado SYSTEM.BROKER.DEFAULT.STREAM, que é o fluxo padrão usado pela interface publicar/assinar enfileirada.
- Um tópico denominado SYSTEM.BROKER.DEFAULT.SUBPOINT, que é o ponto de assinatura RFH2 padrão usado pela interface publicar/assinar enfileirada.
- Um tópico denominado SYSTEM.BROKER.ADMIN.STREAM, que é o fluxo administrador usado pela interface publicar/assinar enfileirada.
- Uma assinatura denominada SYSTEM.DEFAULT.SUB, que é um objeto de assinatura padrão usado para fornecer valores padrão nos comandos DEFINE SUB.

Objetos Padrão do Sistema

Os objetos padrão do sistema são usados para fornecer atributos padrão quando você definir um objeto e não especificar o nome de outro objeto no qual basear a definição.

Os nomes das definições de objeto do sistema padrão começam com os caracteres "SYSTEM.DEFAULT" ou "SYSTEM.DEF." Por exemplo, a fila local padrão do sistema é denominada SYSTEM.DEFAULT.LOCAL.QUEUE.

Esses objetos definem os padrões do sistema para os atributos desses objetos do IBM MQ:

- Filas locais
- Filas Modelo
- Filas de Alias
- Filas Remotas
- Processos
- Listas de Nomes
- Canais
- Classes de Armazenamento
- Informações sobre Autenticação

As filas compartilhadas são um tipo especial de fila local; portanto, ao definir uma fila compartilhada, a definição é baseada no SYSTEM.DEFAULT.LOCAL.QUEUE. Você precisa se lembrar de fornecer um valor para o nome da estrutura Recurso de Acoplamento, porque não está especificado na definição padrão. Como alternativa, você poderia definir sua própria definição de fila compartilhada para usar como base para as filas compartilhadas para que todas herdem os atributos necessários. Lembre-se de que você precisa definir uma fila compartilhada em um gerenciador de filas apenas no grupo de filas compartilhadas.

Objetos do Comando do Sistema

Os nomes dos objetos do comando do sistema iniciam com os caracteres SYSTEM.COMMAND. Você deve definir esses objetos antes de poder usar as operações do IBM MQ e os painéis de controle para emitir comandos para um subsistema IBM MQ.

Existem dois objetos de comando do sistema:

1. A fila de entrada de comando do sistema é uma fila local na qual comandos são colocados antes de serem processados pelo processador de comandos do IBM MQ. Deve ser chamada de SYSTEM.COMMAND.INPUT. Um alias denominado SYSTEM.ADMIN.COMMAND.QUEUE também deve ser definido, para compatibilidade com IBM MQ for Multiplatformse para uso pelo IBM MQ Console e administrative REST API.
2. SYSTEM.COMMAND.REPLY.MODEL é uma fila modelo que define a fila de resposta de comando do sistema.

Há dois objetos extra para serem usados pelo IBM MQ Explorer:

- Fila SYSTEM.MQEXPLORER.REPLY.MODEL
- Canal SYSTEM.ADMIN.SVRCONN

SISTEMA SYSTEM.REST.REPLY.QUEUE é a fila de respostas usada pelo IBM MQ administrative REST API..

Normalmente, os comandos são enviados usando mensagens não persistentes de maneira que ambos os objetos de comandos do sistema devem ter o atributo DEFPSIST(NO) de modo que os aplicativos que os usam (incluindo os aplicativos fornecidos como o programa utilitário e as operações e os painéis de controle) recebam mensagens não persistentes, por padrão. Se você tiver um aplicativo que use mensagens persistentes para os comandos, configure o atributo DEFTYPE(PERMDYN) para a fila de resposta, porque as mensagens de resposta a esses comandos são persistentes.

Objetos de administração do sistema

Os nomes dos objetos de administração do sistema iniciam com os caracteres SYSTEM.ADMIN.

Existem sete objetos de administração do sistema:

- A fila SYSTEM.ADMIN.CHANNEL.EVENT
- A fila SYSTEM.ADMIN.COMMAND.EVENT
- A fila SYSTEM.ADMIN.CONFIG.EVENT
- A fila SYSTEM.ADMIN.PERFM.EVENT
- A fila SYSTEM.ADMIN.QMGR.EVENT
- A fila SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- A fila SYSTEM.ADMIN.ACTIVITY.QUEUE

Filas de canais

Para usar o enfileiramento distribuído, você precisa definir os seguintes objetos:

- Uma fila local com o nome SYSTEM.CHANNEL.SYNCQ, que é usada para manter os número de sequência e os identificadores de unidade lógica de trabalho (LUWID) de canais. Para melhorar o desempenho do canal, você deve definir essa fila com um tipo de índice de MSGID (conforme mostrado na definição de fila de amostra fornecida).
- Uma fila local com o nome SYSTEM.CHANNEL.INITQ, que é usado para os comandos do canal.

Não é possível definir essas filas como filas compartilhadas.

Filas de Clusters

Para usar clusters do IBM MQ, você precisa definir os seguintes objetos:

- Uma fila local denominada SYSTEM.CLUSTER.COMMAND.QUEUE, que é usada para comunicar as mudanças de repositório entre os gerenciadores de filas. As mensagens gravadas nesta fila contêm atualizações para os dados do repositório a serem aplicados na cópia local do repositório ou solicitações para os dados do repositório.
- Uma fila local denominada SYSTEM.CLUSTER.REPOSITORY.QUEUE, que é usada para reter uma cópia persistente do repositório.
- Uma fila local denominada SYSTEM.CLUSTER.TRANSMIT.QUEUE, que é a fila de transmissão para todos os destinos no cluster. Por motivos de desempenho, você deve definir essa fila com um tipo de índice CORRELID (conforme mostrado na definição de fila de amostra).

Essas filas geralmente contêm grandes números de mensagens.

Não é possível definir essas filas como filas compartilhadas.

Filas do grupo de filas compartilhadas

Para usar os canais compartilhados e o enfileiramento intragrupo, você precisa definir os seguintes objetos:

- Uma fila compartilhada com o nome SYSTEM.QSG.CHANNEL.SYNCQ, que é usada para reter as informações de sincronização para os canais compartilhados.
- Uma fila compartilhada com o nome SYSTEM.QSG.TRANSMIT.QUEUE, que é usado como a fila de transmissão para o enfileiramento intragrupo. Se você estiver executando em um grupo de filas compartilhadas, deverá definir essa fila, mesmo se não estiver usando o enfileiramento intragrupo.

Classes de Armazenamento

Recomenda-se que defina as seis classes de armazenamento a seguir. Você deve definir quatro delas porque é requerido pelo IBM MQ. As outras definições de classe de armazenamento são recomendadas porque são usadas nas definições de fila de amostra.

DEFAULT (necessário)

Esta classe de armazenamento é usada para todas as filas de mensagens que não são críticas para o desempenho e não se encaixam em nenhuma outra classe de armazenamento. Também é a classe de armazenamento padrão fornecida, se você não especificar uma ao definir uma fila.

NODEFINE (necessário)

Essa classe de armazenamento será usada se a classe de armazenamento especificada ao definir uma fila não for definida.

REMOTE (necessário)

Essa classe de armazenamento é usada principalmente para filas de transmissão ou seja, filas relacionadas ao sistema com mensagens de vida útil curta e de desempenho crítico.

SYSNLGLV

Essa classe de armazenamento é usada para mensagens de vida útil longa e desempenho crítico.

SYSTEM (necessário)

Essa classe de armazenamento é usada para filas de mensagens relacionadas ao sistema, críticas ao desempenho, por exemplo, o SYSTEM.CHANNEL.SYNCQ e o SYSTEM.CLUSTER.* filas.

SYSVOLAT

Essa classe de armazenamento é usada para mensagens de vida útil curta e desempenho crítico.

É possível modificar seus atributos e incluir outras definições de classe de armazenamento conforme necessário.

Definindo a Fila de Devoluções de Objeto do Sistema

A fila de devoluções será usada se o destino da mensagem não for válido. IBM MQ coloca essas mensagens em uma fila local chamada fila de devoluções. Embora não seja obrigatório ter uma fila de devoluções, ela deve ser considerada essencial, principalmente se você estiver usando enfileiramento distribuído ou uma das pontes do IBM MQ.

Não defina a fila de devoluções como uma fila compartilhada. Uma colocação em uma fila local em um gerenciador de filas pode obter a colocação na fila de mensagens não entregues. Se a fila de mensagens não entregues foi uma fila compartilhada, um manipulador de fila de mensagens não entregues em um sistema diferente poderia processar a mensagem e colocá-la em uma fila com o mesmo nome, mas como ela está em um gerenciador de filas diferente, essa seria a fila errada ou teria um perfil de segurança diferente. Se a fila não existisse, ela falharia ao reprocessá-la.

Se você decidir definir uma fila de devoluções, também deve informar ao gerenciador de filas seu nome. Para isso, use o comando `ALTER QMGR DEADQ(queue-name)`. Para obter mais informações, consulte [Exibindo e mudando atributos do gerenciador de filas](#).

Fila de transmissão padrão

A fila de transmissão padrão é usada quando nenhuma outra fila de transmissão adequada estiver disponível para enviar as mensagens para outro gerenciador de filas. Se você definir uma fila de transmissão padrão, também deve definir um canal para servir a fila. Se não fizer isso, as mensagens que forem colocadas na fila de transmissão padrão não serão transmitidas ao gerenciador de filas remotas e permanecerão na fila.

Se você decidir definir uma fila de transmissão padrão, também deve informar ao gerenciador de filas seu nome. Para isso, use o comando `ALTER QMGR`.

Filas internas

• Fila de dados pendentes

- Uma fila definida para uso interno, `SYSTEM.PENDING.DATA.QUEUE`, suporta o uso de assinaturas duráveis em um ambiente de publicação / assinatura do JMS

• JMS 2.0 fila de preparação de atraso de entrega

- Se a funcionalidade de atraso de entrega fornecida pelo JMS 2.0 for usada, então uma fila de migração interna, `SYSTEM.DDELAY.LOCAL.QUEUE`, deve ser definido. Esta fila é usada pelo gerenciador de filas para armazenar temporariamente mensagens enviadas com um atraso de entrega diferente de zero, até que o atraso de entrega seja concluído, e a mensagem colocada em seu destino. Uma definição de amostra para esta fila é fornecida e comentada em `CSQ4INSG`.
- Ao definir a fila `SYSTEM.DDELAY.LOCAL.QUEUE`, deve-se configurar os atributos `STGCLASS`, `MAXMSGL` e `MAXDEPTH` para o número antecipado de mensagens que serão enviadas com um atraso de entrega. Além disso, ao definir a fila `SYSTEM.DDELAY.LOCAL.QUEUE`, certifique-se de que somente o gerenciador de filas possa colocar mensagens nesta fila. É necessário tomar cuidado para assegurar que nenhum identificador de usuários tenha a autoridade para colocar mensagens nesta fila.

Fila de Autenticação de Canal

Para uso interno de autenticação de canal, a fila `SYSTEM.CHLAUTH.DATA.QUEUE` é necessária. Definições de amostra são fornecidas com o IBM MQ para ajudá-lo a definir esses objetos. Essa amostra é descrita em `CSQ4INSA`, que também define algumas regras padrão.

Ajustando o seu gerenciador de filas no IBM MQ for z/OS

Existem algumas etapas simples que podem ser executadas para assegurar que o gerenciador de filas esteja sintonizado para evitar problemas no desempenho.

Existem inúmeras maneiras nas quais é possível melhorar o desempenho do seu gerenciador de filas, que são controladas pelos atributos do gerenciador de filas configurados pelo comando ALTER QMGR. Essa seção contém informações sobre como é possível fazer isso configurando o número máximo de mensagens permitidas no gerenciador de filas ou executando a 'manutenção' no gerenciador de filas. [IBM MQ SupportPac MP16 - IBM MQ para z/OS Planejamento de capacidade & ajustando](#) fornece mais informações sobre desempenho e ajuste.

Pontos de Sincronização

Uma das funções do gerenciador de filas é o controle de ponto de sincronização em um aplicativo. Um aplicativo constrói uma unidade de trabalho que contém qualquer número de chamadas MQPUT ou MQGET finalizadas com uma chamada MQCMIT.

À medida que o número de chamadas de MQPUT ou de MQGET dentro do escopo de um MQCMIT aumenta, o custo de desempenho da confirmação aumenta significativamente. Em geral, os aplicativos devem ser projetados para não emitir MQPUT/MQGET de um número grande de mensagens em um ponto de sincronização único.

É possível limitar administrativamente o número de mensagens dentro de qualquer ponto de sincronização único usando o atributo do gerenciador de filas MAXUMSGS. Se um aplicativo exceder esse limite, ele receberá MQRC_SYNCPOINT_LIMIT_REACHED no MQPUT, MQPUT1 ou a chamada de MQGET que excede o limite. O aplicativo deve, então, emitir o MQCMIT ou MQBACK conforme apropriado.

O valor padrão do MAXUMSGS é 10000. Esse valor poderá ser diminuído se você desejar impor um limite inferior, o que também pode ajudar a proteger contra loop de aplicativos. Antes de reduzir o MAXUMSGS, certifique-se de que você entenda seus aplicativos existentes para assegurar que eles não excedam o limite ou que possam tolerar o código de retorno MQRC_SYNCPOINT_LIMIT_REACHED

Mensagens Expiradas

As mensagens que expiraram são descartadas pela próxima chamada MQGET apropriada. No entanto, se essa chamada não ocorrer, as mensagens expiradas não serão descartadas e, para algumas filas, especialmente aquelas em que a recuperação da mensagem seja feita por MessageId, CorrelId ou GroupId e a fila seja indexada para desempenho, várias mensagens expiradas podem se acumular. O gerenciador de filas pode varrer periodicamente qualquer fila para as mensagens expiradas, que são excluídas. É possível escolher a frequência na qual essa varredura ocorre, se houver. Existem duas maneiras de fazer isto:

Solicitação Explícita

É possível controlar quais filas são varridas e quando. Emita o comando REFRESH QMGR TYPE(EXPIRY), especificando a fila ou as filas que você deseja que sejam varridas.

Varredura Periódica

É possível especificar um intervalo de expiração no objeto do gerenciador de filas usando o atributo EXPRYINT. O gerenciador de filas mantém informações sobre as mensagens expiradas em cada fila e sabe em que momento uma varredura para as mensagens expiradas é válida. Sempre que o intervalo EXPRYINT é atingido, o gerenciador de filas procura pelas filas de candidatos que precisam ser varridas para mensagens expiradas e varrem apenas as filas que são consideradas válidas. Ele não varre todas as filas. Isso evita que qualquer tempo do processador seja gasto com varreduras desnecessárias.

As filas compartilhadas são varridas apenas por um gerenciador de filas no grupo de filas compartilhadas. Geralmente, o primeiro gerenciador de filas a ser reiniciado ou o primeiro a ter o EXPRYINT configurado executa a varredura.

Nota: Deve-se configurar o mesmo valor EXPRYINT para todos os gerenciadores de filas dentro de um grupo de filas compartilhadas.

Definições de amostra fornecidas com o IBM MQ for z/OS

Use este tópico como uma referência para o JCL de amostra e o código fornecido com o IBM MQ for z/OS.

As seguintes definições de amostra são fornecidas com o IBM MQ na biblioteca thlqual.SCSQPROC. É possível usá-las para definir os objetos do sistema e para customizar seus próprios objetos. É possível incluir alguns deles nos conjuntos de dados de entrada de inicialização (descritos em [Comandos de Inicialização](#)).

Conjunto de dados de entrada de inicialização	Nome da amostra
CSQINP1	CSQ4INP1 CSQ4INPR
CSQINP2	CSQ4INSA CSQ4INYS ¹ CSQ4INSX CSQ4INSG CSQ4INSR CSQ4INSS CSQ4INSJ CSQ4INSM CSQ4INYG CSQ4INYR CSQ4INYC CSQ4INYD CSQ4INSC
CSQINPT	CSQ4INST CSQ4INYT
Outro	CSQ4DISP CSQ4INPX CSQ4IVPQ CSQ4IVPG CSQ4MSTR CSQ4MSRR CSQ4QMIN

Nota:

1. A ordem dessas definições de amostra é importante: ocorrerá um erro, se INYS, INSX e INSG forem ordenadas incorretamente.

Amostras de CSQINP1

Use o conjunto de dados CSQINP1 de amostra thlqual.SCSQPROC(CSQ4INP1) quando estiver usando um conjunto de páginas para cada classe de mensagem ou thlqual.SCSQPROC(CSQ4INPR) quando estiver usando diversos conjuntos de página para classes maiores de mensagem. Ele contém definições de buffer pools, associações de conjunto de páginas para buffer pool e um comando ALTER SECURITY. Inclua a amostra na concatenação CSQINP1 de seu procedimento de tarefa iniciada pelo gerenciador de filas.

Amostras CSQINP2

Amostra de Objetos do Sistema CSQ4INSG

O conjunto de dados CSQINP2 de amostra thlqual.SCSQPROC(CSQ4INSG) contém as definições para os seguintes objetos do sistema para uso geral:

- Objetos Padrão do Sistema
- Objetos do Comando do Sistema
- Objetos de administração do sistema
- Outros objetos para uso do sistema

Você deve definir os objetos nessa amostra, mas precisa fazer isso apenas uma vez quando o subsistema for iniciado pela primeira vez. Incluir definições no conjunto de dados CSQINP2 é a melhor maneira de fazer isso. Elas são mantidas no encerramento e na reinicialização do gerenciador de filas. Você não deve alterar os nomes do objeto, mas pode alterar seus atributos, se necessário.

Quando as seguintes condições forem atendidas, será colocada uma mensagem na fila SYSTEM.DURABLE.SUBSCRIBER.QUEUE (mesmo que a assinatura da publicação não esteja ativa):

- O atributo PSMODE do QMGR é configurado como DISABLED
- A instrução CSQ4INST do objeto de amostra DEFINE SUB('SYSTEM.DEFAULT.SUB') está presente

Para evitar isto, exclua ou comente a linha da instrução DEFINE SUB('SYSTEM.DEFAULT.SUB').

A fila temporária de atraso de entrega do JMS 2.0 , SYSTEM.DDELAY.LOCAL.QUEUE só precisa ser definido se o atraso de entrega do JMS 2.0 for usado Por padrão, a definição de fila é comentada, e é possível remover comentário, se necessário.

Objeto do sistema CSQ4INSA e amostra de autenticação

O conjunto de dados CSQINP2 de amostra thlqual.SCSQPROC(CSQ4INSA) contém a definição de fila do sistema de autenticação de canal. Essa fila retém os registros de autenticação de canal. Ela também contém as regras de autenticação de canal padrão.

Você deve definir os objetos nessa amostra se CHLAUTH for ENABLED no gerenciador de filas e você quiser executar canais ou quiser efetuar SET ou DISPLAY do registro CHLAUTH. Só é necessário defini-los uma vez quando o subsistema for iniciado pela primeira vez. Incluir definições no conjunto de dados CSQINP2 é a melhor maneira de fazer isso. Elas são mantidas durante o encerramento e a reinicialização do gerenciador de filas, e você não deve alterar o nome da fila.

Amostra de Objeto do Sistema CSQ4INSS

Será possível definir objetos adicionais do sistema se você estiver usando grupos de filas compartilhadas.

O conjunto de dados de amostra thlqual.SCSQPROC(CSQ4INSS) contém comandos de amostra para uso com estruturas CF e um conjunto de definições para os objetos de sistema necessários para canais compartilhados e enfileiramento intragrupo.

Não é possível usar essa amostra no estado em que se encontra; você deve customizá-la antes de usar. Em seguida, é possível incluir este membro na concatenação CSQINP2 DD do procedimento de inicialização do gerenciador de filas ou é possível usá-lo como entrada para a função COMMAND do utilitário CSQUTIL para emitir os comandos necessários.

Quando você está definindo o grupo ou objetos compartilhados, é necessário incluí-los na concatenação CSQINP2 DD para apenas um gerenciador de filas no grupo de filas compartilhadas.

Amostra do Objeto do Sistema CSQ4INSX

Você deve definir os objetos adicionais do sistema se estiver usando o armazenamento em cluster e enfileiramento distribuído.

O conjunto de dados de amostra thlqual.SCSQPROC(CSQ4INSX) contém as definições de fila necessários. É possível incluir este membro na concatenação CSQINP2 DD do procedimento de inicialização do gerenciador de filas ou usá-lo como entrada para a função COMMAND no utilitário CSQUTIL para emitir os comandos DEFINE necessários.

Existem dois tipos de definições de objeto:

- SYSTEM.CHANNEL.xx, necessário para qualquer enfileiramento distribuído
- SYSTEM.CLUSTER.xx, necessário para armazenamento em cluster

Amostra do objeto JMS do sistema CSQ4INSJ

Define as filas usadas no domínio de publicação/assinatura do JMS.

Amostra de objeto do sistema CSQ4INSM

Se estiver usando o advanced message security, deve-se definir objetos do sistema adicionais. O conjunto de dados de amostra thlqual.SCSQPROC(CSQ4INSM) contém as definições de fila necessárias.

Amostra do Objeto CSQ4INSR

Define as filas usadas pelo WebSphere Application Server e brokers.

Amostra do Objeto CSQ4INYD

Se você estiver usando o enfileiramento distribuído e precisar configurar suas próprias filas, processos e canais.

O conjunto de dados de amostra thlqual.SCSQPROC(CSQ4INYD) contém definições de amostra que você usa para customizar seus objetos de enfileiramento distribuído. Ele consiste em:

- Um conjunto de definições para a extremidade de envio
- Um conjunto de definições para a extremidade de recebimento
- Um conjunto de definições para usar os clientes

Não é possível usar essa amostra no estado em que se encontra, você deve customizá-la antes de usar. Em seguida, é possível incluir este membro na concatenação CSQINP2 DD do procedimento de inicialização do gerenciador de filas ou é possível usá-lo como entrada para a função COMMAND do utilitário CSQUTIL para emitir os comandos DEFINE necessários. (Isso é preferível porque significa que você não precisa redefinir esses objetos sempre que reiniciar o gerenciador de filas).

Amostra do Objeto CSQ4INYC

Se você estiver usando o armazenamento em cluster, as definições equivalentes às definições de canal e definições de fila remota de enfileiramento distribuído serão criadas automaticamente, quando necessário. No entanto, algumas definições de canal manual são necessárias - um canal do receptor de clusters para o cluster e uma definição do emissor de cluster para, pelo menos, um gerenciador de filas do repositório do cluster.

O conjunto de dados de amostra: thlqual.SCSQPROC(CSQ4INYC) contém as seguintes definições de amostra que você pode usar para customizar os objetos de armazenamento em cluster:

- Definições para o gerenciador de filas
- Definições para o canal de recebimento
- Definições para o canal de envio
- Definições para as filas de clusters
- Definições para listas de clusters

Não é possível usar essa amostra no estado em que se encontra, você deve customizá-la antes de usar. Em seguida, é possível incluir este membro na concatenação CSQINP2 DD do procedimento de

inicialização do gerenciador de filas ou é possível usá-lo como entrada para a função COMMAND do utilitário CSQUTIL para emitir os comandos DEFINE necessários. Isso é preferível porque significa que você não precisa redefinir esses objetos a cada vez que reiniciar o IBM MQ.

Amostra de Objeto CSQ4INYG

O conjunto de dados de amostra thlqual.SCSQPROC(CSQ4INYG) contém as seguintes definições de amostra que podem ser usadas para customizar seus próprios objetos para uso geral:

- Fila de devoluções
- Fila de transmissão padrão
- Objetos do adaptador CICS

Não é possível usar essa amostra no estado em que se encontra, você deve customizá-la antes de usar. Em seguida, é possível incluir este membro na concatenação CSQINP2 DD do procedimento de inicialização do gerenciador de filas ou é possível usá-lo como entrada para a função COMMAND do utilitário CSQUTIL para emitir os comandos DEFINE necessários. Isso é preferível porque significa que você não precisa redefinir esses objetos a cada vez que reiniciar o IBM MQ.

Além das definições de amostra aqui, é possível usar as definições de objeto do sistema como base para as suas próprias definições de recurso. Por exemplo, é possível fazer uma cópia de trabalho do SYSTEM.DEFAULT.LOCAL.QUEUE e nomeá-la MY.DEFAULT.LOCAL.QUEUE. Em seguida, alterar quaisquer dos parâmetros nesta cópia conforme necessário. Em seguida, emita um comando DEFINE por qualquer método que escolher, contanto que tenha autoridade para criar os recursos desse tipo.

Fila de transmissão padrão

Leia a descrição da Fila de transmissão padrão antes de decidir se deseja definir uma fila de transmissão padrão.

- Se decidir que deseja definir uma fila de transmissão padrão, lembre-se de que também deve definir um canal para atendê-la.
- Se decidir que não deseja definir uma, lembre-se de remover a instrução DEFXMITQ do comando ALTER QMGR na amostra.

Objetos do adaptador CICS

A amostra define uma fila de inicialização denominada CICS01.INITQ. Esta fila é usada pela transação CKTI fornecida pelo IBM MQ. É possível alterar o nome dessa fila; no entanto, ele deve corresponder ao nome especificado na tabela de inicialização do sistema (SIT) ou SYSIN do CICS na instrução INITPARM.

Amostras do objeto CSQ4INYS/CSQ4INYR

Definições de classe de armazenamento para uso:

- um conjunto de páginas para cada classe de mensagem
- diversos conjuntos de páginas para classes maiores de mensagem

Por exemplo, SYSTEM.COMMAND.INPUT usa STGCLASS ('SYSVOLAT') e SYSTEM.CLUSTER.TRANSMIT.QUEUE usa STGCLASS ('REMOTE'). Em CSQ4INYS, ambas as classes de armazenamento usam o mesmo conjunto de páginas. Em CSQ4INYR, essas classes de armazenamento usam diferentes conjuntos de páginas para diminuir o impacto do preenchimento de fila de transmissão.

Amostras de CSQINPT

CSQ4INST

O conjunto de dados de amostra: thlqual.SCSQPROC(CSQ4INST) contém a definição para a assinatura padrão do sistema.

Você deve definir o objeto nesta amostra, mas é necessário fazer isso apenas uma vez quando o mecanismo de publicação/assinatura é iniciado pela primeira vez. Incluir a definição no conjunto de dados CSQINPT é a melhor maneira de fazer isto. Ela é mantida no encerramento e na reinicialização do gerenciador de filas. Você não deve alterar o nome do objeto, mas é possível alterar seus atributos, se necessário.

CSQ4INYT

O conjunto de dados amostra: thlqual.SCSQPROC(CSQ4INYT) contém um conjunto de comandos que talvez você deseje executar quando o mecanismo de publicação/assinatura é iniciado. Essa amostra exibe informações sobre o Tópico e Assinatura.

Outro

Amostra de exibição CSQ4DISP

O conjunto de dados de amostra de amostra: thlqual.SCSQPROC(CSQ4DISP) contém um conjunto de comandos DISPLAY genéricos que exibem todos os recursos definidos em seu gerenciador de filas. Isso inclui as definições para todos os objetos do IBM MQ e definições como classes de armazenamento e rastreamento. Esses comandos podem gerar uma grande quantidade de saída. É possível usar essa amostra no conjunto de dados CSQINP2 ou como entrada para a função COMMAND do utilitário CSQUTIL.

Amostra CSQ4INPX

O conjunto de dados de amostra thlqual.SCSQPROC(CSQ4INPX) contém um conjunto de comandos que você talvez queira executar sempre que o inicializador de canais for iniciado. Você deve customizar essa amostra antes do uso; em seguida, pode incluí-la no conjunto de dados CSQINPX para o inicializador de canais.

Amostras CSQ4IVPQ e CSQ4IVPG

Os conjuntos de dados de amostra thlqual.SCSQPROC(CSQ4IVPQ) e thlqual.SCSQPROC(CSQ4IVPG) contêm conjuntos de comandos DEFINE que são necessários para executar os programas de verificação de instalação (IVPs).

É possível incluir essas amostras no conjunto de dados CSQINP2. Quando você tiver executado os IVPs com êxito, não precisará executá-los novamente sempre que o gerenciador de filas for reiniciado. Portanto, você não precisa manter essas amostras na concatenação CSQINP2 permanentemente.

Amostras CSQ4MSTR e CSQ4MSRR

Esses são procedimentos de tarefa iniciada de amostra para o gerenciador de filas: thlqual.SCSQPROC (CSQ4MSTR) e thlqual.SCSQPROC (CSQ4MSRR).

CSQ4MSRR usa CSQ4INYR na concatenação CSQINP2 para que filas importantes sejam difundidas em diferentes conjuntos de páginas.

Será possível remover os comentários para usar o cartão CSQMINI para gerenciadores de filas recém-criados, se necessário.

Amostra CSQ4QMIN

Um conjunto de dados QMINI de amostra, thlqual.SCSQPROC (CSQ4QMIN).

Consulte [Conjunto de dados QMINI](#) para obter detalhes do conjunto de dados QMINI e da sub-rotina **TransportSecurity**.

Use os links neste tópico para saber sobre os recursos do IBM MQ for z/OS para reinicialização e recuperação.

IBM MQ for z/OS possui recursos robustos para reinicialização e recuperação. Para obter informações sobre como um gerenciador de filas se recupera depois de ter sido interrompido e o que acontece quando ele é reiniciado, veja os subtópicos a seguir:

- [“Como as mudanças são feitas nos dados no IBM MQ for z/OS” na página 262](#)
- [“Como a consistência é mantida no IBM MQ for z/OS” na página 264](#)
- [“O que acontece durante a finalização em IBM MQ for z/OS” na página 266](#)
- [“O que acontece durante a reinicialização e a recuperação no IBM MQ for z/OS” na página 268](#)
- [“Como as Unidades de Recuperação em Dúvida São Resolvidas” na página 270](#)
- [“Recuperação de Fila Compartilhada” na página 272](#)

Conceitos relacionados

 [Ações de recuperação de do IBM MQ for z/OS](#)

Tarefas relacionadas

[Planejamento para backup e recuperação](#)

 [Administrando z/OS](#)

Referências relacionadas

 [Mensagens do IBM MQ for z/OS](#)

Como as mudanças são feitas nos dados no IBM MQ for z/OS

IBM MQ for z/OS deve interagir com outros subsistemas para manter todos os dados consistentes. Este tópico contém informações sobre as *unidades de recuperação*, o que eles são e como são usados nas *restaurações*.

Unidades de recuperação

Uma *unidade de recuperação* é o processamento feito por um único gerenciador de filas para o programa de aplicativo, que altera os dados do IBM MQ de um ponto de consistência a outro. Um *ponto de consistência* - também denominado *ponto de sincronização* ou *ponto de confirmação* - é um ponto no tempo em que todos os dados recuperáveis que um programa de aplicativo acessa são consistentes.

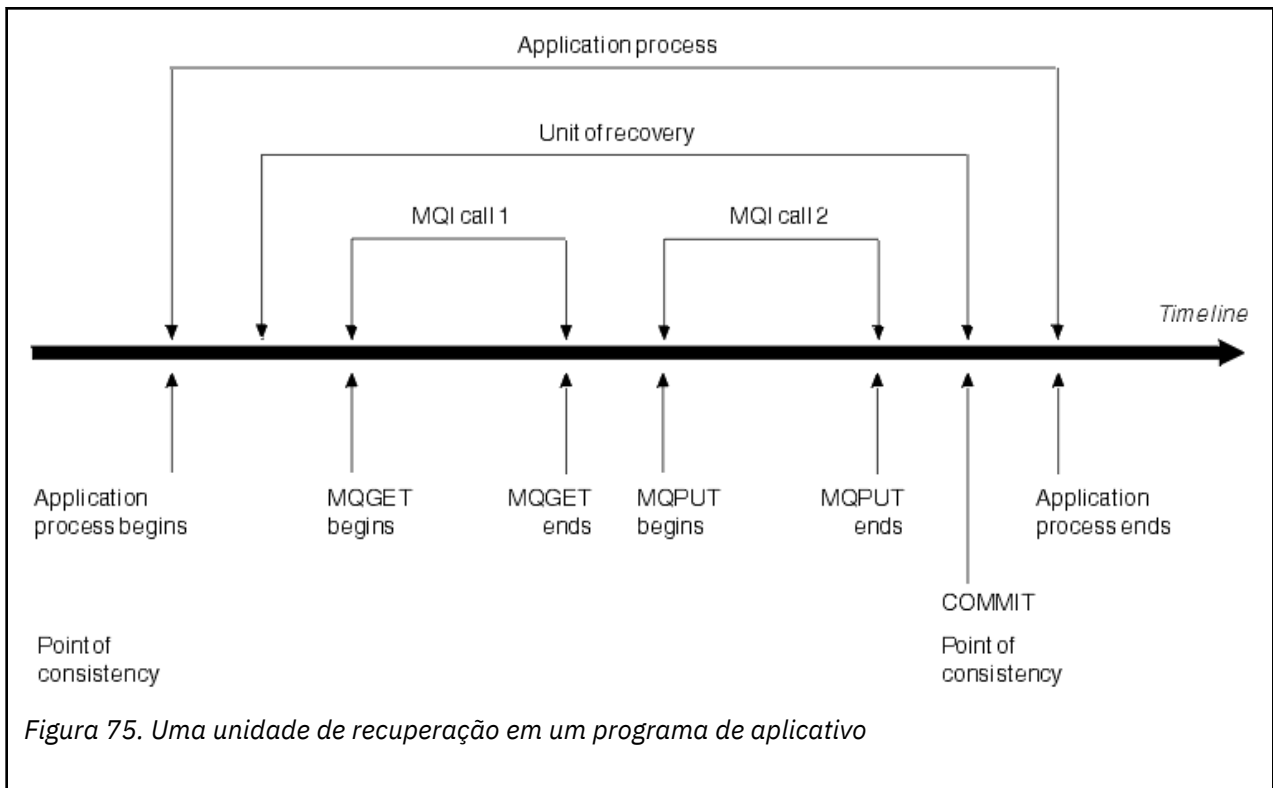


Figura 75. Uma unidade de recuperação em um programa de aplicativo

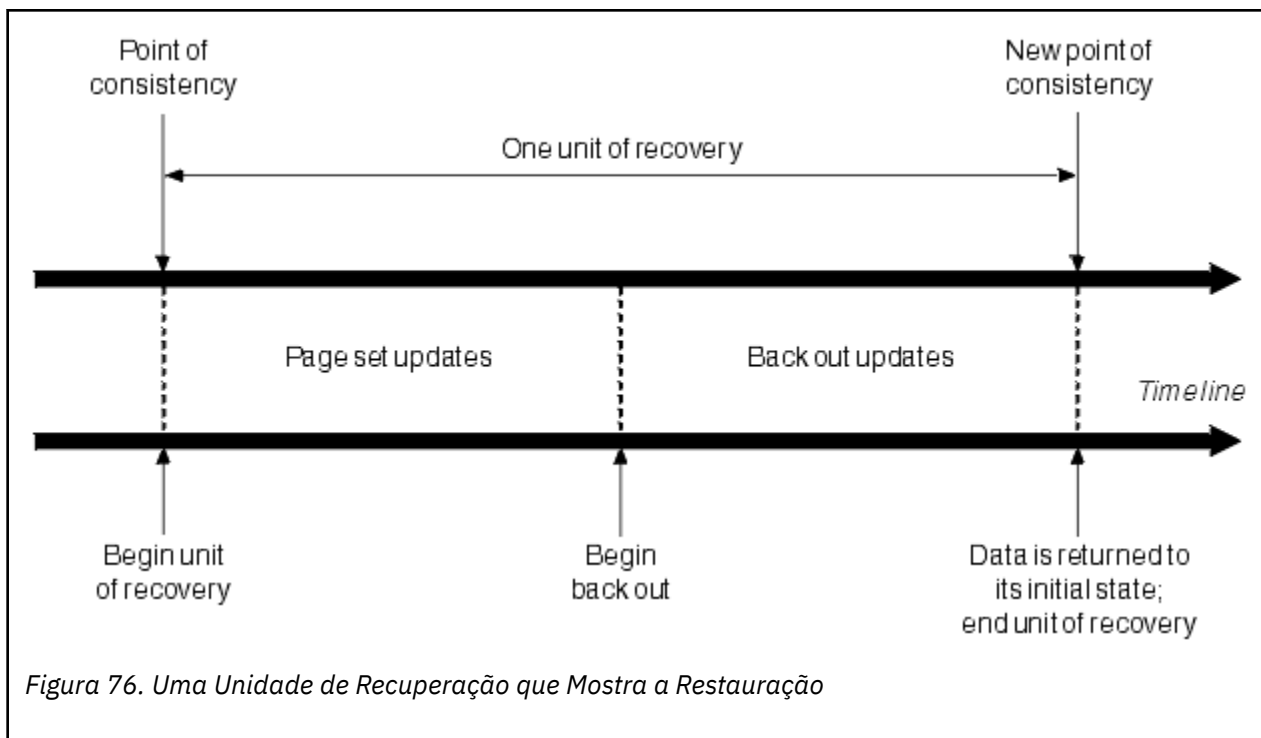
Uma unidade de recuperação é iniciada na primeira mudança nos dados depois do início do programa ou seguindo o ponto de consistência anterior; ela termina com um ponto de consistência posterior. [Figura 75 na página 263](#) mostra o relacionamento entre as unidades de recuperação, o ponto de consistência e um programa de aplicativo. Neste exemplo, o programa de aplicativo faz mudanças nas filas por meio das chamadas 1 e 2 de MQI. O programa de aplicativo pode incluir mais de uma unidade de recuperação ou apenas uma. No entanto, qualquer unidade de recuperação completa termina em um ponto de confirmação.

Por exemplo, uma transação bancária transfere fundos de uma conta a outra. Primeiro, o programa subtrai a quantia da primeira conta, a conta A. Em seguida, ele inclui a quantia na segunda conta, B. Após subtrair a quantia de A, as duas contas se tornam inconsistentes e o IBM MQ não pode executar a confirmação. Elas se tornam consistentes quando a quantia é incluída na conta B. Quando ambas as etapas estiverem concluídas, o programa poderá anunciar um ponto de consistência por meio de uma confirmação, tornando as mudanças visíveis para outros programas de aplicativos.

A terminação normal de um programa de aplicativo causa automaticamente um ponto de consistência. Algumas solicitações de programa no CICS e programas do IMS também causam um ponto de consistência, por exemplo, EXEC CICS SYNCPOINT.

Restaurando o Trabalho

Se ocorrer um erro em uma unidade de recuperação, o IBM MQ removerá quaisquer mudanças nos dados, retornando os dados ao seu estado no início da unidade de recuperação; ou seja, o IBM MQ restaura o trabalho. Os eventos são mostrados em [Figura 76 na página 264](#).



z/OS Como a consistência é mantida no IBM MQ for z/OS

Os dados no IBM MQ for z/OS devem estar consistentes com o lote, CICS, IMS ou TSO. Quaisquer dados mudados em um devem ser correspondidos por uma mudança em outro.

Antes que um sistema confirme os dados mudados, ele deve saber se outro sistema pode fazer a mudança correspondente. Portanto, os sistemas devem se comunicar.

Durante um *two-phase commit* (por exemplo, em CICS), um subsistema coordena o processo. Esse subsistema é denominado *coordenador*; o outro é o *participante*. CICS ou IMS é sempre o coordenador em interações com IBM MQ e IBM MQ é sempre o participante. No ambiente TSO ou lote, o IBM MQ pode participar em protocolos two-phase commit coordenados por z/OS RRS.

Durante um *single-phase commit* (por exemplo, em TSO ou lote), o IBM MQ é sempre o coordenador nas interações e controla completamente o processo de confirmação.

Em um ambiente do WebSphere Application Server, as semânticas do objeto de sessão de JMS determinam se uma coordenação single-phase ou two-phase commit é usada.

Consistência com CICS ou IMS

A conexão entre IBM MQ e CICS ou IMS suporta os seguintes protocolos de sincronização :

- Two-phase commit - para transações que atualizam recursos de propriedade de mais de um gerenciador de recursos.

Esse é o protocolo do ponto de sincronização distribuído padrão. Envolve mais fluxos de mensagens e criação de log do que um single-phase commit.

- Single-phase commit – para transações que atualizam recursos de propriedade de um único gerenciador de recursos (IBM MQ).

Este protocolo é otimizado para fluxos de mensagens e criação de log.

- Bypass do ponto de sincronização – para transações que envolvem o IBM MQ, mas que não fazem nada no gerenciador de filas que requer um ponto de sincronização (por exemplo, pesquisa em uma fila).

Em cada caso, o CICS ou IMS atua como o gerenciador de ponto de sincronização.

Os estágios do two-phase commit que o IBM MQ usa para se comunicar com o CICS ou IMS são os seguintes:

1. Na fase 1, cada sistema determina independentemente se registrou informações de recuperação suficientes em seu log e pode confirmar seu trabalho.

No fim da fase, os sistemas se comunicam. Se eles concordarem, cada um inicia a próxima fase.

2. Nas fase 2, as mudanças se tornam permanentes. Se um dos sistemas for encerrado de maneira anormal durante a fase 2, a operação será concluída pelo processo de recuperação durante a reinicialização.

Ilustração do processo two-phase commit

Figura 77 na página 265 ilustra o processo two-phase commit. Eventos no coordenador de CICS ou IMS são mostrados na linha superior, os eventos no IBM MQ na linha inferior.

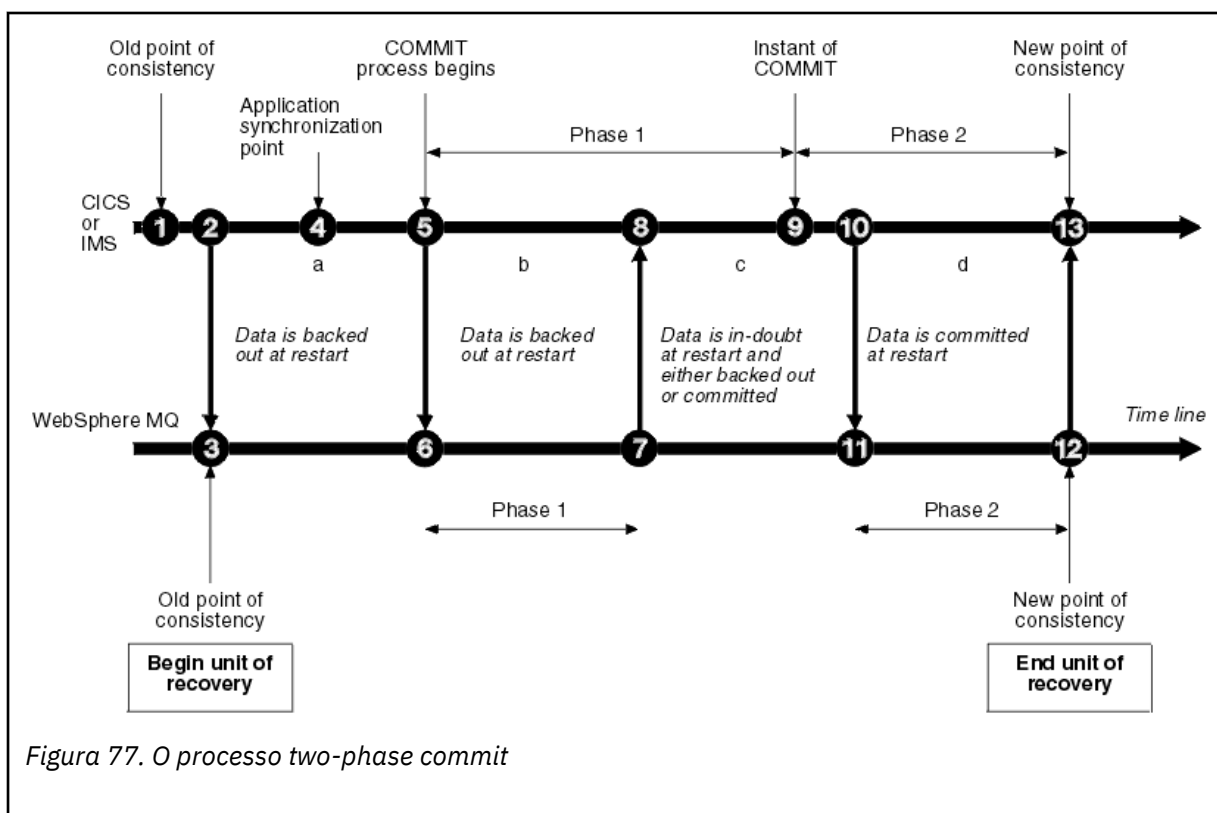


Figura 77. O processo two-phase commit

Os números na seguinte seção estão vinculados aos mostrados na figura.

1. Os dados no coordenador estão em um ponto de consistência.
2. Um programa de aplicativo no coordenador chama IBM MQ para atualizar uma fila incluindo uma mensagem.
3. Isso inicia uma unidade de recuperação no IBM MQ.
4. O processamento continua no coordenador até que um ponto de sincronização de aplicativo seja atingido.
5. O coordenador então inicia o processamento de confirmação. Os programas CICS usam um comando SYNCPOINT ou uma terminação de aplicativo normal para iniciar a confirmação. Os programas IMS podem iniciar a confirmação usando uma chamada CHKP, uma chamada SYNC, uma chamada GET UNIQUE para IOPCB ou uma terminação de aplicativo normal. A fase 1 do processamento de confirmação é iniciada.
6. Conforme o coordenador inicia o processamento da fase 1, o IBM MQ também o faz.
7. IBM MQ conclui com êxito a fase 1, grava esse fato em seu log e notifica o coordenador.

8. O coordenador recebe a notificação.
9. O coordenador conclui com êxito seu processamento de fase 1. Agora ambos os subsistemas concordam em confirmar as mudanças de dados, porque ambos concluíram a fase 1 e poderiam recuperar-se de quaisquer erros. O coordenador registra em seu log o instante da confirmação - a decisão irrevogável dos dois subsistemas de fazer as mudanças.
O coordenador agora inicia a fase 2 do processamento - a confirmação real.
10. O coordenador notifica o IBM MQ para iniciar sua fase 2.
11. IBM MQ registra o início da fase 2.
12. A fase 2 é concluída com êxito e agora esse é um novo ponto de consistência para o IBM MQ. IBM MQ então notifica o coordenador que concluiu seu processamento da fase 2.
13. O coordenador conclui seu processamento da fase 2. Os dados controlados por ambos os subsistemas agora estão consistentes e disponíveis para outros aplicativos.

Como a Consistência é Mantida após uma Finalização Anormal

Quando um gerenciador de filas é reiniciado após uma finalização anormal, ele deve determinar se deve confirmar ou restaurar as unidades de recuperação que estavam ativas no momento da finalização. Para algumas unidades de recuperação, o IBM MQ tem informações suficientes para tomar a decisão. Para outras, não tem, e deve obter as informações do coordenador quando a conexão for restabelecida.

O Figura 77 na página 265 mostra quatro períodos dentro das duas fases: a, b, c e d. O status de uma unidade de recuperação depende do período em que a finalização aconteceu. O status pode ser um dos seguintes:

Em Andamento

O gerenciador de filas terminou antes de concluir a fase 1 (período a ou b); durante a reinicialização, o IBM MQ restaura as atualizações.

Não assegurado

O gerenciador de filas terminou após concluir a fase 1 e antes de iniciar a fase 2 (período c); apenas o coordenador sabe se o erro aconteceu antes ou após a confirmação (ponto 9). Se isso aconteceu antes, o IBM MQ deve restaurar suas mudanças; se isso aconteceu depois, o IBM MQ deve fazer suas mudanças e confirmá-las. Na reinicialização, o IBM MQ espera pelas informações do coordenador antes de processar essa unidade de recuperação.

Em Confirmação

O gerenciador de filas terminou após ter iniciado seu próprio processamento da fase 2 (período d); e faz mudanças confirmadas.

Em Restauração

O gerenciador de filas terminou após uma unidade de recuperação ter iniciado backout mas antes que o processo estivesse concluído (não mostrado na figura) durante o reinício, o IBM MQ continua a restaurar as mudanças.

O que acontece durante a finalização em IBM MQ for z/OS

Um gerenciador de filas finaliza normalmente em resposta ao comando STOP QMGR. Se um gerenciador de filas parar por qualquer outra razão, a finalização será anormal.

Observe que, durante a finalização do gerenciador de filas, o IBM MQ internamente emite o comando

```
DISPLAY CONN(*) TYPE(CONN) ALL WHERE (APPLTYPE NE SYSTEMAL)
```

para que você fique ciente de quais encadeamentos podem impedir a conclusão do encerramento do gerenciador de filas.

SYSTEMAL corresponde a APPLTYPES de SYSTEM ou CHINIT, portanto, os tipos de aplicativos de filtragem do comando DISPLAY CONN não correspondentes a SYSTEMAL retornam as informações do log da tarefa sobre encadeamentos que poderiam impedir o encerramento normal.

Finalização normal

Em uma finalização normal, o IBM MQ para toda a atividade de maneira ordenada. É possível parar o IBM MQ usando o modo quiesce, de força ou modo de reinicialização. Os efeitos são fornecidos em Tabela 23 na página 267.

<i>Tabela 23. Finalização usando QUIESCE, FORCE e RESTART</i>			
Tipo de Encadeamento	QUIESCE	FORCE	RESTART
Encadeamentos ativos	Executar na conclusão	Restaurar	Restaurar
Novos encadeamentos	Pode iniciar	Não permitido	Não permitido
Novas conexões	Não permitido	Não permitido	Não permitido

Os aplicativos em lote serão notificados se ocorrer uma finalização enquanto o aplicativo ainda estiver conectado.

Com o CICS, um encadeamento atual executa apenas no fim da unidade de recuperação. Com o CICS, parar um gerenciador de filas no modo quiesce para o adaptador CICS e, se uma tarefa ativa contiver mais de uma unidade de recuperação, a tarefa não será executada necessariamente até a conclusão.

Se você parar um gerenciador de filas no modo de força ou reinicialização, nenhum novo encadeamento será alocado e o trabalho nos encadeamentos conectados será restaurado. Usar esses modos podem criar unidades de recuperação em dúvida que estão entre as fases de processamento de confirmação. Eles são resolvidos quando o IBM MQ é reconectado com o controle do CICS, IMS, ou do subsistema RRS.

Ao parar um gerenciador de filas, em qualquer modo, as etapas são:

1. As conexões são encerradas.
2. O IBM MQ para de aceitar os comandos.
3. IBM MQ assegura que qualquer atualização pendente nos conjuntos de páginas seja concluída.
4. O comando DISPLAY USAGE é emitido internamente pelo IBM MQ de modo que o RBA de reinicialização seja registrado no log do console do z/OS.
5. O ponto de verificação de encerramento é obtido e o BSDS é atualizado.

A finalização que especifica o modo quiesce não afeta as unidades de recuperação em dúvida. Qualquer unidade que estiver em dúvida permanecerá em dúvida.

Finalização anormal

Uma finalização anormal pode deixar os dados em um estado inconsistente, por exemplo:

- uma unidade de recuperação foi interrompida antes de atingir um ponto de consistência.
- Os dados confirmados não foram gravados nos conjuntos de páginas.
- Os dados não confirmados foram gravados nos conjuntos de páginas.
- um programa de aplicativo foi interrompido entre a fase 1 e a fase 2 do processo de confirmação, deixando a unidade de recuperação em dúvida.

O IBM MQ resolve quaisquer inconsistências de dados provenientes da finalização anormal durante a reinicialização e a recuperação.

O que acontece durante a reinicialização e a recuperação no IBM MQ for z/OS

IBM MQ usa seu log de recuperação e os conjuntos de dados de autoinicialização (BSDS) para determinar o que deve ser recuperado quando for reiniciado. O BSDS identifica os conjuntos de dados do log de archive e ativos, e o local do ponto de verificação mais recente do IBM MQ no log.

Introdução à Reinicialização e Recuperação

Após o IBM MQ ter sido inicializado, ocorrerá o processo de reinicialização do gerenciador de filas da seguinte forma:

- Inicialização do log
- Reconstrução de status atual
- Recuperação de log de encaminhamento
- Recuperação regressiva de log
- Reconstrução do índice de fila

Quando a recuperação não tiver sido concluída:

- As mudanças confirmadas são refletidas nos dados.
- A atividade em dúvida é refletida nos dados. Entretanto, os dados são bloqueados e não podem ser usados até que o IBM MQ reconheça e aja a respeito da decisão em dúvida.
- As mudanças interrompidas em andamento e em interrupção foram removidas das filas. As mensagens são consistentes e podem ser usadas.
- Um novo ponto de verificação foi obtido.
- Novos índices foram construídos para as filas em índice que contêm as mensagens persistentes (descritas em [“Reconstruindo Índices de Fila”](#) na página 269).

Se BSDSs duais estiverem em uso, o IBM MQ verificará a consistência dos registros de data e hora no BSDS:

- Se ambas as cópias do BSDS forem atuais, o IBM MQ testará se os dois registros de hora são iguais. Se não forem, o IBM MQ emitirá a mensagem CSQJ120E e será finalizado. Isso pode acontecer quando duas cópias do BSDS forem mantidas em volumes DASD separados e um dos volumes for restaurado enquanto o gerenciador de filas for parado. IBM MQ detecta a situação na reinicialização.
- Se uma cópia do BSDS foi desalocada e a criação de log continuou com um único BSDS, poderia ocorrer um problema. Se ambas as cópias do BSDS forem mantidas em um único volume e o volume foi restaurado, ou se ambas as cópias BSDS foram restauradas separadamente, o IBM MQ talvez não detecte a restauração. Nesse caso, os registros de log não observados no BSDS seriam desconhecidos no sistema.

Os aplicativos em lote não são notificados quando ocorre a reinicialização *depois* que o aplicativo tiver solicitado uma conexão.

Entendendo um Intervalo de Log Necessário para Recuperação

Durante a reinicialização, o intervalo de dados do log que deve ser lido depende de vários fatores:

- No momento da finalização anormal, geralmente existem várias unidades de trabalho incompletas no sistema. Conforme descrito anteriormente, o processamento de reinicialização colocará o sistema em um estado de consistência, que pode envolver a restauração de unidade de trabalho em andamento ou recuperar bloqueios em unidades de trabalho em dúvida. A recuperação da unidade de trabalho requer que todos os registros de log da unidade de trabalho para unidades de trabalho em andamento, em restauração e em dúvida fiquem disponíveis. IBM MQ 'desviará' antigas unidades de trabalho, de modo que a recuperação da unidade de trabalho possa ser executada usando um intervalo muito menor de dados de log.

- No momento da finalização anormal, geralmente existem várias atualizações persistentes que são retidas apenas no cache do buffer pool. Elas ainda não foram gravadas no disco. Essas mudanças devem ser lidas a partir do log e reaplicadas para os dados retidos nos conjuntos de páginas. Os RBAs de recuperação do conjunto de páginas no ponto de verificação descrevem o menor RBA de log necessário para atualizar os conjuntos de páginas para um estado consistente.
- Se os antigos conjuntos de páginas foram introduzidos no sistema, por exemplo, um backup do conjunto de páginas foi introduzido para recuperar-se de um falha de mídia, todas as mudanças devem ser lidas do log a partir do momento em que o backup foi tirado. Essas mudanças foram reaplicadas para os dados retidos no conjunto de páginas sendo recuperado. Os RBAs de recuperação do conjunto de páginas retidos na página 0 do conjunto de páginas descrevem o menor RBA de log necessário para a recuperação de mídia de um conjunto de páginas.
- Se estiver usando mensagens persistentes em filas compartilhadas, um intervalo de dados de log será necessário para recuperar CFSTRUCTs que estão retendo mensagens persistentes. Os dados de log mais antigos, que seriam necessários para executar uma recuperação CFSTRUCT, são do horário aproximado do antigo CFSTRUCT BACKUP.

Durante a execução normal, o comando `DISPLAY USAGE TYPE(DATASET)` pode ser usado para visualizar o intervalo do log de recuperação associado a esses fatores (ele não consegue fornecer informações devido à reintrodução dos antigos conjuntos de página, obviamente). Para evitar quaisquer problemas que possam prolongar uma reinicialização do gerenciador de filas no caso de uma finalização anormal, monitore regularmente a saída de valores a partir do `DISPLAY USAGE TYPE(DATASET)`.

Além disso, o gerenciador de filas emite mensagens informativas relacionadas a esses fatores:

- `CSQJ160I` e `CSQJ161I` avisam de unidades de trabalho de longa execução.
- `CSQR026I` e `CSQR027I` fornecem informações sobre se essas unidades de trabalho de longa execução foram desviadas com êxito.
- `CSQE040I` e `CSQE041E` avisam que os backups de estrutura estão ficando antigos e conseqüentemente uma operação `RECOVER CFSTRUCT` demoraria um longo tempo.

Determinando Qual Aplicativo tem uma Unidade de Trabalho de Longa Execução

É possível determinar o aplicativo com a unidade de trabalho de longa execução. Para fazer isso, use o comando `DISPLAY CONN`.

O comando `DISPLAY CONN` retorna informações de conexão para todos os aplicativos conectados ao gerenciador de filas, juntamente com as informações adicionais que ajudam a determinar qual(is) aplicativo(s) possui(em) atualmente uma unidade de trabalho de longa execução. As informações retornadas pelo comando `DISPLAY CONN` são semelhantes às informações retornadas pelo comando `DISPLAY QSTATUS`, mas a principal diferença é que o `DISPLAY CONN` exibe informações sobre os objetos e informações transacionais para uma determinada conexão, em vez de detalhes de quais conexões estão associadas a um determinado objeto.

Para cada aplicativo conectado, o comando `DISPLAY CONN` retorna as seguintes informações:

- Informações básicas que incluem o ID de Conexão e o PID.
- As informações transacionais para essa conexão, incluindo a data e a hora em que a transação foi criada (ou seja, quando o primeiro `MQGET/PUT` foi criado sob o ponto de sincronização) e quando a transação foi gravada primeiro no log.
- As informações do horário do log que indicam qual aplicativo ainda tem uma unidade de trabalho de longa execução.
- Uma lista de todos os objetos que a conexão tem atualmente abertos. Os detalhes para cada objeto são retornados como uma mensagem separada, com um ID de Conexão usado como uma chave. Como existem diferentes tipos de objeto como filas e gerenciadores de fila, as informações exibidas com o objeto são específicas para seu tipo de objeto.

Reconstruindo Índices de Fila

Para aumentar a velocidade das operações MQGET em uma fila em que as mensagens não são recuperadas sequencialmente, é possível especificar que você deseja que o IBM MQ mantenha um índice da mensagem ou identificadores de correlação ou ID para todas as mensagens nessa fila.

Quando um gerenciador de filas for reiniciado, esses índices serão reconstruídos para cada fila. Isso se aplica apenas às mensagens persistentes; as mensagens não persistentes são excluídas na reinicialização. Se suas filas indexadas contiverem grandes números de mensagens persistentes, isso aumentará o tempo consumido para reiniciar o gerenciador de filas.

É possível optar por fazer com que os índices sejam reconstruídos de maneira assíncrona na inicialização do gerenciador de filas, usando o parâmetro QINDEXBLD da macro CSQ6SYSP. Se você configurar o QINDEXBLD=NOWAIT, o IBM MQ será reiniciado sem esperar que os índices sejam reconstruídos.

Como as Unidades de Recuperação em Dúvida São Resolvidas

Se o IBM MQ perder sua conexão com outro gerenciador de recursos, normalmente ele tentará recuperar todos os objetos inconsistentes na reinicialização.

Se o IBM MQ perder sua conexão com o CICS, IMS ou RRS, normalmente ele tentará recuperar todos os objetos inconsistentes na reinicialização. As informações necessárias para resolver as unidades de recuperação em dúvida devem vir do sistema de coordenadas. As próximas seções descrevem o processo de resolução para diferentes ambientes.

- [Como unidades de recuperação em dúvida são resolvidas por meio do CICS](#)
- [Como unidades de recuperação em dúvida são resolvidas por meio do IMS](#)
- [Como unidades de recuperação em dúvida são resolvidas por meio do RRS](#)
- [Como unidades de recuperação em dúvida com uma disposição de unidade de recuperação GROUP são resolvidas](#)

Como unidades de recuperação em dúvida são resolvidas a partir do CICS

Em algumas circunstâncias, o CICS não pode executar o processo IBM MQ para resolver unidades de recuperação em dúvida. Quando isso acontece, o IBM MQ envia uma das seguintes mensagens:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E

seguidas pela mensagem CSQC408I.

Para obter detalhes sobre o que significa essas mensagens, consulte o manual [IBM MQ for z/OS mensagens, conclusão, e códigos de razão](#).

A resolução de unidades em dúvida não afeta os recursos do CICS. CICS está no controle da coordenação de recuperação e, quando é reiniciado, confirma automaticamente ou restaura cada unidade, dependendo de se houve um registro de log marcando o início da confirmação. A existência de objetos em dúvida não bloqueia os recursos do CICS enquanto o IBM MQ estiver sendo reconectado.

Uma das funções do adaptador CICS é manter os dados sincronizados entre CICS e IBM MQ. Se um gerenciador de filas encerrar de forma anormal enquanto conectado ao CICS, é possível para o CICS confirmar ou restaurar o trabalho sem que o IBM MQ esteja ciente disso. Quando o gerenciador de filas for reiniciado, esse trabalho será denominado *em dúvida*.

IBM MQ não pode resolver essas unidades de recuperação em dúvida (ou seja, confirmar ou restaurar as mudanças feitas nos recursos do IBM MQ) até que a conexão com o CICS seja reiniciada ou reconectada.

Um processo para resolver unidades de recuperação em dúvida é iniciado durante a inicialização do adaptador CICS. O processo é iniciado quando o adaptador solicita uma lista de unidades de recuperação em dúvida. Em seguida:

- O adaptador recebe uma lista de unidades de recuperação em dúvida para este ID de conexão a partir do IBM MQ e as passa para o CICS para resolução.
- CICS compara as entradas dessa lista com as entradas em seu próprio log. CICS determina de sua própria lista qual ação que levou para cada unidade de recuperação em dúvida.

Para todas as unidades resolvidas, o IBM MQ atualiza as filas conforme necessário e libera os bloqueios correspondentes. As unidades não resolvidas podem permanecer após a reinicialização. Resolva-as pelos métodos descritos no [Administrando IBM MQ for z/OS](#).

Como unidades de recuperação em dúvida são resolvidas a partir do IMS

Resolver unidades de recuperação em dúvida no IMS não afeta os recursos DL/I. IMS está no controle da coordenação de recuperação e, quando é reiniciado, confirma automaticamente ou restaura o trabalho DL/I incompleto. A decisão de confirmar ou restaurar para regiões online (sem atalho de caminho) está na presença ou ausência de tipos de registros de log do IMS, X'3730' e X'3801'. A existência das unidades de recuperação em dúvida não indica que os registros DL/I estejam bloqueados até que o IBM MQ se conecte.

Durante a reinicialização de gerenciador de filas, o IBM MQ faz uma lista de unidades de recuperação em dúvida. IMS constrói sua própria lista de entradas de recuperação residual (RREs). As RREs são registradas em log no ponto de verificação de IMS até que todas as entradas sejam resolvidas.

Durante a reconexão de uma região IMS para IBM MQ, IMS indica para o IBM MQ se confirmar ou restaurar as unidades de trabalho marcadas em IBM MQ como em dúvida.

Quando as unidades em dúvida são resolvidas:

1. Se o IBM MQ reconhecer que marcou uma entrada para confirmação e IMS a tiver marcado para que seja restaurada, o IBM MQ emitirá a mensagem CSQQ010E. IBM MQ emite essa mensagem para todas as inconsistências desse tipo entre IBM MQ e IMS.
2. Se o IBM MQ tiver alguma unidade em dúvida restante, o adaptador emitirá a mensagem CSQQ008I.

Para todas as unidades resolvidas, o IBM MQ atualiza as filas conforme necessário e libera os bloqueios correspondentes.

IBM MQ mantém os blocos em trabalho em dúvida que não foram resolvidos. Isso pode causar uma lista não processada no sistema, se importantes blocos estiverem sendo retidos. A conexão permanece ativa para que você possa resolver as RREs do IMS. Recupere os encadeamentos em dúvida pelos métodos descritos no [Administrando IBM MQ for z/OS](#).

Todo o trabalho em dúvida deve ser resolvido, a menos que existam problemas operacionais ou no software, como com um cold start do IMS. A resolução em dúvida pela região de controle do IMS ocorre em duas circunstâncias:

1. No início da conexão com o IBM MQ, durante a qual a resolução é feita de maneira síncrona.
2. Quando um programa é finalizado de maneira anormal, durante o qual a resolução é feita de maneira assíncrona.

Como as Unidades de Recuperação em Dúvida São Resolvidas no RRS

Uma das funções do adaptador RRS é manter os dados sincronizados entre o IBM MQ e outros gerenciadores de recursos participantes do RRS. Se ocorrer uma falha quando o IBM MQ tiver concluído a fase um da confirmação e estiver aguardando uma decisão do RRS (o coordenador de confirmação), a unidade de recuperação entrará no estado em dúvida.

Quando a comunicação é restabelecida entre RRS e IBM MQ, o RRS confirmará ou restaurará automaticamente cada unidade de recuperação, dependendo de se houve um registro de log marcando o início da confirmação. IBM MQ não pode resolver essas unidades de recuperação em dúvida (ou seja, confirmar ou restaurar as mudanças feitas nos recursos do IBM MQ) até que a conexão com o RRS seja restabelecida.

Em algumas circunstâncias, o RRS não pode resolver unidades de recuperação em dúvida. Quando isso acontece, o IBM MQ envia uma das seguintes mensagens para o console do z/OS:

- CSQ3011I
- CSQ3013I
- CSQ3014I
- CSQ3016I

Para obter detalhes sobre o que significa essas mensagens, consulte o manual [IBM MQ for z/OS mensagens, conclusão, e códigos de razão](#).

Para todas as unidades de recuperação resolvidas, o IBM MQ atualiza as filas conforme necessário e libera os bloqueios correspondentes. As unidades de recuperação não resolvidas podem permanecer após a reinicialização. Resolva-as pelo método descrito no [Administrando IBM MQ for z/OS](#).

Como as Unidades de Recuperação em Dúvida com uma Disposição da Unidade de Recuperação GROUP São Resolvidas

As transações em dúvida que possuem uma disposição da unidade de recuperação GROUP podem ser resolvidas pelo coordenador de transação por qualquer gerenciador de filas no grupo de filas compartilhadas (QSG), no qual o atributo do gerenciador de filas GROUPUR está ativado. Sempre que um coordenador de transação se reconecta, geralmente ele solicita uma lista de quaisquer transações pendentes em dúvida e, em seguida, as resolve usando as informações de seus logs.

Quando um coordenador de transação, que se conectou a uma disposição da unidade de recuperação GROUP, solicita a lista de transações em dúvida, a lista retornada inclui todas as transações em dúvida com uma disposição de unidade de recuperação GROUP que existe em todo o grupo de filas compartilhadas. Esta lista não depende do gerenciador de filas no qual as transações em dúvida foram iniciadas. Um gerenciador de filas processando tal solicitação compila a lista comunicando-se com todos os outros gerenciadores de filas ativos no grupo de filas compartilhadas usando o SYSTEM.QSG.UR.RESOLUTION.QUEUE. O gerenciador de filas então lê os logs de quaisquer gerenciadores de filas inativos, a partir de seus últimos pontos de verificação, para identificar qualquer transação adicional em dúvida que eles tenham relatado, caso tenham ficado ativos.

Quando um coordenador de transação solicita a resolução de uma transação em dúvida, o gerenciador de filas ao qual está conectado identifica se a transação foi originada de si mesmo e, nesse caso, resolve isso da mesma maneira que as transações com uma disposição da unidade de recuperação QMGR. Se a transação foi originada em outro gerenciador de filas ativo no QSG, uma solicitação para concluir a resolução será roteada para esse gerenciador de filas usando o SYSTEM.QSG.UR.RESOLUTION.QUEUE. No caso em que a transação foi originada em um gerenciador de filas inativo no QSG, qualquer trabalho de fila compartilhada é resolvido imediatamente e uma solicitação para resolver qualquer trabalho de fila privado restante é colocada no SYSTEM.QSG.UR.RESOLUTION.QUEUE. O gerenciador de filas inativo processa essa solicitação após a inicialização antes de aceitar o novo trabalho. Neste cenário, os logs do gerenciador de filas original ainda refletirão que a unidade de recuperação está em dúvida até que tenha reiniciado e processado a solicitação.

Recuperação de Fila Compartilhada

Use este tópico para entender a recuperação do IBM MQ e a resiliência de vários componentes no ambiente do grupo de filas compartilhadas.

- [“Recuperação Transacional” na página 273](#)
- [“Recuperação Peer-to-peer” na página 273](#)
- [“Definições da Fila Compartilhada” na página 273](#)
- [“Criação de Log” na página 274](#)
- [“Falhas da Estrutura e do Recurso de Acoplamento” na página 274](#)
- [“Cenários de Falha de Estrutura” na página 275](#)

- [“Resiliência para Falhas de Conectividade do Recurso de Acoplamento” na página 276](#)
- [“Gerenciando Resiliência para Falhas de Conectividade do Recurso de Acoplamento” na página 277](#)
- [“Comportamento Operacional” na página 279](#)

Recuperação Transacional

Quando um aplicativo emite uma chamada MQBACK ou é encerrado de forma anormal (por exemplo, devido a uma EXEC CICS ROLLBACK ou uma finalização anormal do IMS), informações de nível de encadeamento armazenadas no gerenciador de filas asseguram que a unidade de trabalho em andamento seja recuperada. As operações MQPUT e MQGET no ponto de sincronização nas filas compartilhadas são retrocedidas da mesma maneira que as atualizações nas filas não compartilhadas.

Recuperação Peer-to-peer

Se o gerenciador de filas falhar, ele será desconectado de forma anormal das estruturas do recurso de acoplamento às quais está atualmente conectado. Se a conexão entre a instância do z/OS e o recurso de acoplamento falhar (por exemplo, falha de link físico ou desligamento de uma partição ou recurso de acoplamento) isso também será detectado como uma finalização anormal da conexão entre o gerenciador de filas e as estruturas do recurso de acoplamento envolvidos. Outros gerenciadores de filas no mesmo grupo de filas compartilhadas que permanecem conectados a essa estrutura detectam a desconexão anormal e todas as tentativas de iniciar a *recuperação peer-to-peer* para o gerenciador de filas com falha nessa estrutura. Apenas um desses gerenciadores de filas inicia a recuperação peer-to-peer com êxito, mas todos os outros gerenciadores de filas cooperam na recuperação de unidades de trabalho que eram de propriedade do gerenciador de filas que falhou.

Se um gerenciador de filas falhar quando não houver peers conectados a uma estrutura, a recuperação será executada quando outro gerenciador de filas se conectar a essa estrutura ou quando o gerenciador de filas que falhou for reiniciado.

A recuperação peer-to-peer, geralmente referida como Peer Level Recovery (PLR), é executada em uma base estrutura por estrutura e é possível que um único gerenciador de filas participe da recuperação de mais de uma estrutura ao mesmo tempo. No entanto, o conjunto de peers que coopera na recuperação de diferentes estruturas pode variar dependendo de quais gerenciadores de filas foram conectados às diferentes estruturas no momento da falha.

Quando o gerenciador de filas com falha for reiniciado, ele se reconectará às estruturas às quais estava conectado no momento da falha e recuperará todas as unidades de trabalho não resolvidas restantes que não foram recuperadas pela recuperação peer-to-peer.

A recuperação peer-to-peer é um processo de várias fases. Durante a primeira fase, as unidades de trabalho que tinha avançado além da fase em andamento serão recuperadas; isso pode envolver mensagens para unidades de trabalho que são mensagens em confirmação e de bloqueio para unidades de trabalho que estão em dúvida. Durante a segunda fase, as filas que tinham encadeamentos ativos no gerenciador de filas com falha serão verificadas, as mensagens não confirmadas relacionadas às unidades de trabalho em andamento serão retrocedidas e as informações sobre as manipulações ativas nas filas compartilhadas no gerenciador de filas com falha serão reconfiguradas. Isso significa que o IBM MQ reconfigura quaisquer indicadores que o gerenciador de filas com falha tinham para uma fila compartilhada aberta para entrada exclusiva, permitindo que outros gerenciadores de filas ativos abrissem a fila para entrada.

Definições da Fila Compartilhada

Os objetos de fila que representam os atributos de uma fila compartilhada são retidos no repositório compartilhado do Db2 usado pelo grupo de filas compartilhadas. Assegure-se de que os procedimentos adequados estejam acessíveis para o backup e a recuperação do Db2 usado para reter objetos do IBM MQ. Você também pode usar o utilitário CSQUTIL do IBM MQ para criar comandos MQSC para reprodução

em um gerenciador de filas para redefinir os objetos IBM MQ, incluindo definições de grupo e fila compartilhada armazenadas em Db2.

Criação de Log

Os grupos de filas compartilhadas podem suportar mensagens persistentes porque as mensagens em filas compartilhadas podem ser registradas nos logs do gerenciador de filas.

Falhas da Estrutura e do Recurso de Acoplamento

Há dois tipos de falha que podem ser restauradas para uma estrutura do recurso de acoplamento (CF): falha de estrutura e perda de conectividade. Serviços sysplex para compartilhamento de dados (XES) informam o IBM MQ de uma falha na estrutura CF ou de uma falha CF com um evento de falha de estrutura. Se o XES criar uma perda de evento de conectividade isso não indica necessariamente que existe um problema na estrutura, pode ser que não exista nenhuma conexão disponível para se comunicar com a estrutura. É possível que nem todos os gerenciadores de filas recebam uma perda de evento de conectividade para a estrutura; isso depende da configuração de conexões para o CF. Um evento de perda de conectividade também pode ser recebido devido a comandos do operador, por exemplo, VARY PATH OFFLINE ou CONFIG CHP OFFLINE.

As estruturas CF que são usadas pelo IBM MQ podem ser configuradas para usar a duplexação gerenciada por sistema. Isso significa que se houver uma única falha, o processamento de failover gerenciado pelo sistema ocultará a falha de uma estrutura ou a perda de conectividade e o gerenciador de filas não será informado da falha. Se houver uma falha de ambas as instâncias de uma estrutura duplexada ou conexão, o gerenciador de filas receberá o evento aproximado e manipulará isso da mesma maneira que um evento de falha para uma estrutura simplex. Os detalhes de como o gerenciador de filas manipula os eventos estão descritos nos [Cenários](#).

No improvável caso de uma falha na estrutura ou CF, quaisquer mensagens não persistentes armazenadas nas estruturas do aplicativo afetadas serão perdidas. É possível recuperar as mensagens persistentes usando o comando RECOVER CFSTRUCT. Se a estrutura do aplicativo recuperável tiver falhado, qualquer atividade de aplicativo adicional para essa estrutura será evitada até que a estrutura tenha sido recuperada.

Para assegurar que seja possível recuperar uma estrutura CF em um período de tempo razoável, execute backups frequentes, usando o comando BACKUP CFSTRUCT. É possível optar por executar os backups em qualquer gerenciador de filas no grupo de filas compartilhadas ou dedicar um gerenciador de filas para executar todos os backups. Automatize o processo de executar os backups para assegurar que são executados regularmente.

Cada backup é gravado no conjunto de dados de log ativo do gerenciador de filas que executa o backup. A fila compartilhada do repositório do Db2 registra o nome da estrutura CF que está tendo backup, o nome do gerenciador de filas fazendo o backup, o intervalo em RBA para esse backup no log do gerenciador de filas, e o tempo do backup.

A estrutura de administração contém informações sobre as unidades de trabalho incompletas nas filas compartilhadas no momento de qualquer falha na estrutura do aplicativo para que a estrutura de administração fique disponível durante o processamento RECOVER CFSTRUCT. Se a estrutura de administração tiver falhado, todos os gerenciadores de fila no grupo de filas compartilhadas deverão ter reconstruídas suas entradas da estrutura de administração antes que o comando RECOVER CFSTRUCT possa ser emitido.

Os gerenciadores de filas reconstróem suas entradas de estrutura de administração automaticamente e sem finalização. Se um gerenciador de filas não estiver em execução no momento da falha, suas entradas da estrutura de administração poderão ser reconstruídas por outro gerenciador de filas no grupo de filas compartilhadas que está em execução no mesmo nível ou nível superior.

Para recuperar uma estrutura do aplicativo, emita um comando RECOVER CFSTRUCT para o gerenciador de filas que deseja para executar a recuperação. É possível recuperar uma única estrutura CF ou é possível recuperar diversas estruturas CF simultaneamente. É possível recuperar o uso de qualquer

gerenciador de filas no grupo de filas compartilhadas. Ele não precisa ser aquele que executou o backup ou aquele que foi conectado anteriormente à estrutura com falha.

O comando RECOVER CFSTRUCT usa o backup, localizado por meio do repositório de informações do Db2 (Db2 deve, portanto, estar disponível no gerenciador de filas no qual a recuperação está sendo realizada) e recupera isso para o ponto da falha.

O comando RECOVER CFSTRUCT faz isso aplicando registros de log de cada gerenciador de filas no grupo de filas compartilhadas que executou um MQPUT ou MQGET entre o início do backup e o horário da falha, para qualquer fila compartilhada que mapeia para a estrutura de CF. A mesclagem resultante dos logs pode requerer a leitura de uma quantidade considerável de dados do log, pois todos os dados do log gravados por gerenciadores de filas participantes desde o backup são lidos. É altamente recomendado que execute frequentes backups (por exemplo, de hora em hora), especialmente se houver grandes mensagens no backup.

Cenários de Falha de Estrutura

Cenários

Se for relatada uma falha para uma estrutura CF, a ação obtida pelos gerenciadores de filas conectados depende do seguinte:

- O tipo de falha relatada pelo componente XES do z/OS para o IBM MQ.
- O tipo de estrutura (aplicativo ou requisição)
- O nível do gerenciador de filas
- O CFLEVEL do objeto CFSTRUCT IBM MQ (2, 3, 4 ou 5. Este não é o CFLEVEL do microcódigo CFCC)
- O atributo RECAUTO de um objeto CFSTRUCT do IBM MQ no CFLEVEL(5)

Os seguintes cenários descrevem o que acontece quando uma falha é relatada para a estrutura de administração:

- Se um evento de falha de estrutura for recebido para a estrutura de administração, a estrutura será realocada e reconstruída automaticamente sem a finalização do gerenciador de filas. Uma nova instância da estrutura é alocada pelo XES (Cross-system Extended Services) quando um gerenciador de filas tenta se conectar a ela.

Quando o gerenciador de filas se conecta à nova instância da estrutura, ele grava as entradas para si mesmo na estrutura. Este processamento é executado pelo gerenciador de filas e não faz parte do processamento de reconstrução XES.

Se um gerenciador de filas não estava em execução no momento da falha ou se ele foi finalizado antes de a recuperação de sua parte da estrutura de administração ter sido concluída, suas entradas da estrutura de administração poderão ser reconstruídas por outro gerenciador de filas no grupo de filas compartilhadas.

As entradas da estrutura de administração de um gerenciador de filas só poderão ser reconstruídas por outro gerenciador de filas em execução no mesmo nível ou nível superior. Se as entradas da estrutura de administração de um gerenciador de filas não puderem ser reconstruídas por outro gerenciador de filas no grupo de filas compartilhadas, reinicie o gerenciador de filas para que ele possa concluir a reconstrução de sua parte da estrutura.

Determinadas ações são suspensas até que as entradas da estrutura de administração para todos os gerenciadores de filas tenham sido reconstruídas. As ações suspensas incluem o seguinte:

- Abrir e fechar as filas compartilhadas.
- Confirmar ou restaurar as unidades de recuperação.
- Aplicativos serializados que se conectam a ou desconectam do gerenciador de filas.
- Fazendo backup ou recuperando uma estrutura do aplicativo.

Quaisquer aplicativos serializados que já tenha se conectado ao gerenciador de filas podem continuar o processamento. Todo aplicativo serializado que tenta se conectar aos parâmetros

MQCNO_SERIALIZE_CONN_TAG_QSG ou MQCNO_RESTRICT_CONN_TAG_QSG recebe o código de retorno MQRC_CONN_TAG_NOT_USABLE.

Quando as entradas da estrutura de administração para o gerenciador de filas tiverem sido reconstruídas, as ações suspensas serão continuadas.

Os seguintes cenários descrevem o que acontece quando uma falha é relatada para uma estrutura do aplicativo:

- Se um evento de falha na estrutura for recebido para uma estrutura do aplicativo, e o CFLEVEL for 1 ou 2, o gerenciador de filas será finalizado. Reiniciar o gerenciador de filas. O primeiro gerenciador de filas a tentar conectar à estrutura novamente faz com que XES aloque uma nova instância da estrutura.
- Se um evento de falha na estrutura for recebido para uma estrutura do aplicativo, e o CFLEVEL for 3, 4 ou 5, os gerenciadores de filas conectados à estrutura continuarão a ser executados. Os aplicativos que não usarem as filas na estrutura com falha podem continuar o processamento normal.

No entanto, os aplicativos que tentarem operações em filas na estrutura com falha receberão um erro MQRC_CF_STRUC_FAILED até que a estrutura tenha sido reconstruída com sucesso, ponto em que o aplicativo poderá abrir as filas novamente.

A reconstrução da estrutura é iniciada automaticamente para as estruturas do aplicativo CFLEVEL (5) definidas com RECAUTO(YES). Caso contrário, a estrutura será reconstruída quando o comando RECOVER CFSTRUCT for emitido.

Resiliência para Falhas de Conectividade do Recurso de Acoplamento

O que é resiliência para falhas de conectividade do recurso de acoplamento?

A resiliência a falhas de conectividade do recurso de acoplamento se refere à capacidade dos gerenciadores de filas em um grupo de filas compartilhadas de tolerar a perda de conectividade com uma estrutura do recurso de acoplamento sem serem finalizados. Essa função também tenta reconstruir a estrutura em outro recurso de acoplamento com melhor conectividade para obter acesso novamente a filas compartilhadas o mais rápido possível.

O que é perda parcial de conectividade?

IBM MQ define como perda parcial de conectividade aquela situação em que um ou mais sistemas no sysplex perdem a conectividade com o recurso de acoplamento no qual a estrutura sendo acessada pelo sistema é alocada, mas pelo menos um sistema no sysplex mantém a conectividade com o mesmo recurso de acoplamento.

O que é perda total de conectividade?

IBM MQ define como perda total de conectividade a situação em que nenhum sistema no sysplex tem conectividade com o recurso de acoplamento e com a estrutura alocada dentro dele.

Por que você ativaria esta função?

A resiliência para falhas de conectividade do recurso de acoplamento melhora a disponibilidade do IBM MQ, permitindo que filas não compartilhadas permaneçam disponíveis após um gerenciador de filas perder a conectividade com uma ou mais estruturas do recurso de acoplamento. Além disso, os gerenciadores de filas que perdem a conectividade com uma estrutura do recurso de acoplamento tentam automaticamente reconstruir a estrutura em outro recurso de acoplamento disponível, melhorando a disponibilidade das filas compartilhadas dentro do grupo de filas compartilhadas.

Considerações ao ativar essa função

Um gerenciador de filas que tolera perda de conectividade com estruturas do recurso de acoplamento sem ser finalizado pode não conseguir se reconectar a uma estrutura do recurso de acoplamento por algum tempo se não houver um recurso de acoplamento alternativo disponível. Filas compartilhadas definidas em uma estrutura que sofreu perda de conectividade permanecem indisponíveis até a conectividade com a estrutura ser restaurada. Nesta situação, os aplicativos que se conectam aos

membros do grupo de filas compartilhadas para executar o trabalho de fila compartilhada podem achar que as filas compartilhadas que precisam acessar não estão disponíveis. Para evitar essa situação, é recomendado que os gerenciadores de filas sejam configurados para serem finalizados quando a conectividade com uma estrutura do recurso de acoplamento for perdida. Essa finalização força os aplicativos a se conectarem a outro membro do grupo de filas compartilhadas que tenha conectividade com as estruturas do recurso de acoplamento nas quais as filas compartilhadas que o aplicativo requer são definidas.

Gerenciando Resiliência para Falhas de Conectividade do Recurso de Acoplamento

Como ativar essa funcionalidade?

As etapas a seguir devem ser executadas para a ativação da resiliência para conectividade do recurso de acoplamento.

1. Assegure-se de que o Couple Data Set CFRM tenha sido formatado para suportar a reconstrução gerenciada pelo sistema. Isso permite que os gerenciadores de filas iniciem uma reconstrução gerenciada pelo sistema para recriar uma estrutura em um recurso de acoplamento disponível. Use o comando **DISPLAY XCF, COUPLE, TYPE=CFRM** para determinar o formato do Couple Data Set CFRM. Para suportar a reconstrução gerenciada pelo sistema, o couple data set CFRM deve ser formatado especificando:

```
"ITEM NAME(SMREBLD) NUMBER(1) "
```

Consulte a documentação [z/OS MVS Configurando um Sysplex](#) para obter mais informações sobre como formatar um par de dados CFRM.

2. Assegure que um recurso de acoplamento alternativo esteja disponível e esteja na lista de preferência do CFRM para todas as estruturas do recurso de acoplamento do IBM MQ. Isso permite que os gerenciadores de filas tentem reconstruir estruturas em um recurso de acoplamento alternativo disponível para restaurar acesso às estruturas o mais rápido possível.

As estruturas de IBM MQ devem ser definidas com ENFORCEORDER(NO) na política do CFRM, de modo que XCF seja capaz de escolher o CF ideal na configuração se o IBM MQ precisar realocar a estrutura.

Consulte a documentação [z/OS MVS Configurando um Sysplex](#) para obter mais informações sobre as listas de preferências de estrutura
3. Altere para CFLEVEL(5) todas as estruturas do recurso de acoplamento do aplicativo que precisam tolerar perda de conectividade. Esse é o nível mínimo que pode tolerar uma perda de conectividade.
4. Determine os valores necessários para os atributos **QMGR CFCONLOS** e **CFSTRUCT CFCONLOS** e altere-os adequadamente. O atributo **QMGR CFCONLOS** controla se a perda de conectividade com a estrutura de administração é tolerada, e o atributo **CFSTRUCT CFCONLOS** controla se a perda de conectividade é tolerada por cada estrutura do recurso de acoplamento do aplicativo. Se os valores padrão para esses atributos forem retidos, o gerenciador de filas é finalizado segundo a perda de conectividade com qualquer estrutura do recurso de acoplamento.
5. Determine os valores necessários para o atributo **CFSTRUCT RECAUTO** para cada estrutura do recurso de acoplamento do aplicativo e altere-os de acordo. Esse atributo controla se as estruturas do recurso de acoplamento devem ser recuperadas automaticamente usando dados registrados após a perda total de conectividade. Se o valor padrão para esse atributo for retido, nenhuma recuperação automática é executada para estruturas de aplicativos segundo a perda total de conectividade

Cenário 1 - Perda de conectividade com a estrutura de administração

Os gerenciadores de filas podem tolerar a perda de conectividade com a estrutura de administração sem finalizar

Quando a conectividade com a estrutura de administração é perdida por qualquer gerenciador de filas que tenha sido configurado para tolerar a perda de conectividade com a estrutura de

administração, todos os membros do grupo de filas compartilhadas são desconectados da estrutura de administração. Então, todos os gerenciadores de filas ativos no grupo de filas compartilhadas tentam se reconectar à estrutura de administração fazendo com que ela seja realocada no recurso de acoplamento com a melhor conectividade com todos os sistemas no sysplex e reconstrua os dados da estrutura de administração.

Nota: Esse pode não ser necessariamente o recurso de acoplamento com a melhor conectividade com todos os sistemas com gerenciadores de filas ativos.

Se um gerenciador de filas não puder reconectar à estrutura de administração, por exemplo, porque nenhuma das estruturas de acoplamento na lista de preferências de CFRM para a estrutura de administração está disponível, algumas operações da fila compartilhada permanecem indisponíveis até o gerenciador de filas poder reconectar com sucesso à estrutura de administração e reconstruir seus dados da estrutura de administração. A reconexão ocorre automaticamente quando um recurso de acoplamento adequado se torna disponível no sistema.

A falha na conexão com a estrutura de administração durante a inicialização do gerenciador de filas como resultado da ausência de conectividade com o recurso de acoplamento ou ausência de um recurso de acoplamento adequado disponível para alocar a estrutura, não é tolerada. Então, todos os gerenciadores de filas ativos no grupo de filas compartilhadas tentam se reconectar à estrutura de administração fazendo com que ela seja realocada em outro recurso de acoplamento, se houver um disponível, e reconstrua os dados da estrutura de administração.

Cenário 2 - Perda de conectividade com a estrutura do aplicativo

A perda de conectividade com estruturas do aplicativo em **CFLEVEL (5)** ou superior pode ser tolerada sem que o gerenciador de filas seja finalizado. Gerenciadores de filas conectados às estruturas do aplicativo em **CFLEVEL (4)** ou inferior ou às estruturas em **CFLEVEL (5)** que não foram configuradas para tolerar perda de conectividade são encerrados de forma anormal com o código de razão 00C510AB quando a conectividade com a estrutura for perdida.

Quando a conectividade for perdida em uma estrutura do aplicativo que foi configurada para tolerar perda de conectividade, todos os gerenciadores de filas que perderam conectividade com a estrutura são desconectados. O comportamento subsequente do gerenciador de filas depende de se a perda de conectividade é parcial ou total.

Perda parcial de conectividade com uma estrutura do aplicativo

Se a perda de conectividade for determinada como parcial, os gerenciadores de filas que perderam a conectividade com a estrutura tentam iniciar uma reconstrução gerenciada pelo sistema para mover a estrutura para outro recurso de acoplamento com melhor conectividade. Se essa reconstrução for bem-sucedida, mensagens persistentes e não persistentes na estrutura são copiadas para outro recurso de acoplamento e acesso às filas na estrutura é restaurado. Os gerenciadores de filas que não perderam conectividade permanecem conectados à estrutura; no entanto, as operações que acessam a estrutura são atrasadas durante o processo de reconstrução gerenciado pelo sistema.

Se uma estrutura do aplicativo não puder ser reconstruída para outro recurso de acoplamento com melhor conectividade ou se alguns gerenciadores de filas ainda não tiverem conectividade com a estrutura após ter sido reconstruída em outro recurso de acoplamento, filas definidas na estrutura permanecem indisponíveis nos gerenciadores de filas que não têm conectividade com a estrutura até a conectividade ser restaurada com o recurso de acoplamento. Os gerenciadores de filas reconectam automaticamente à estrutura quando estiver disponível e acesso às filas compartilhadas definidas na estrutura é restaurado.

Perda total de conectividade com uma estrutura do aplicativo

Se todos os sistemas MVS no sysplex tiverem perdido conectividade com o recurso de acoplamento no qual a estrutura do aplicativo está alocada, o z/OS desalocará a estrutura do recurso de acoplamento sempre que for feita uma tentativa de reconectar à estrutura. É possível que o gerenciador de filas tente se reconectar à estrutura por diversas razões, como uma tentativa de um aplicativo de abrir uma fila compartilhada ou uma notificação do sistema de que novos recursos de acoplamento podem estar disponíveis. É, portanto, provável que todas as mensagens

não persistentes na estrutura afetada sejam perdidas seguindo a perda total de conectividade com uma estrutura do aplicativo.

Estruturas de aplicativos recuperáveis são recuperadas automaticamente segundo a perda total de conectividade, se tiverem sido definidas com **RECAUTO(YES)**. A recuperação começa quase que imediatamente se um recurso de acoplamento alternativo estiver disponível para alocar a estrutura ou sempre que um recurso de acoplamento desse tipo ficar disponível. Se uma estrutura não foi definida com **RECAUTO(YES)**, a recuperação poderá ser iniciada emitindo o comando **RECOVER CFSTRUCT**. Isso recupera todas as mensagens persistentes na estrutura, mas todas as mensagens não persistentes são perdidas. Como esse processo envolve a leitura do log de gerenciador de filas, ele pode levar algum tempo para ser concluído, portanto, é recomendado fazer backups da estrutura regularmente para reduzir a quantidade de tempo levada para que o acesso às filas compartilhadas na estrutura seja restaurado.

Gerenciadores de filas tentam reconectar a estruturas de aplicativos não recuperáveis assim que um aplicativo tenta abrir uma fila compartilhada definida na estrutura ou que uma notificação é recebida do sistema de que novos recursos de acoplamento estão disponíveis. Se um recurso de acoplamento adequado estiver disponível para alocar a estrutura, uma nova estrutura é alocada e acesso às filas compartilhadas definidas na estrutura é restaurado. Como mensagens persistentes não podem ser colocadas em filas definidas em estruturas não recuperáveis, todas as mensagens nas filas compartilhadas são perdidas.

Comportamento Operacional

Se um gerenciador de filas IBM WebSphere MQ 7.1 ou mais recente, configurado para tolerar a perda de conectividade com uma estrutura de recurso de acoplamento específica perde conectividade, os membros do grupo de compartilhamento de fila tentam se recuperar automaticamente da falha e se reconectar à estrutura. Essa atividade pode envolver realocação de estrutura em outro recurso de acoplamento com melhor conectividade, caso haja algum disponível. Porém, ainda pode ser necessária a intervenção do operador para uma recuperação de perda de conectividade.

Normalmente a ação necessária do operador é:

1. Resolver a causa da falha que resultou na perda de conectividade.
2. Assegurar que um recurso de acoplamento onde as estruturas do IBM MQ podem ser alocadas esteja disponível em todos os sistemas no sysplex

Qualquer estrutura que foi automaticamente realocada em outro recurso de acoplamento após a perda de evento de conectividade, pode ser movida para o recurso de acoplamento com a conectividade ideal para todos os gerenciadores de filas no grupo de filas compartilhadas. Se necessário, isso poderá ser feito iniciando o comando de reconstrução gerenciado pelo sistema **SETXCF START,REBUILD** conforme documentado em [z/OS MVS Referência de Comandos do Sistema](#).

No caso de uma perda parcial de conectividade com uma estrutura do aplicativo, os gerenciadores de filas que perderam a conectividade com a estrutura tentam iniciar uma reconstrução gerenciada pelo sistema. Esse processo aloca a estrutura em outro recurso de acoplamento somente se esse recurso de acoplamento tiver conectividade com todos os gerenciadores de filas atualmente conectados à estrutura. Portanto, é possível que nos locais em que a maioria dos gerenciadores de filas em um grupo de filas compartilhadas tenha perdido a conectividade com uma estrutura de aplicativo, eles não sejam capazes de reconstruir a estrutura em outro recurso de acoplamento devido aos gerenciadores de filas que ainda estão conectados à estrutura original. Nessa situação, os gerenciadores de filas que ainda estão conectados à estrutura original podem ser encerrados para permitir que a estrutura seja reconstruída ou o comando **RESET CFSTRUCT ACTION(FAIL)** pode ser emitido para causar a falha da estrutura. A recuperação pode ser iniciada em estruturas aplicáveis pela emissão do comando **RECOVER CFSTRUCT**.

Nota: Ao falhar e recuperar a estrutura, todas as mensagens não persistentes na estrutura são perdidas.

Conceitos de segurança no IBM MQ for z/OS

Use este tópico para entender a importância da segurança para o IBM MQ, e as implicações de não ter configurações de segurança adequadas em seu sistema.

Por que você deve proteger os recursos do IBM MQ

O IBM MQ manipula a transferência de informações que são potencialmente de valor. Aplicar a segurança assegura que os recursos que o IBM MQ possui e gerencia sejam protegidos contra acesso não autorizado. Esse acesso pode levar à perda ou divulgação das informações.

É necessário assegurar que nenhum dos recursos a seguir sejam acessados ou mudados por qualquer usuário ou processo não autorizado:

- Conexões com o IBM MQ
- Objetos do IBM MQ tais como filas, processos e listas de nomes
- Links de transmissão do IBM MQ, ou seja, canais do IBM MQ
- Comandos de controle do sistema IBM MQ
- Mensagens IBM MQ
- Informações de contexto associadas às mensagens

Para fornecer a segurança necessária, o IBM MQ usa o System Authorization Facility (SAF) do z/OS para rotear solicitações de autorização para um External Security Manager (ESM), por exemplo Security Server (anteriormente conhecido como RACF). O IBM MQ não faz nenhuma verificação de segurança por iniciativa própria. Onde o enfileiramento distribuído ou clientes estão sendo usados, você pode requerer medidas de segurança adicionais, para as quais o IBM MQ fornece registros de autenticação de canal, saídas do canal, o atributo do canal MCAUSER e TLS.

A decisão para permitir acesso a um objeto é tomada pelo ESM e o IBM MQ segue essa decisão. Se o ESM não puder tomar uma decisão, o IBM MQ impedirá o acesso ao objeto.

O que acontece se você não proteger os recursos do IBM MQ

Se você não fizer nada com relação à segurança, o efeito mais provável é que *todos* os usuários possam acessar e alterar *cada* recurso. Isto inclui não apenas usuários locais, mas também aqueles em sistemas remotos que usam enfileiramento distribuído ou clientes, em que os controles de segurança de logon podem ser menos restritos do que é normalmente o caso para z/OS.

Para ativar a verificação de segurança, você deve fazer o seguinte:

- Instalar e ativar um ESM (por exemplo, o Security Server).
- Definir a classe MQADMIN se estiver usando um ESM diferente do Security Server.
- Ativar a classe MQADMIN.

Você deve considerar se usar nomes de recurso compostos por letras maiúsculas e minúsculas seria benéfico para sua empresa. Se você usar nomes de recurso compostos por letras maiúsculas e minúsculas em seus perfis do ESM, deverá definir e ativar a classe MXADMIN.

Data Set Encryption do z/OS

O Data Set Encryption (DSE) fornece a capacidade para criptografar os conjuntos de dados do z/OS para que os dados que eles contêm só possam ser visualizados ou modificados por IDs de usuários com a permissão específica concedida. Isso fornece criptografia de dados em repouso no sistema de arquivos e evita a divulgação inadequada de informações confidenciais para usuários que têm uma legítima necessidade de negócios e permissões para gerenciar os conjuntos de dados em si.

Antes da IBM MQ for z/OS 9.1.4, o IBM MQ for z/OS não suportava o uso de DSE com logs ativos, conjuntos de páginas e conjuntos de dados de mensagens compartilhadas (SMDS) que fornecem os mecanismos de persistência primária para as mensagens do IBM MQ.

Em vez disso, o [Advanced Message Security](#) fornece uma solução de criptografia de ponta a ponta para o sistema de mensagens do IBM MQ, que abrange toda a rede do IBM MQ, criptografia de dados em andamento, em repouso e até mesmo dentro dos processos do IBM MQ de tempo de execução.

Outros conjuntos de dados VSAM e sequenciais usados em um subsistema IBM MQ podem ser criptografados usando o DSE. Por exemplo:

- Conjunto de dados de autoinicialização (BSDS)
- Arquivos sequenciais que retêm comandos de configuração do sistema (MQSC) lidos na inicialização usando CSQINPx DDNAMEs
- Logs de archive do IBM MQ, geralmente usados para arquivamento de longo prazo de dados do log do IBM MQ para propósitos de auditoria.

É possível criptografar usando o DSE alocando uma classe de dados que é definida com um rótulo de chave do conjunto de dados. Para obter mais informações, veja [Planejando seu armazenamento de archive de log](#).

A partir da IBM MQ for z/OS 9.1.4, o IBM MQ for z/OS suporta o uso de DSE com os logs e os conjuntos de páginas ativos, além do suporte fornecido em liberações anteriores.

O IBM MQ for z/OS não suporta o uso de DSE para conjuntos de dados de mensagens compartilhadas (SMDS).

Consulte a seção [Confidencialidade para dados em repouso no IBM MQ for z/OS com criptografia do conjunto de dados](#) para obter informações adicionais.

Conceitos relacionados

[Conceitos de Segurança](#)

[Registros de Autenticação de Canal](#)

[Autoridade para trabalhar com objetos do IBM MQ no z/OS](#)

[Protocolos de segurança criptográficos: TLS](#)

Tarefas relacionadas

[Configurando a Segurança em z/OS](#)

[Comparando a segurança no nível do link com a segurança no nível do aplicativo](#)

Referências relacionadas

[Mensagens do IBM MQ for z/OS](#)

Controle de segurança e opções no IBM MQ for z/OS

É possível especificar se a segurança está ativada para o subsistema IBM MQ inteiro, e se você deseja executar verificações de segurança no nível do gerenciador de filas ou do grupo de filas compartilhadas. Também é possível controlar o número de IDs de usuário verificados para a segurança de recurso de API.

Segurança do Subsistema

A segurança do subsistema é um controle que especifica se alguma verificação de segurança é feita para o gerenciador de filas inteiro. Se você não precisar de verificação de segurança (por exemplo, em um sistema de teste) ou se estiver satisfeito com o nível de segurança em todos os recursos que podem se conectar ao IBM MQ (incluindo clientes e canais), será possível desligar a verificação de segurança para o gerenciador de filas ou para o grupo de filas compartilhadas para que nenhuma verificação de segurança adicional seja feita.

Essa é a única verificação que pode desligar a segurança completamente e determinar se alguma outra verificação de segurança é executada ou não. Ou seja, se você desligar a verificação para o gerenciador de filas ou para o grupo de filas compartilhadas, nenhuma outra verificação do IBM MQ será feita. Se você a deixar ligada, o IBM MQ verificará seus requisitos de segurança para outros recursos do IBM MQ.

Você também pode ligar ou desligar a segurança para determinados conjuntos de recursos, como comandos.

Verificação de nível de grupo de filas compartilhadas ou gerenciador de filas

É possível implementar a segurança no nível do gerenciador de filas ou no nível do grupo de filas compartilhadas. Se você implementar a segurança no nível do grupo de filas compartilhadas, todos os gerenciadores de filas no grupo compartilharão os mesmos perfis. Isso significa que existem menos perfis para definir e manter, facilitando o gerenciamento da segurança. Isso também facilita a inclusão de um novo gerenciador de filas no grupo de filas compartilhadas porque ele herda os perfis de segurança existentes.

Também será possível implementar uma combinação de ambos, se sua instalação precisar dela, por exemplo, durante a migração ou se você tiver um gerenciador de filas no grupo de filas compartilhadas que requer diferentes níveis de segurança para os outros gerenciadores de filas no grupo.

Segurança no nível do grupo de filas compartilhadas

A verificação de segurança no nível do grupo de filas compartilhadas é executada para o grupo de filas compartilhadas inteiro. Isso permite simplificar a administração de segurança porque requer que você defina menos perfis de segurança. A autorização de um ID do usuário para usar um recurso específico é manipulada no nível do grupo de filas compartilhadas e independe do gerenciador de filas que o ID do usuário está usando para acessar o recurso.

Por exemplo, imagine que um aplicativo do servidor seja executado sob o usuário ID SERVER e queira acessar uma fila denominada SERVER.REQUEST e você deseja executar uma instância de SERVER em cada imagem do z/OS no sysplex. Em vez de permitir que o SERVER abra o SERVER.REQUEST em cada gerenciador de filas individualmente (segurança de nível do gerenciador de filas), é possível permitir acesso apenas no nível do grupo de filas compartilhadas.

É possível usar os perfis de segurança de nível do grupo de filas compartilhadas para proteger todos os tipos de recursos, sejam locais ou compartilhados.

Segurança em Nível de Gerenciador de Fila

É possível usar os perfis de segurança de nível do gerenciador de filas para proteger todos os tipos de recursos, independente de se local ou compartilhado.

Combinação de Ambos os Níveis

É possível usar uma combinação de ambos, o gerenciador de filas e a segurança no nível do grupo de filas compartilhadas.

É possível substituir as configurações de segurança no nível do grupo de filas compartilhadas para um gerenciador de filas específico que é um membro desse grupo. Isso significa que você pode executar um nível diferente de verificações de segurança em um gerenciador de filas individual para os executados nos outros gerenciadores de fila no grupo.

Para obter mais informações, consulte [Perfis para controlar a segurança no nível do grupo de filas compartilhadas ou do gerenciador de filas](#).

Controlando o Número de IDs de Usuário Verificados

RESLEVEL é um perfil do Servidor de segurança que controla o número de IDs de usuário verificados para segurança do recurso do IBM MQ. Normalmente, quando um usuário tenta acessar um recurso do IBM MQ, o Servidor de segurança verifica o ID do usuário relevante ou os IDs para ver se o acesso é permitido para esse recurso. Ao definir um perfil do RESLEVEL, é possível controlar se zero, um ou, onde aplicável, dois IDs de usuário são verificados.

Esses controles são feitos em uma base de conexão por conexão e o último para a vida útil da conexão.

Existe apenas um perfil RESLEVEL para cada gerenciador de filas. O controle é implementado pelo acesso que um ID do usuário tem a esse perfil.

Classes do IBM MQ RACF com letras maiúsculas e minúsculas

Agora é possível usar o suporte de perfil do RACF com letras maiúsculas e minúsculas, que permite usar nomes de recurso compostos de maiúsculas e minúsculas e definir perfis do IBM MQ RACF para protegê-los.

É possível optar por:

- Continuar usando maiúsculas apenas em classes do IBM MQ RACF como nas liberações anteriores ou
- Utilizar classes do IBM MQ RACF com letras maiúsculas e minúsculas.

Sem o uso de perfis do RACF com maiúsculas e minúsculas, você ainda pode usar os nomes de recursos compostos por letras maiúsculas e minúsculas no IBM MQ for z/OS; no entanto, esses nomes de recursos podem ser protegidos apenas pelos perfis genéricos do RACF em classes do IBM MQ com letras maiúsculas. Ao usar o suporte de perfil do IBM MQ RACF com letras maiúsculas e minúsculas, é possível fornecer um nível mais granular de proteção definindo os perfis do IBM MQ RACF em classes do IBM MQ com letras maiúsculas e minúsculas.

Recursos que podem proteger no IBM MQ for z/OS

Quando um gerenciador de filas inicia ou quando instruído por um comando do operador, o IBM MQ for z/OS determina quais recursos você deseja proteger.

É possível controlar quais verificações de segurança são executadas para cada gerenciador de filas individual. Por exemplo, é possível implementar várias verificações de segurança em um gerenciador de filas de produção, mas nenhuma em um gerenciador de filas de teste.

Segurança de Conexão

A verificação de segurança de conexão é executada quando um programa de aplicativo tenta se conectar a um gerenciador de filas. Isso é feito por meio da emissão de uma solicitação MQCONN ou MQCONNX ou quando o inicializador de canais ou CICS ou adaptador do IMS, emite uma solicitação de conexão.

Se estiver usando a segurança no nível do gerenciador de filas, será possível desativar a verificação de segurança da conexão para um gerenciador de filas específico. No entanto, se você fizer isto, qualquer usuário poderá se conectar a esse gerenciador de filas.

Para o adaptador CICS, somente o ID do usuário do espaço de endereço do CICS é usado para a verificação de segurança da conexão, não o ID do usuário do terminal CICS individual. Para o adaptador IMS, quando as regiões de controle ou dependentes do IMS se conectam ao IBM MQ, o ID do usuário do espaço de endereço do IMS é verificado. Para o inicializador de canais, o ID do usuário usado pelo espaço de endereço do inicializador de canais é verificado.

É possível ligar ou desligar a verificação de segurança de conexão no nível do gerenciador de filas ou do grupo de filas compartilhadas.

Segurança de Comando

A verificação de segurança de comando é executada quando um usuário emite um comando MQSC a partir de qualquer uma das origens descritas em [Origens a partir das quais é possível emitir comandos MQSC e PCF em IBM MQ for z/OS](#). É possível fazer uma verificação separada no recurso especificado pelo comando conforme descrito em [“Segurança do Recurso de Comando” na página 284](#).

Se você desativar a verificação de comando, os emissores dos comandos não serão verificados para ver se eles possuem a autoridade para emitir o comando.

Se comandos MQSC forem inseridos a partir de um console, o console deverá ter o atributo de autoridade de console SYS do z/OS. Os comandos que são emitidos a partir dos conjuntos de dados CSQINP1 ou CSQINP2 ou internamente pelo gerenciador de filas, são isentos de toda verificação de segurança enquanto aqueles para CSQINPX usam o ID do usuário do espaço de endereço do inicializador de canais. Você deve controlar quem tem permissão para atualizar estes conjuntos de dados por meio da proteção de conjunto de dados normal.

É possível ativar ou desativar a verificação de segurança de comando no nível do grupo de filas compartilhadas ou do gerenciador de filas.

Segurança do Recurso de Comando

Alguns comandos MQSC, por exemplo a definição de uma fila local, envolvem a manipulação de recursos do IBM MQ. Quando a segurança do recurso de comando está ativa, toda vez que um comando que envolve um recurso é emitido, o IBM MQ verifica se o usuário tem permissão para alterar a definição desse recurso.

É possível usar a segurança do recurso de comando para ajudar a impingir normas de nomenclatura. Por exemplo, um administrador de folha de pagamento pode ter permissão para excluir e definir somente filas com nomes que iniciam com "PAYROLL". Se a segurança do recurso de comando estiver inativa, nenhuma verificação de segurança será feita no recurso que está sendo manipulado pelo comando. Não confunda segurança do recurso de comando com segurança de comando; as duas são independentes.

A desativação da verificação de segurança do recurso de comando não afeta a verificação de recursos que é feita especificamente para outros tipos de processamento que não envolvem comandos.

É possível ativar e desativar a verificação de segurança do recurso de comando no nível do gerenciador de filas ou do grupo de filas compartilhadas.

Considerações de Segurança do Canal

Segurança de canal

Quando você está usando canais, os recursos de segurança disponíveis dependem de qual protocolo de comunicação será usado. Se você usar TCP, não haverá recursos de segurança fornecidos com o protocolo de comunicação, embora seja possível usar o TLS. Se você estiver usando APPC, poderá fluir informações do ID do usuário do MCA de envio por meio da rede para o MCA de destino para verificação.

Para ambos os protocolos, é possível especificar quais IDs do usuário você deseja verificar para propósitos de segurança e quantos. Novamente, as opções disponíveis dependem de qual protocolo está sendo usado, do que você especificar quando definir o canal e das configurações de RESLEVEL para o inicializador de canais.

Para obter mais informações sobre os tipos de segurança de canal disponíveis, consulte [Registros de autenticação de canal](#) e [Visão geral da saída de segurança](#)

Referências relacionadas

“Segurança do recurso da API no IBM MQ for z/OS” na página 284

Os recursos são verificados quando um aplicativo abre um objeto com uma chamada MQOPEN ou MQPUT1. O acesso necessário para abrir um objeto depende de quais opções abertas são especificadas quando a fila é aberta.

Segurança do recurso da API no IBM MQ for z/OS

Os recursos são verificados quando um aplicativo abre um objeto com uma chamada MQOPEN ou MQPUT1. O acesso necessário para abrir um objeto depende de quais opções abertas são especificadas quando a fila é aberta.

A segurança de recurso de API é subdividida nas seguintes verificações:

- [Fila](#)
- [Processo](#)
- [Lista de nomes](#)
- [Usuário alternativo](#)
- [Contexto](#)

Nenhuma verificação de segurança é executada ao abrir o objeto do gerenciador de filas ou ao acessar os objetos da classe de armazenamento.

Fila

A verificação de segurança da fila controla quem tem permissão para abrir qual fila e quais opções possuem permissão para abrir com ela. Por exemplo, um usuário pode ter permissão para abrir uma fila denominada PAYROLL.INCREASE.SALARY para procurar as mensagens na fila (usando a opção MQOO_BROWSE), mas não remover as mensagens da fila (usando uma das opções MQOO_INPUT_*). Se você desligar a verificação para filas, qualquer usuário poderá abrir qualquer fila com qualquer opção aberta válida (ou seja, qualquer opção MQOO_* válida em uma chamada MQOPEN ou MQPUT1).

É possível ligar ou desligar a verificação de segurança da fila no nível do gerenciador de filas ou do grupo de filas compartilhadas.

Processo

A verificação de segurança do processo é executada quando um usuário abre um objeto de definição de processo. Se você desligar a verificação para os processos, qualquer usuário poderá abrir qualquer processo.

É possível ligar ou desligar a verificação de segurança do processo no nível do grupo de filas compartilhadas ou do gerenciador de filas.

Lista de Nomes

A verificação de segurança da lista de nomes é executada quando um usuário abre uma lista de nomes. Se você desligar a verificação para listas de nomes, qualquer usuário poderá abrir qualquer lista de nomes.

É possível ligar ou desligar a verificação de segurança da lista de nomes em qualquer nível do grupo de filas compartilhadas ou gerenciador de filas.

Alternar usuário

A segurança de usuário alternativa controla se um ID do usuário pode usar a autoridade de outro ID do usuário para abrir um objeto IBM MQ.

Por exemplo:

- Um programa do servidor que é executado sob o ID do usuário PAYSERV recupera uma mensagem de solicitação a partir de uma filha que foi colocada na fila pelo ID do usuário USER1.
- Quando o programa do servidor recebe a mensagem de solicitação, ele processa a solicitação e coloca a resposta de volta na fila de resposta especificada com a mensagem de solicitação.
- Em vez de usar seu próprio ID do usuário (PAYSERV) para autorizar a abertura da fila de resposta, o servidor pode especificar algum outro ID do usuário, neste caso, USER1. Neste exemplo, a segurança de usuário alternativo controla se o ID do usuário PAYSERV tem permissão para especificar o ID do usuário USER1 como um ID do usuário alternativo ao abrir a fila de resposta.

O ID do usuário alternativo é especificado no campo *AlternateUserId* do descritor de objeto (MQOD).

É possível usar os IDs de usuários alternativos em qualquer objeto do IBM MQ, por exemplo, processos ou listas de nomes. Isso não afeta o ID do usuário usado por qualquer outro gerenciador de recursos, por exemplo, para a segurança do CICS ou para a segurança do conjunto de dados do z/OS.

Se a segurança de usuário alternativo não estiver ativada, qualquer usuário poderá usar qualquer outro ID de usuário como um ID de usuário alternativo.

É possível ligar ou desligar a verificação de segurança de usuário alternativo no nível do grupo de filas compartilhadas ou do gerenciador de filas.

Context

O contexto são informações aplicáveis a uma determinada mensagem e está contido no descritor de mensagens (MQMD) que faz parte da mensagem. As informações de contexto são fornecidas em duas seções:

Seção de Identidade

O usuário do aplicativo que coloca primeiro a mensagem em uma fila. Consiste nos seguintes campos:

- *UserIdentifier*
- *AccountingToken*
- *ApplIdentityData*

Seção de Origem

O aplicativo que coloca a mensagem na fila em que está armazenada atualmente. Consiste nos seguintes campos:

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

Os aplicativos podem especificar os dados de contexto quando uma chamada MQPUT ou MQPUT1 é feita. O aplicativo pode gerar os dados, os dados podem ser passados adiante a partir de outra mensagem ou o gerenciador de filas pode gerar os dados, por padrão. Por exemplo, os programas do servidor podem usar os dados de contexto para verificar a identidade do solicitante ou seja, essa mensagem foi fornecida a partir do aplicativo correto? Geralmente, o campo *UserIdentifier* é usado para determinar o ID do usuário de um usuário alternativo.

Use a segurança de contexto para controlar se o usuário pode especificar qualquer uma das opções de contexto em qualquer chamada MQOPEN ou MQPUT. Para obter informações sobre as opções de contexto, consulte as opções [MQOPEN relacionadas ao contexto da mensagem](#). Para obter descrições dos campos do descritor de mensagens relacionados ao contexto, consulte [MQMD - Descritor de Mensagens MQMD - Descritor de Mensagens](#).

Se você desligar a verificação de segurança de contexto, qualquer usuário poderá usar qualquer opção de contexto que seja permitida pela segurança da fila.

É possível ativar ou desativar a verificação de segurança de contexto em qualquer nível do grupo de filas compartilhadas, gerenciador de filas ou fila.

z/OS

Disponibilidade no z/OS

IBM MQ for z/OS possui muitos recursos para alta disponibilidade. Este tópico descreve algumas das considerações para disponibilidade.

Diversos recursos do IBM MQ podem aumentar a disponibilidade do sistema se o gerenciador de filas ou o inicializador de canais falhar. Para obter informações adicionais sobre esses recursos, consulte as seguintes seções:

- [Considerações sobre o Sysplex](#)
- [Filas compartilhadas](#)
- [Canais compartilhados](#)
- [Disponibilidade de rede do IBM MQ](#)
- [Usando o Gerenciador de reinício automático \(ARM\) do z/OS](#)
- [Usando o Recurso de recuperação estendido \(XRF\) do z/OS](#)
- [Usando o atributo GROUPUR do z/OS para recuperação em um grupo de filas compartilhadas](#)

- Onde localizar mais informações sobre a disponibilidade

Considerações do Sysplex

Em um *sysplex*, diversas imagens do sistema operacional z/OS colaboram em uma imagem de sistema único e se comunicam usando o recurso de acoplamento. O IBM MQ pode usar os recursos do ambiente *sysplex* para disponibilidade aprimorada.

Remover as afinidades entre um gerenciador de filas e uma determinada imagem do z/OS permite que um gerenciador de filas seja reiniciado em uma imagem diferente do z/OS no caso de uma falha na imagem. O mecanismo de reinicialização pode ser manual, usar o ARM ou usar a automação do sistema, se você assegurar o seguinte:

- Todos os conjuntos de páginas, logs, conjuntos de dados de autoinicialização, bibliotecas de código e conjuntos de dados de configuração do gerenciador de filas são definidos nos volumes compartilhados.
- A definição do subsistema tem escopo *sysplex* e um nome exclusivo dentro do *sysplex*.
- O nível do *código anterior* instalado em cada imagem do z/OS no tempo IPL está no mesmo nível.
- Endereços IP virtuais TCP (VIPA) estão disponíveis em cada pilha TCP no *sysplex* e você configurou os listeners TCP do IBM MQ e conexões de entrada para usar VIPAs em vez de nomes de host padrão.

Para obter informações adicionais sobre o uso de TCP em um *sysplex*, consulte *TCP/IP em um sysplex*, SG24-5235, uma publicação IBM Redbooks.

Além disso, é possível configurar diversos gerenciadores de filas em execução em diferentes imagens do sistema operacional em um *sysplex* para operar como um grupo de filas compartilhadas, o que pode tirar proveito de filas compartilhadas e canais compartilhados para maior disponibilidade e balanceamento de carga de trabalho.

Filas compartilhadas

No ambiente do grupo de filas compartilhadas, um aplicativo pode se conectar a qualquer um dos gerenciadores de filas dentro do grupo de filas compartilhadas. Como todos os gerenciadores de filas no grupo de filas compartilhadas podem acessar o mesmo conjunto de filas compartilhadas, o aplicativo não depende da disponibilidade de um gerenciador de filas específico. Qualquer gerenciador de filas no grupo de filas compartilhadas pode atender a qualquer fila. Isso fornecerá uma maior disponibilidade se um gerenciador de filas parar porque todos os outros gerenciadores de filas no grupo de filas compartilhadas poderão continuar processando a fila. Para obter informações sobre alta disponibilidade de filas compartilhadas, consulte [“Vantagens de Usar Filas Compartilhadas”](#) na página 194.

Quando possível, para aprimorar ainda mais a disponibilidade de mensagens em um grupo de filas compartilhadas, o IBM MQ detectará se outro gerenciador de filas no grupo se desconectar anormalmente do recurso de acoplamento e concluirá as unidades de trabalho para esse gerenciador de filas que ainda estiverem pendentes. Isso é conhecido como *recuperação peer-to-peer* e é descrito em [“Recuperação Peer-to-peer”](#) na página 273.

A recuperação *peer-to-peer* não pode recuperar unidades de trabalho que estavam em dúvida no momento da falha. É possível usar Automatic Restart Manager (ARM) para reiniciar todos os sistemas envolvidos na falha (CICS, Db2 e IBM MQ por exemplo), e assegurar que todos sejam reiniciados no mesmo processador novo. Isso significa que eles podem resincronizar e fornece recuperação rápida de unidades de trabalho em dúvida. Isso é descrito no [“Usando o z/OS Automatic Restart Manager \(ARM\)”](#) na página 288.

Canais Compartilhados

No ambiente do grupo de filas compartilhadas, o IBM MQ fornece funções que fornecem alta disponibilidade para a rede. O inicializador de canais permite que você use produtos de rede que equilibrem as solicitações de rede em todo um conjunto de servidores elegíveis e oculte falhas do

servidor da rede (por exemplo, recursos genéricos VTAM). O IBM MQ usa uma porta genérica para solicitações de entrada para que as solicitações de conexão possam ser roteadas para qualquer inicializador de canais disponível no grupo de filas compartilhadas. Isso é descrito no [“Canais Compartilhados”](#) na página 215.

Os canais de saída compartilhada recebem as mensagens que enviam de uma fila de transmissão compartilhada. As informações sobre o status de um canal compartilhado são mantidas em um local para o nível do grupo de filas compartilhadas inteiro. Isso significa que um canal poderá ser reiniciado automaticamente em um inicializador de canais diferente no grupo de compartilhamento de fila se o inicializador de canais, o gerenciador de filas ou o subsistema de comunicações falhar. Isso é chamado de *recuperação de canal peer* e é descrito em [Canais de saída compartilhados](#).

Disponibilidade de rede do IBM MQ

As mensagens do IBM MQ são executadas a partir do gerenciador de filas para o gerenciador de filas em uma rede do IBM MQ usando canais. É possível alterar a configuração em inúmeros níveis para melhorar a disponibilidade da rede de um gerenciador de filas e a capacidade de um canal do IBM MQ detectar um problema de rede e se reconectar.

O TCP *Keepalive* está disponível para canais TCP/IP. Isso faz com que o TCP envie os pacotes periodicamente entre as sessões para detectar as falhas na rede. O atributo do canal KAINTE determina a frequência desses pacotes para um canal.

O *AdoptMCA* permite que um canal, bloqueado no processamento de recebimento como resultado de uma indisponibilidade na rede, seja finalizado e substituído por uma nova solicitação de conexão. Controle o *AdoptMCA* usando a propriedade do gerenciador de filas *ADOPTMCA* com o utilitário MQSC ou a propriedade *AdoptNewMCAType* com a interface de Formatos de Comando Programáveis.

O *ReceiveTimeout* evita que um canal seja bloqueado permanentemente em uma chamada de recepção de rede. Os parâmetros do inicializador de canais *RCVTIME* e *RCVTIME* determinam as características de tempo limite de recebimento para os canais, como uma função de seus intervalos de pulsação. Consulte [Parâmetro do gerenciador de filas](#) para obter mais detalhes.

Usando o z/OS Automatic Restart Manager (ARM)

É possível usar o IBM MQ for z/OS em conjunto com o gerenciador de reinício automático (ARM) do z/OS. Se um gerenciador de filas ou um inicializador de canais falhar, o ARM o reiniciará na mesma imagem do z/OS. Se o z/OS falhar, um grupo inteiro de subsistemas e aplicativos relacionados também falhará. ARM pode reiniciar todos os sistemas com falha automaticamente, em uma ordem predefinida, em outra imagem do z/OS no sysplex. Isso é denominado reinicialização de sistema cruzado.

O ARM permite a rápida recuperação de transações em dúvida no ambiente de fila compartilhada. Ele também fornece maior disponibilidade caso você não esteja usando grupos de filas compartilhadas.

É possível usar o ARM para reiniciar um gerenciador de filas em uma imagem diferente do z/OS dentro do sysplex no caso de uma falha do z/OS.

Para ativar a reinicialização automática, você deve fazer o seguinte:

1. Configure um conjunto de dados de acoplamento ARM.
2. Defina as ações de reinício automático que você deseja que o z/OS execute em uma *política de ARM*.
3. Inicie a política de ARM.

Se você deseja reiniciar os gerenciadores de fila em diferentes imagens do z/OS automaticamente, cada gerenciador de filas em cada imagem do z/OS na qual esse gerenciador de filas possa ser reiniciado deve ser definida com um nome de subsistema de 4 caracteres exclusivos com sysplex.

O uso do ARM com o IBM MQ é descrito em [Usando o ARM em uma rede do IBM MQ](#).

Usando o z/OS Extended Recovery Facility (XRF)

É possível usar o IBM MQ em um ambiente do recurso de recuperação estendido (XRF). Todos os conjuntos de dados de propriedade do IBM MQ (código executável, BSDSs, logs e conjuntos de páginas) devem estar no DASD compartilhado entre os processadores XRF alternativos e ativos.

Se você usar XRF para recuperação, deve parar o gerenciador de filas no processador ativo e iniciá-lo no processador alternativo. Para o CICS, é possível fazer isso usando a tabela de lista de comandos (CLT) fornecida pelo CICS, ou o operador do sistema pode fazê-lo manualmente. Para o IMS, essa é uma operação manual e você deve fazer isso depois que o sistema de coordenação do IMS concluiu o computador do processador.

Os utilitários do IBM MQ devem ser concluídos ou terminados antes que o gerenciador de filas possa ser alternado para o processador alternativo. Considere cuidadosamente o efeito dessa interrupção em potencial ao planejar seus planos de recuperação XRF.

Tome cuidado para evitar que o gerenciador de filas seja iniciado no processador alternativo antes que o gerenciador de filas no processador ativo seja finalizado. Um início prematuro pode causar problemas de integridade nos dados, catálogo e log. Usar a serialização de recurso global (GRS) ajuda a evitar problemas de integridade evitando o uso simultâneo do IBM MQ nos dois sistemas. Você deve incluir o BSDS como um recurso protegido e deve incluir os processadores XRF ativos e alternativos no anel GRS.

Usando o atributo GROUPUR do z/OS para recuperação em um grupo de filas compartilhadas

Os grupos de filas compartilhadas (QSG) permitem recursos transacionais adicionais que são descritos neste tópico. O atributo GROUPUR permite que os aplicativos clientes XA tenham qualquer recuperação de transação em dúvida, que possa ser necessária, executada qualquer membro do QSG.

Se um aplicativo cliente XA se conectar a um grupo de filas compartilhadas (QSG) por meio de um Sysplex, ele não poderá garantir a qual gerenciador de filas específico se conectará. O uso do atributo GROUPUR pelos gerenciadores de filas dentro do grupo de filas compartilhadas pode permitir qualquer recuperação de transação em dúvida que possa ser necessária para ocorrer em qualquer membro do QSG. Mesmo se o gerenciador de filas ao qual o aplicativo foi inicialmente conectado não estiver disponível, a recuperação de transação pode ocorrer.

Este recurso libera os aplicativos clientes XA de qualquer dependência em membros específicos do QSG e assim estende a disponibilidade do gerenciador de filas. O grupo de filas compartilhadas aparece para o aplicativo transacional como uma entidade única fornecendo todos os recursos do IBM MQ e sem um ponto de falha único do gerenciador de filas.

Esta funcionalidade não parece ser o aplicativo transacional.

Onde Localizar Mais Informações sobre a Disponibilidade

É possível localizar informações adicionais sobre esses tópicos a partir das seguintes origens:

Tópico	Onde Localizar
Grupos de compartilhamento de filas	“Filas compartilhadas e grupos de filas compartilhadas” na página 172
Parâmetros do sistema	Configurando parâmetros do sistema
Usando o Automatic Restart Manager Programas utilitário	Usando o ARM em uma rede do IBM MQ

Tabela 24. Onde Localizar Mais Informações sobre a Disponibilidade (continuação)

Tópico	Onde Localizar
Comandos MQSC	<u>Comandos do MQSC</u>

z/OS

Monitoramento e estatísticas no IBM MQ for z/OS

O IBM MQ for z/OS tem um conjunto de recursos para monitorar o gerenciador de filas e reunir estatísticas.

O IBM MQ fornece recursos para monitorar o sistema e coletar as estatísticas. Para obter informações adicionais sobre esses recursos, consulte as seguintes seções:

- [“Monitoramento on-line” na página 290](#)
- [“Rastreamento do IBM MQ” na página 290](#)
- [“Eventos” na página 291](#)

Monitoramento on-line

O IBM MQ inclui os seguintes comandos para monitorar o status dos objetos IBM MQ:

- DISPLAY CHSTATUS exibe o status de um canal especificado.
- DISPLAY QSTATUS exibe o status de uma fila especificada.
- DISPLAY CONN exibe o status de uma conexão especificada.

Para obter informações adicionais sobre esses comandos, consulte [Os Comandos MQSC](#).

Rastreamento do IBM MQ

O IBM MQ fornece um recurso de rastreamento que pode ser usado para reunir as seguintes informações enquanto o gerenciador de filas estiver em execução:

Estatísticas de Desempenho

O rastreamento de estatísticas reúne as seguintes informações para ajudá-lo a monitorar o desempenho e sintonizar o seu sistema:

- Contagens de diferentes solicitações MQI (estatísticas do gerenciador de mensagens)
- Contagens de solicitações de objeto diferentes (estatísticas do gerenciador de dados)
- Informações sobre o uso do Db2 (estatísticas do gerenciador Db2)
- Informações sobre o uso do Recurso de Acomodamento (estatísticas do gerenciador de Recurso de Acomodamento)
- Informações sobre o uso do SMDS (estatísticas do conjunto de dados de mensagens compartilhadas)
- Informações sobre o uso do buffer pool (estatísticas do gerenciador de buffers)
- Informações sobre a criação de log (estatísticas do gerenciador de log)
- Informações sobre o uso de armazenamento (estatísticas do gerenciador de armazenamento)
- Informações sobre as solicitações de bloqueio (estatísticas do gerenciador de bloqueios)

Dados de Contabilidade

- O rastreamento de contabilidade reúne informações sobre o tempo de processador gasto processando as chamadas MQI e sobre o número de solicitações MQPUT e MQGET feitas por um determinado usuário.

- IBM MQ também pode reunir as informações sobre cada tarefa usando o IBM MQ. Esses dados são reunidos como um registro de contabilidade de nível de encadeamento. Para cada encadeamento, o IBM MQ também reúne informações sobre cada fila usada por esse encadeamento.

Os dados gerados pelo rastreamento são enviados ao Recurso de Gerenciamento de Sistema (SMF) ou recurso de rastreamento generalizado (GTF).

Eventos

Os eventos do IBM MQ fornecem informações sobre erros, avisos e outras ocorrências significativas em um gerenciador de filas. Ao incorporar esses eventos em seu próprio aplicativo de gerenciamento de sistemas, é possível monitorar as atividades em vários gerenciadores de filas, para diversos aplicativos IBM MQ. Em específico, é possível monitorar todos os gerenciadores de fila em seu sistema a partir de um único gerenciador de filas.

Os eventos podem ser relatados por meio de um mecanismo de relatório gravado pelo usuário para um aplicativo de administração que suporte a apresentação dos eventos para um operador. Os eventos também permitem que os aplicativos atuem como agentes para outras redes de administração, por exemplo NetView, para monitorar os relatórios e criar os alertas apropriados.

Tarefas relacionadas

[Usando o rastreamento do IBM MQ](#)

[Usando eventos do IBM MQ](#)

z/OS

Disposição de unidade de recuperação no z/OS

Determinados aplicativos transacionais podem usar uma disposição da unidade de recuperação GROUP, em vez de um QMGR, quando conectado a um gerenciador de filas em um grupo de filas compartilhadas (QSG), especificando o nome do QSG ao invés do nome do gerenciador de filas ao se conectarem. Isso permite que a recuperação de transação seja mais flexível e robusta, removendo o requisito para se reconectar ao mesmo gerenciador de filas no QSG.

As transações iniciadas por aplicativos que foram conectados usando o nome do grupo de filas compartilhadas também possuem uma disposição de unidade de recuperação GROUP.

Quando um aplicativo transacional se conecta a uma disposição da unidade de recuperação GROUP, ele está logicamente conectado ao grupo de filas compartilhadas e não tem uma afinidade com nenhum gerenciador de filas específico. Qualquer transação 2-phase commit que tenha sido iniciada que tenha concluído a fase 1 do processo de confirmação ou seja, que esteja em dúvida, pode ser consultado e resolvido, quando conectado a qualquer gerenciador de filas no QSG. Em um cenário de recuperação isso significa que o coordenador de transação não precisa se reconectar ao mesmo gerenciador de filas, que pode estar indisponível nesse momento.

Os aplicativos que se conectam a uma disposição de unidade de recuperação QMGR possuem uma afinidade direta para o gerenciador de filas ao qual estão conectados. Em um cenário de recuperação, o coordenador da transação deve reconectar-se ao mesmo gerenciador de filas para resolver qualquer transação em dúvida, independentemente de se o gerenciador de filas pertence a um grupo de filas compartilhadas.

Quando os aplicativos especificam um nome de grupo de filas compartilhadas e, portanto, se conectam a um gerenciador de filas em um QSG com uma disposição de unidade de recuperação GROUP, o grupo de filas compartilhadas é logicamente um gerenciador de recursos separado. Isso significa que as transações em dúvida ficam visíveis apenas a um aplicativo se ele se reconectar à mesma disposição de unidade de recuperação. As transações em dúvida com uma disposição da unidade de recuperação QMGR não são visíveis para os aplicativos que foram conectados a uma disposição da unidade de recuperação GROUP e vice-versa.

Conceitos relacionados

[“Ativando as Unidades de Recuperação GROUP” na página 292](#)

Um grupo de filas compartilhadas pode configurar e ativar suporte para unidades de recuperação GROUP.

[“Suporte do Aplicativo” na página 293](#)

Use esta página para determinar quais aplicativos podem se conectar a uma disposição da unidade de recuperação GROUP.

Ativando as Unidades de Recuperação GROUP

Um grupo de filas compartilhadas pode configurar e ativar suporte para unidades de recuperação GROUP.

Para usar as unidades de recuperação GROUP em um gerenciador de filas em um QSG, ative o atributo do gerenciador de filas GROUPUR. Para obter mais informações sobre este conceito, consulte [“Disposição de unidade de recuperação no z/OS” na página 291](#) antes de ler o restante do tópico.

Quando o atributo GROUPUR do gerenciador de filas é ativado, o gerenciador de filas aceita novas conexões com uma disposição da unidade de recuperação GROUP. Se você desativar esse atributo, novas conexões com essa disposição não serão aceitas, embora os aplicativos já conectados não sejam afetados até serem desconectados.

Quando um aplicativo se conecta a uma disposição da unidade de recuperação GROUP e consulta quais transações estão em dúvida ou tenta resolver uma transação que foi iniciada em outro lugar no grupo de filas compartilhadas (QSG), o gerenciador de filas ao qual ele está agora conectado deve ser capaz de se comunicar com os outros membros do grupo de filas compartilhadas para que possa processar a solicitação. Para isso, usa uma fila compartilhada denominada SYSTEM.QSG.UR.RESOLUTION.QUEUE. Essa fila deve estar em uma estrutura do aplicativo recuperável denominada CSQSYSAPPL. A estrutura deve ser recuperável porque as mensagens persistentes são armazenadas nessa fila ao processar as solicitações de resolução.

Antes de poder ativar unidades de recuperação GROUP, você deve assegurar que a estrutura do recurso de acoplamento e a fila compartilhada estejam definidas. É possível usar as definições na amostra CSQ4INSS. Quando a fila é definida ou detectada durante a inicialização, cada gerenciador de filas no grupo de filas compartilhadas abre a fila para que possa receber solicitações recebidas. Se deseja excluir ou mover a fila porque foi definida incorretamente, será possível solicitar que os gerenciadores de fila fechem suas manipulações abertas, atualizando o objeto de fila para inibir as solicitações MQGET. Quando tiver efetuado as correções necessárias, permitindo que os aplicativos recebam as mensagens de uma fila mais uma vez direcionando cada gerenciador de filas para reabri-la. Use o comando DISPLAY QSTATUS para identificar quais manipulações estão abertas em uma fila.

Quando você tiver concluído essa configuração, será possível ativar unidades de recuperação GROUP em cada gerenciador de filas com os quais os aplicativos transacionais deverão estar aptos para se conectar com uma disposição da unidade de recuperação GROUP. Isso não precisa ser feito em todos os gerenciadores de filas dentro do grupo de filas compartilhadas, mas, se você optar por ativar essa funcionalidade apenas em um subconjunto do grupo de filas compartilhadas, deverá assegurar que seus aplicativos tentem apenas se conectar aos gerenciadores de filas nos quais você o ativou. Para obter informações adicionais, consulte [“Suporte do Aplicativo” na página 293](#).

Ao tentar ativar o atributo do gerenciador de filas GROUPUR, inúmeras verificações de configuração são executadas. O gerenciador de filas verifica se:

- Pertence a um grupo de filas compartilhadas.
- A fila compartilhada denominada SYSTEM.QSG.UR.RESOLUTION.QUEUE foi definida, de acordo com a definição em CSQ4INSS.
- O SYSTEM.QSG.UR.RESOLUTION.QUEUE está em uma estrutura CF recuperável denominada CSQSYSAPPL.

Se alguma das verificações acima falhar, o atributo GROUPUR permanecerá exibido e um código de mensagem será retornado.

Essas verificações de configuração também serão executadas na inicialização do gerenciador de filas, se o atributo do gerenciador de filas estiver ativado. Se alguma das verificações falhar durante a inicialização as unidades de recuperação GROUP de inicialização serão desativadas e o gerenciador de filas emitirá uma mensagem identificando qual verificação falhou. Quando tiver executado a ação corretiva necessário, você então deve reativar o atributo do gerenciador de filas.

Suporte do Aplicativo

Use esta página para determinar quais aplicativos podem se conectar a uma disposição da unidade de recuperação GROUP.

O suporte para a disposição da unidade de recuperação GROUP está limitado a certos tipos de aplicativos transacionais para os quais o IBM MQ for z/OS é um gerenciador de recursos, mas não o coordenador de transação. Os aplicativos transacionais atualmente suportados são:

- Aplicativos do cliente transacional estendido do IBM MQ
- Aplicativos IBM MQ classes for JMS em execução em um servidor de aplicativos, como WebSphere Application Server.
- CICS em execução no CICS Transaction Server 4.2 ou posterior, quando a definição de recurso MQCONN do CICS é configurada com RESYNCMEMBER (GROUPRESYNC).

Conceitos relacionados

[“Aplicativos do cliente transacional estendido do IBM MQ” na página 293](#)

Use esta página para determinar como os aplicativos clientes transacionais estendidos do IBM MQ podem usar a disposição da unidade de recuperação GROUP.

[“Aplicativos CICS” na página 293](#)

Use esta página para determinar como o CICS pode usar a unidade de disposição de recuperação GROUP.

Aplicativos do cliente transacional estendido do IBM MQ

Use esta página para determinar como os aplicativos clientes transacionais estendidos do IBM MQ podem usar a disposição da unidade de recuperação GROUP.

Um exemplo de um aplicativo cliente transacional estendido do IBM MQ é aquele que usa o JMS e é executado no WebSphere Application Server, se conectando ao IBM MQ sobre TCP/IP, em vez das ligações locais. Esses aplicativos clientes se conectam ao IBM MQ for z/OS sobre conexões de rede, como via TCP/IP. Para esses aplicativos, é o valor especificado para o parâmetro QMNAME da sequência xa_info passada na chamada xa_open que especifica se uma disposição da unidade de recuperação QMGR ou GROUP é usada. Para obter informações adicionais sobre xa_open, consulte [O Formato de uma Sequência xa_open](#) e [Processamento de Erros Adicionais para xa_open](#). Para aplicativos JMS, isso é feito especificando-se o nome do grupo de filas compartilhadas (QSG) no ConnectionFactory em vez do nome de um gerenciador de filas específico.

Para que os aplicativos clientes XA aproveitem o uso da disposição da unidade de recuperação GROUP, deve-se configurar sua configuração TCP/IP para permitir que seus aplicativos clientes sejam roteados para os gerenciadores de filas no grupo de filas compartilhadas que possuem o atributo GROUPUR ativado, em vez de um gerenciador de filas específico. Uma das tecnologias de endereço IP virtual dinâmico que pode ser usada para fazer isso é z/OS SysPlex Distributor. Consulte [z/OS Communications Server](#) e [z/OS Qualificações Básicas: Endereçamento virtual dinâmico](#) para obter detalhes adicionais. Se você deseja ativar as unidades de recuperação GROUP em um subconjunto dos gerenciadores de filas em seu grupo de filas compartilhadas, certifique-se de que seus aplicativos clientes não possam ser roteados para aqueles nos quais ele não está ativado.

Seus aplicativos clientes não precisam se conectar ao grupo de filas compartilhadas usando canais compartilhados.

Aplicativos CICS

Use esta página para determinar como o CICS pode usar a unidade de disposição de recuperação GROUP.

CICS 4.2 e posterior fornece a opção de ressincronização do grupo, RESYNCMEMBER (GROUPRESYNC) em uma definição de recurso MQCONN. Um CICS configurado com essa opção pode se conectar a qualquer gerenciador de filas adequado em um grupo de filas compartilhadas que está em execução na mesma LPAR da região do CICS. Para suportar a opção GROUPRESYNC do CICS, um gerenciador de filas deve estar em execução no MQ V7.1 ou mais recente, e ser ativado para o suporte de GROUPUR.

Transações em execução dentro de uma região CICS conectada ao MQ usando GROUPRESYNC criam unidades de trabalho com unidade de disposição de recuperação GROUP.

É possível usar RESYNCMEMBER (GROUPRESYNC) para permitir uma recuperação mais rápida após uma falha do gerenciador de filas, pois ela permite que a região CICS se conecte imediatamente a um gerenciador de filas elegível alternativo em execução na mesma LPAR, resolvendo quaisquer transações indeterminadas conforme necessário, sem aguardar a reinicialização do gerenciador de filas.

RESYNCMEMBER (GROUPRESYNC) também permite opções de reinicialização mais flexíveis para CICS. Uma região CICS com sua conexão MQ configurada para usar GROUPRESYNC e filas compartilhadas do MQ pode ser reiniciada em qualquer LPAR em que há um gerenciador de filas em execução como membro do mesmo grupo de filas compartilhadas.

z/OS IBM MQ e outros Produtos z/OS

Use este tópico para entender como o IBM MQ pode trabalhar com outros produtos z/OS.

Conceitos relacionados

[“IBM MQ e CICS” na página 294](#)

Todas as versões do CICS suportadas pelo IBM MQ 9.0.0 e mais recente usam a versão fornecida pelo CICS do adaptador e ponte.

[“IBM MQ for z/OS e WebSphere Application Server” na página 301](#)

Use este tópico para entender o uso do IBM MQ for z/OS pelo WebSphere Application Server.

Referências relacionadas

[“IBM MQ e IMS” na página 295](#)

Use este tópico para entender como o IBM MQ trabalha com IMS. O adaptador IMS permite que você conecte o gerenciador de filas ao IMS e permite que os aplicativos IMS usem o MQI.

[“Adaptadores IBM MQ e z/OS Batch, TSO e RRS” na página 299](#)

Use este tópico para entender como o IBM MQ trabalha com adaptadores z/OS Batch, TSO e RRS.

z/OS IBM MQ e CICS

Todas as versões do CICS suportadas pelo IBM MQ 9.0.0 e mais recente usam a versão fornecida pelo CICS do adaptador e ponte.

Para obter mais informações sobre como configurar o adaptador IBM MQ CICS e os componentes do IBM MQ CICS bridge, consulte a seção [Configurando conexões para IBM MQ](#) da documentação do CICS.

Tarefas relacionadas

[Usando o IBM MQ com o CICS](#)

z/OS Conexão do grupo CICS

A conexão do grupo CICS fornece a capacidade de uma região do CICS se conectar a qualquer membro ativo de um grupo de filas compartilhadas do IBM MQ na mesma LPAR em vez de especificar um gerenciador de filas individual. CICS ainda se conecta a um gerenciador de filas único por vez.

Você precisa de pelo menos dois gerenciadores de filas na LPAR para suporte à conexão do grupo CICS. A utilização da conexão do grupo fornece maior disponibilidade porque você não precisa que um gerenciador de filas específico esteja ativo. O CICS se conecta a qualquer gerenciador de filas no grupo de filas compartilhadas na LPAR.

Para obter mais informações, consulte a documentação do CICS no recurso MQCONN.

O CICS tenta se conectar ao MQNAME transmitido como se fosse um gerenciador de filas:

- Se o gerenciador de filas existir e estiver ativo, a conexão funcionará.
- Se a conexão falhar, o CICS consultará o status de gerenciadores de filas no grupo para determinar quais estão ativos na mesma LPAR.
- Se vários gerenciadores de filas estiverem ativos, o CICS verificará RESYNCMEMBER (YES) e o status da UOW para determinar se o CICS precisa se conectar ou deve se conectar a um determinado membro ou aguardar se não estiver ativo.

- Se não houver necessidade de se conectar a um membro específico, o CICS selecionará um gerenciador de filas (utilizando um algoritmo de randomização).
- O CICS tenta se conectar ao gerenciador de filas escolhido.
- Se a tentativa falhar, então, dependendo do código de retorno, o CICS escolherá o próximo membro e, em seguida, passará pelo loop de seleção novamente.
- Se não houver gerenciadores de filas ativos, o CICS emitirá conexões múltiplas na lista de gerenciadores de filas e aguardará na ECBLIST até que o primeiro gerenciador de filas se torne disponível.

Conceitos relacionados

[“Unidades de grupo de recuperação \(GROUPUR\) para CICS” na página 295](#)

O GROUPUR do IBM MQ para o CICS fornece recuperação peer-to-peer para unidades de trabalho em dúvida em um grupo de filas compartilhadas (QSG). Um gerenciador de filas do IBM MQ pode resolver unidades de trabalho indeterminadas em nome de outro gerenciador de filas no grupo de filas compartilhadas. Isso significa que se as reconexões do CICS por meio do grupo se conectarem a um gerenciador de filas diferente no QSG, ele pode resolver transações indeterminadas de uma conexão do IBM MQ anterior.

Informações relacionadas

[Suporte para grupos de filas compartilhadas do IBM MQ](#)

Unidades de grupo de recuperação (GROUPUR) para CICS

O GROUPUR do IBM MQ para o CICS fornece recuperação peer-to-peer para unidades de trabalho em dúvida em um grupo de filas compartilhadas (QSG). Um gerenciador de filas do IBM MQ pode resolver unidades de trabalho indeterminadas em nome de outro gerenciador de filas no grupo de filas compartilhadas. Isso significa que se as reconexões do CICS por meio do grupo se conectarem a um gerenciador de filas diferente no QSG, ele pode resolver transações indeterminadas de uma conexão do IBM MQ anterior.

Se uma região CICS estiver trabalhando com um gerenciador de filas, e o gerenciador de filas for encerrado de forma anormal, todas as transações pendentes serão recuperadas. Isso elimina a necessidade da região CICS esperar pelo gerenciador de filas que está trabalhando para reiniciar e resolve todas as unidades de trabalho pendentes. Isso significa que você precisa de pelo menos dois gerenciadores de filas na LPAR, de modo que o CICS possa se conectar a outro gerenciador de filas no caso de uma finalização anormal do gerenciador de filas primeiro.

A nova configuração RESYNCMEMBER (GROUPRESYNC) na definição de MQCONN CICS :

- Usa a função de conexão do grupo IBM MQ e a recuperação peer-to-peer.
- Requer um gerenciador de filas com o atributo GROUPUR ativado.
- Ainda suporta as configurações existentes do CICS MQCONN RESYNCMEMBER (YES e NO):
 - Usa a função de conexão do grupo CICS existente e nenhuma recuperação peer-to-peer.
 - A mudança das configurações de RESYNCMEMBER entra em vigor na próxima vez em que CICS se conectar a IBM MQ

Conceitos relacionados

[“Ativando as Unidades de Recuperação GROUP” na página 292](#)

Um grupo de filas compartilhadas pode configurar e ativar suporte para unidades de recuperação GROUP.

IBM MQ e IMS

Use este tópico para entender como o IBM MQ trabalha com IMS. O adaptador IMS permite que você conecte o gerenciador de filas ao IMS e permite que os aplicativos IMS usem o MQI.

O IBM MQ – IMS bridge adicional opcional permite que os aplicativos executem um aplicativo IMS que não use o MQI. Isso significa que é possível usar seus aplicativos legados com o IBM MQ, sem a necessidade de regravá-los.

Para obter informações adicionais sobre estes componentes, consulte os seguintes subtópicos:

Conceitos relacionados

[Aplicativos IMS e ponte IMS no IBM MQ for z/OS](#)

Tarefas relacionadas

[Configurando o adaptador do IMS](#)

[Configurando a Ponte IMS](#)

[Operando o adaptador IMS](#)

Referências relacionadas

[MQIIH - Informações de cabeçalho do IMS](#)

O adaptador IMS

O adaptador IMS é uma interface entre os programas de aplicativo IMS e um subsistema IBM MQ.

Os adaptadores IBM MQ permitem que diferentes ambientes de aplicativos enviem e recebam mensagens por meio de uma rede de enfileiramento de mensagens. O adaptador IMS é a interface entre programas de aplicativo IMS e um subsistema IBM MQ. Isso torna possível o uso de MQI pelos programas de aplicativo IMS.

O adaptador IMS recebe e interpreta solicitações para acesso ao IBM MQ usando o External Subsystem Attach Facility (ESAF) fornecido por IMS. Geralmente, o IMS se conecta ao IBM MQ automaticamente sem intervenção do operador.

O adaptador IMS fornece acesso aos recursos do IBM MQ para programas sendo executados nos seguintes modos ou estados:

- Modo de tarefa (TCB)
- Estado do problema
- Modo não de memória cruzada
- Modo não de registro de acesso

O adaptador fornece um encadeamento de conexão a partir de um bloco de controle de tarefas (TCB) para IBM MQ.

O adaptador suporta um protocolo two-phase commit para mudanças feitas aos recursos pertencentes ao IBM MQ com o IMS atuando como o coordenador de sincronização. Conversas em que o IMS não é o coordenador de sincronização, por exemplo, conversas protegidas por APPC (SYNCLVL=SYNCPT), não são suportadas pelo adaptador IMS.

O adaptador também fornece uma transação do monitor acionador (CSQQTRMN). Isso é descrito no [“O monitor acionador do IMS”](#) na página 297.

É possível usar o IBM MQ com o IMS Extended Recovery Facility (XRF) para ajudar na recuperação de um erro do IMS.

Nota: A partir do IMS 15.2 Extended Recovery Facility (XRF) não é mais suportado. Consulte a documentação [IMS](#) para obter mais informações.

Usando o Adaptador

Os programas de aplicativo e o adaptador IMS executam no mesmo espaço de endereço. O gerenciador de filas é separado, em seu próprio espaço de endereço.

Você deve linkeditar cada programa que emite uma ou mais chamadas MQI para um módulo de interface de linguagem IMS adequada e, a menos que use chamadas MQI dinâmicas, o programa stub de API fornecido pelo IBM MQ, CSQQSTUB. Quando o aplicativo emite uma chamada MQI, o stub transfere o controle para o adaptador por meio da interface de subsistema externa do IMS, que gerencia o processamento da solicitação pelo gerenciador de fila de mensagens.

Administração e operação do sistema com o IMS

Um operador de terminal IMS autorizado pode emitir comandos do IMS para controlar e monitorar a conexão com o IBM MQ. No entanto, o operador de terminal do IMS não tem controle sobre o espaço de endereço do IBM MQ. Por exemplo, o operador não pode encerrar o IBM MQ a partir de um espaço de endereço do IMS.

Restrições

As seguintes chamadas API do IBM MQ não são suportadas em um aplicativo que usa o adaptador do IMS:

- MQCB
- MQCB_FUNCTION
- MQCTL

O monitor acionador do IMS

O monitor acionador IMS (**CSQQTRMN**) é um aplicativo IBM MQ-fornecido IMS que inicia uma transação IMS quando ocorre um evento IBM MQ , por exemplo, quando uma mensagem é colocada em uma fila específica

Como Isso Funciona

Quando uma mensagem for colocada em uma fila de mensagens do aplicativo, um acionador será gerado se as condições acionadoras forem atendidas. O gerenciador de filas então grava uma mensagem (contendo alguns dados definidos pelo usuário), conhecida como uma *mensagem do acionador*, para a fila de inicialização que foi especificada para essa fila de mensagens. Em um ambiente do IMS, é possível iniciar uma instância do CSQQTRMN para monitorar uma fila de inicialização e recuperar as mensagens do acionador à medida que elas chegam. Geralmente, o CSQQTRMN planeja outra transação do IMS por um INSERT (ISRT) na fila de mensagens do IMS. O aplicativo IMS lê a mensagem a partir da fila de mensagens do aplicativo e, em seguida, a processa. O CSQQTRMN deve ser executado como um BMP de não mensagem.

Cada cópia do CSQQTRMN atende uma única fila de inicialização. Quando tiver iniciado, o monitor acionador é executado até que o IBM MQ ou IMS seja encerrado.

A macro APPLCTN para CSQQTRMN deve especificar SCHDTYP=PARALLEL.

Como o monitor acionador é um BMP orientado em lote, as transações do IMS que são iniciadas pelo monitor acionador contêm o seguinte:

- Espaços em branco no campo LTERM do IOPCB
- O nome PSB do monitor acionador BMP no campo ID do Usuário do IOPCB

Se a transação de destino do IMS estiver protegida pelo Servidor de segurança (anteriormente conhecido como RACF), talvez você precise definir CSQQTRMN como um ID de usuário para o Servidor de segurança.

O IBM MQ - IMS bridge

O IBM MQ – bridge é o componente do IMS que permite acesso direto de aplicativos IBM MQ for z/OS para aplicativos IBM MQ em seu sistema IMS.

O IBM MQ – IMS bridge permite o *suporte de MQI implícito*. Isso significa que você pode executar reengenharia de aplicativos legados que foram controlados pelos terminais conectados pelo 3270 para que sejam controlados pelas mensagens do IBM MQ, sem ter que regravá-los, recompilá-los ou revinculá-los. O aplicativo de ligação é um cliente IMS *Open Transaction Manager Access* (OTMA).

Nos aplicativos de ligação não existem chamadas do IBM MQ no aplicativo IMS. O aplicativo obtém sua entrada usando um GET UNIQUE (GU) no IOPCB e envia sua saída usando um ISRT para IOPCB. Os aplicativos IBM MQ usam o cabeçalho IMS (a estrutura MQIIH) nos dados de mensagem para assegurar que os aplicativos possam ser executados como faziam quando conduzidos por terminais não programáveis. Se você estiver usando um aplicativo IMS que processa mensagens de diversos segmentos, observe que todos os segmentos devem estar contidos em uma mensagem do IBM MQ.

O IMS bridge é ilustrado em [Figura 78](#) na página 298.

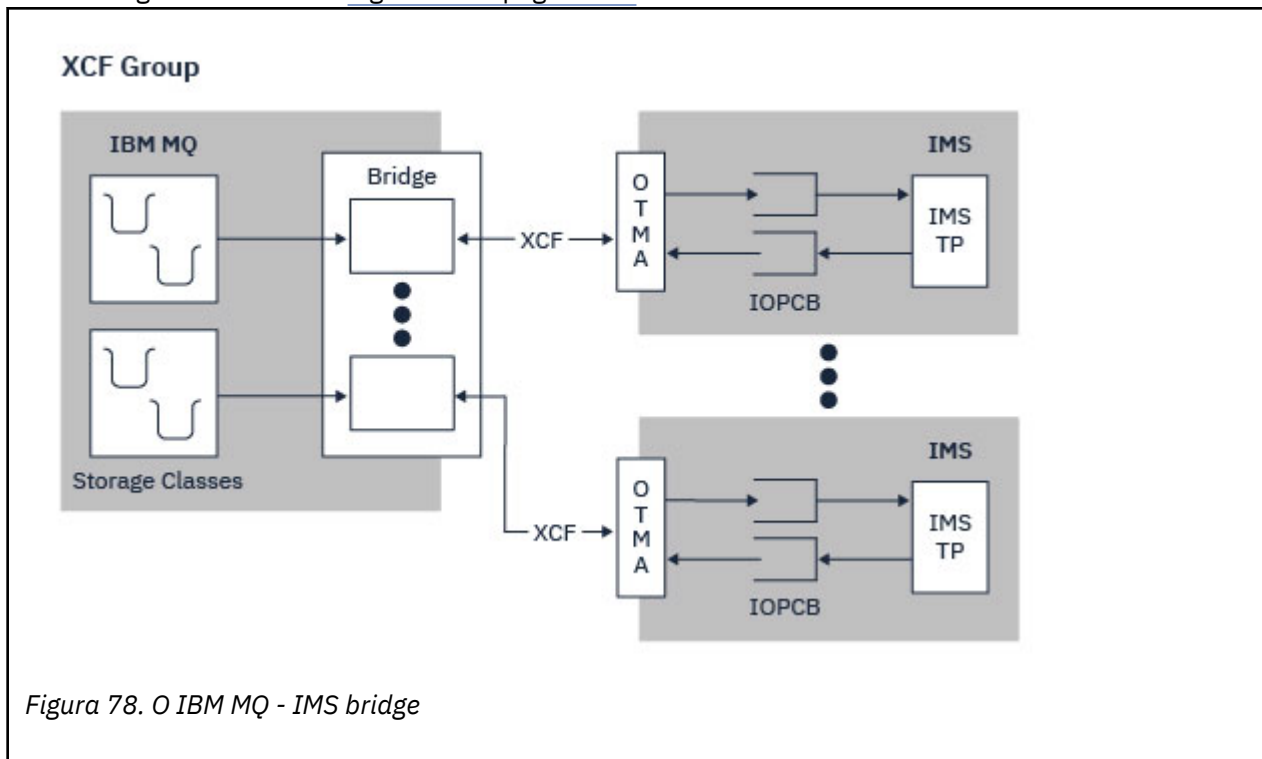


Figura 78. O IBM MQ - IMS bridge

Um gerenciador de filas pode se conectar a um ou mais sistemas IMS e mais de um gerenciador de filas pode se conectar a um sistema IMS. A única restrição é que todos devem pertencer ao mesmo grupo XCF e todos devem estar no mesmo sysplex.

Consulte [Configurando a ponte do IMS](#) para obter informações sobre como configurar uma ponte do IMS e incluir uma conexão do IMS adicional com o mesmo gerenciador de filas.

O que é o OTMA?

O recurso OTMA IMS é um protocolo cliente / servidor sem conexão baseado em transação que é executado no IMS. Ele funciona como uma interface para servidores de comunicações baseados em host que acessam aplicativos IMS TM por meio do [z/OS Cross Systems coupling facility \(XCF\)](#).

OTMA permite que os cliente se conectem ao IMS para fornecer alto desempenho para interações entre os clientes e o IMS para uma grande rede ou um grande número de sessões. O OTMA é implementado em um ambiente sysplex do z/OS. Portanto, o domínio de OTMA é restrito ao domínio de XCF.

Monitoramento de Recurso OTMA

O suporte para as mensagens do protocolo OTMA x '3C', disponível no IMS v10 ou superior, é incluído na ponte IBM MQ - IMS no IBM MQ for z/OS.. Essas mensagens são enviadas aos clientes OTMA pelo IMS para relatar seu status de funcionamento.

Se um parceiro IMS for incapaz de processar o volume de solicitações de transação que estão sendo enviadas, ele notificará o IBM MQ de que ocorreu um aviso de estouro. Em resposta, o IBM MQ diminuirá a taxa na qual as solicitações são enviadas sobre o aplicativo de ligação.

Se o IMS ainda não puder processar as solicitações de transação e uma condição de estouro integral ocorrer, todos os TPIPEs para o parceiro IMS serão suspensos. Mediante notificação do parceiro IMS de que a condição de estouro ou de aviso de estouro foi aliviada, o IBM MQ retomará todos os TPIPEs suspensos, se apropriado, e aumentará gradualmente a taxa na qual as solicitações de transação são enviadas até a taxa máxima ser atingida. As mensagens do console são emitidas pelo IBM MQ em resposta a uma mudança no status dos parceiros IMS.

Se parceiros do IMS v10 estiverem sendo usados, será necessário assegurar-se de que a PTF UK45082 tenha sido aplicada.

Enviando transações do IMS por meio do IBM MQ

Para enviar uma transação IMS que usa o aplicativo de ligação, os aplicativos colocam as mensagens em uma fila IBM MQ da maneira usual. As mensagens contêm dados de transação do IMS; elas podem ter um cabeçalho IMS (a estrutura MQIIH) ou permitir que o IBM MQ – IMS bridge faça suposições sobre os dados na mensagem.

IBM MQ, em seguida, coloca a mensagem em uma fila do IMS (ele é enfileirado no IBM MQ primeiro para permitir que o uso dos pontos de sincronização assegurem a integridade de dados). A classe de armazenamento da fila IBM MQ determina se a fila é uma *fila OTMA* (ou seja, uma fila usada para transmitir mensagens para o IBM MQ – IMS bridge) e o parceiro específicos do IMS para o qual os dados da mensagem são enviados.

Os gerenciadores de filas remotas também podem iniciar transações do IMS gravando nessas filas OTMA no IBM MQ for z/OS.

Os dados retornados do sistema IMS são gravados diretamente para a fila de resposta do IBM MQ especificada na estrutura do descritor de mensagens (MQMD). (Isso pode ser uma fila de transmissão para o gerenciador de filas especificado no campo **ReplyToQMGr** do MQMD.)

Conceitos relacionados

[Aplicativos IMS e ponte IMS no IBM MQ for z/OS](#)

Tarefas relacionadas

[Customizando a Ponte IMS](#)

Referências relacionadas

[“IBM MQ e IMS” na página 295](#)

Use este tópico para entender como o IBM MQ trabalha com IMS. O adaptador IMS permite que você conecte o gerenciador de filas ao IMS e permite que os aplicativos IMS usem o MQI.

z/OS

Adaptadores IBM MQ e z/OS Batch, TSO e RRS

Use este tópico para entender como o IBM MQ trabalha com adaptadores z/OS Batch, TSO e RRS.

introdução aos Adaptadores em Lote

Os adaptadores Batch/TSO são a interface entre os programas de aplicativos IBM MQ e z/OS em execução no JES, TSO ou z/OS UNIX System Services. Esses adaptadores ativam os programas de aplicativos do z/OS para usar o MQI.

Os adaptadores concedem acesso aos recursos do IBM MQ para programas sendo executados nos seguintes modos ou estados:

- Modo de tarefa (TCB)
- Problema ou estado do supervisor
- Modo não de memória cruzada
- Modo não de registro de acesso

Conexões entre os programas de aplicativo e o IBM MQ estão no nível de tarefa. Os adaptadores fornecem um encadeamento de conexão a partir de um bloco de controle de tarefa (TCB) do aplicativo para o IBM MQ.

O adaptador Batch/TSO suporta um protocolo single-phase commit para mudanças feitas nos recursos de propriedade do IBM MQ. Ele não suporta protocolos multi-phase commit. O adaptador RRS permite que os aplicativos IBM MQ participem dos protocolos two-phase commit com outros produtos ativos por RRS, coordenados por z/OS Resource Recovery Services (RRS).

Os adaptadores usam o serviço z/OS STIMERM para planejar um evento assíncrono a cada segundo. Esse evento executa um bloco de solicitação de interrupção (IRB) que não envolve qualquer espera pela tarefa do aplicativo em lote. Esse IRB verifica se o ECB de término do IBM MQ foi postado. Se o ECB de término tiver sido postado, o IRB postará quaisquer ECBs de aplicativo que estejam esperando um evento no IBM MQ (por exemplo, um sinal ou espera).

O Adaptador em Lote/TSO

O adaptador Batch/TSO do IBM MQ fornece o suporte do IBM MQ para aplicativos z/OS Batch e TSO. Todos os programas de aplicativo que são executados no z/OS Batch ou TSO devem ter o programa stub CSQBSTUB de API com link editado. O stub fornece ao aplicativo acesso a todas as chamadas MQI. Use o single-phase commit e a restauração para aplicativos, emitindo as chamadas MQI **MQCMIT** e **MQBACK**.

O Adaptador RRS

Serviços de Recuperação de Recursos (RRS) é um subcomponente do z/OS que fornece um serviço de abrangência do sistema para coordenar o two-phase commit nos produtos z/OS. O adaptador IBM MQ Batch/TSO RRS (o adaptador RRS) fornece o suporte do IBM MQ para aplicativos z/OS Batch e TSO que desejam usar esses serviços. O adaptador RRS permite que o IBM MQ se torne um participante integral na coordenação RRS. Os aplicativos podem participar em um processamento two-phase commit com outros produtos que suportam RRS (por exemplo, Db2).

O adaptador RRS fornece dois stubs; você deve editar o link dos programas de aplicativo que desejam usar o RRS com um desses stubs.

CSQBRSTB

Esse stub permite usar o two-phase commit e a restauração para aplicativos, usando os serviços de recuperação de recursos RRS de chamada em vez de chamadas MQI **MQCMIT** e **MQBACK**.

Você também deve linkeditar o módulo ATRSCSS a partir da biblioteca SYS1.CSSLIB com seu aplicativo. Se usar as chamadas MQI **MQCMIT** e **MQBACK**, receberá o código de retorno MQRC_ENVIRONMENT_ERROR.

CSQBRSI

Este stub permite usar as chamadas MQI **MQCMIT** e **MQBACK**; IBM MQ implementa de fato essas chamadas como chamadas RRS **SRRCMIT** e **SRRBACK**.

Para obter informações sobre como construir programas de aplicativos que usam o adaptador RRS, consulte O Adaptador em Lote RRS.

Onde localizar mais informações sobre os adaptadores z/OS Batch, TSO e RRS

É possível localizar informações adicionais sobre os tópicos nesta seção a partir das seguintes origens:

Tópico	Onde Localizar
Configurando os Adaptadores em Lote	<u>Tarefa 19: Configurar Lote, TSO e Adaptadores RRS</u>

Tabela 25. Onde localizar mais informações sobre o uso do z/OS Batch com o IBM MQ (continuação)

Tópico	Onde Localizar
Serviços de Recuperação de Recursos RRS de Chamada	<i>Programação MVS: Serviços de Chamada para Idiomas de Alto Nível</i>

z/OS IBM MQ for z/OS e WebSphere Application Server

Use este tópico para entender o uso do IBM MQ for z/OS pelo WebSphere Application Server.

Aplicativos gravados em Java que estão em execução em WebSphere Application Server podem usar a especificação Java Message Service (JMS) para executar o sistema de mensagens. O sistema de mensagens ponto a ponto neste ambiente pode ser fornecido por um gerenciador de filas do IBM MQ for z/OS.

Um benefício do uso de um gerenciador de filas do IBM MQ for z/OS para fornecer o sistema de mensagens é que a conexão de aplicativos JMS pode participar totalmente da funcionalidade de uma rede do IBM MQ. Por exemplo, eles podem usar a ponte IMS, ou trocar mensagens com gerenciadores de fila em execução em outras plataformas.

Conexão entre o WebSphere Application Server e um gerenciador de filas

Consulte [Usando IBM MQ e WebSphere Application Server juntos](#) para obter mais informações..

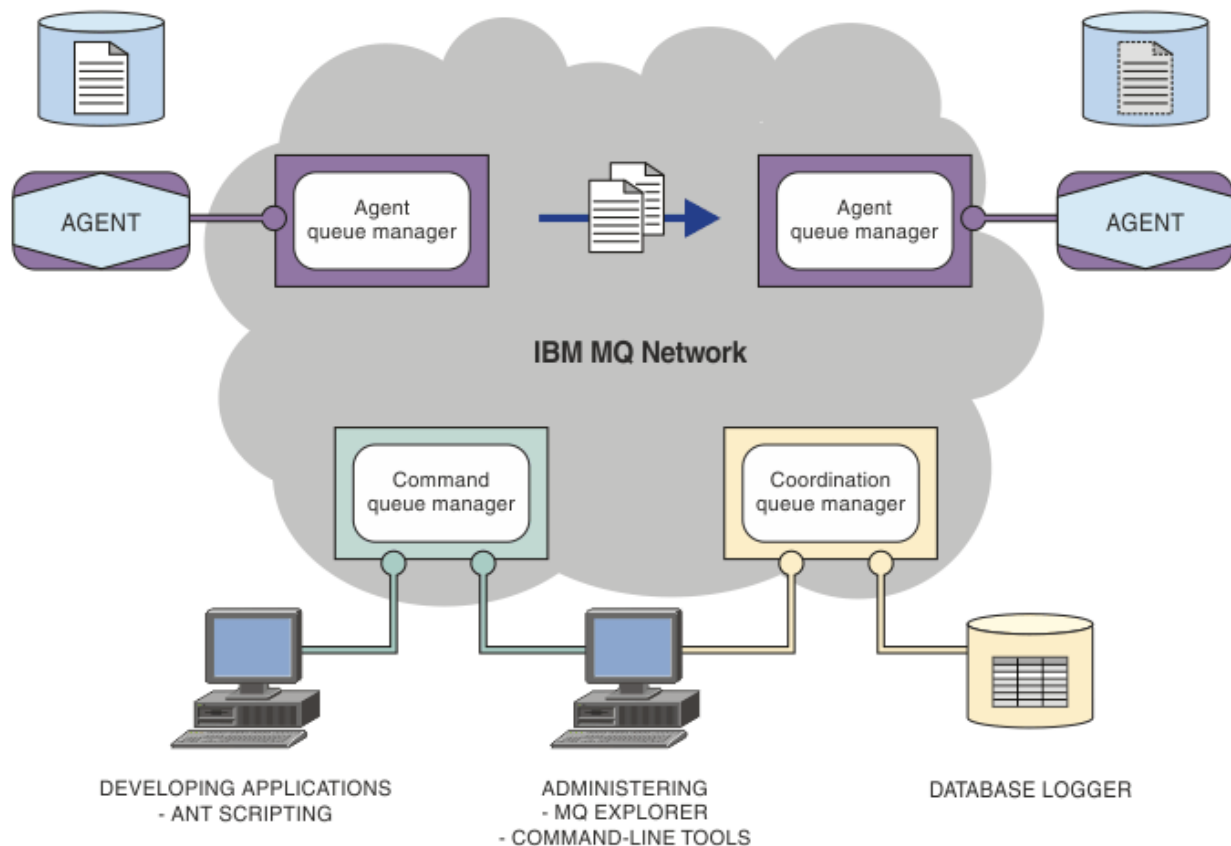
Usando funções do IBM MQ de aplicativos JMS

Por padrão, mensagens JMS retidas em filas do IBM MQ usam um cabeçalho MQRFH2 para manter algumas das informações do cabeçalho de mensagem de JMS. Muitos aplicativos legados do IBM MQ não podem processar mensagens com esses cabeçalhos e requerem seus próprios cabeçalhos característicos, por exemplo, os aplicativos MQCIH for CICS Bridge ou MQWIH for IBM MQ Workflow. Para obter mais detalhes sobre essas considerações especiais, consulte [Mapeando mensagens do JMS em mensagens do IBM MQ](#).

Managed File Transfer

O Managed File Transfer transfere arquivos entre sistemas de maneira gerenciada e auditável, independentemente do tamanho do arquivo ou do sistema operacional usado.

Você pode usar o Managed File Transfer para construir uma solução personalizada, escalável e automatizada que permite a você gerenciar, confiar e proteger transferências de arquivos. O Managed File Transfer elimina a redundância de custo, reduz os custos de manutenção e maximiza seus investimentos existentes de TI.





O diagrama mostra uma topologia simples do Managed File Transfer. Há dois agentes, cada um conecta-se a seu próprio gerenciador de filas do agente em uma rede do IBM MQ. Um arquivo é transferido do agente em um lado do diagrama, por meio da rede do IBM MQ, para o agente no outro lado do diagrama. Além disso, na rede do IBM MQ, estão o gerenciador de filas de coordenação e um gerenciador de filas de comando. Os aplicativos e as ferramentas se conectam a esses gerenciadores de filas para configurar, administrar, operar e registrar a atividade do Managed File Transfer na rede do IBM MQ.

O Managed File Transfer pode ser instalado como quatro opções diferentes, dependendo do seu sistema operacional e da configuração geral. Essas opções são Managed File Transfer Agent, Managed File Transfer Logger, Managed File Transfer Service ou Managed File Transfer Tools. Para obter mais informações, consulte [Opções do produto Managed File Transfer](#).

Você pode usar o Managed File Transfer para executar as seguintes tarefas:

- Criar transferências de arquivos gerenciados
 - Windows Linux Criar novas transferências de arquivos do IBM MQ Explorer nas plataformas Linux ou Windows.
 - Criar novas transferências de arquivos da linha de comandos em todas as plataformas suportadas.
 - Integrar a função de transferência de arquivos à ferramenta do Apache Ant.
 - Gravar aplicativos que controlam o Managed File Transfer colocando mensagens nas filas de comando do agente.
 - Programar transferência de arquivos para ocorrer em momento posterior. Também é possível acionar as transferências de arquivos planejadas com base em um intervalo de eventos do sistema de arquivos, por exemplo, um novo arquivo que está sendo criado.
 - Monitorar continuamente um recurso, por exemplo, um diretório e iniciar uma tarefa quando o conteúdo desse recurso atender alguma condição predefinida. Essa tarefa pode ser uma transferência de arquivos, um script do Ant ou uma tarefa de JCL.

- Transferir arquivos para e de filas do IBM MQ.
- Transferir arquivos para e a partir de servidores FTP, FTPS ou SFTP.
- Transferir arquivos para e de nós Connect:Direct.
- Transferir ambos, arquivos de texto e binários. Os arquivos texto são convertidos automaticamente entre páginas de código e convenções de final-de-linha dos sistemas de origem e destino.
- As transferências podem ser protegidas, usando os padrões de mercado para conexões com base em Secure Socket Layer (SSL).
- Visualizar transferências em andamento e informação de registro sobre todas as transferências em sua rede.
 -  Visualizar o status de transferências em andamento do IBM MQ Explorer em plataformas Linux ou Windows.
 -  Verificar o status das transferências concluídas usando o IBM MQ Explorer nas plataformas Linux ou Windows.
 - Usar o recurso de criador de logs de banco de dados do Managed File Transfer para salvar mensagens de log para um banco de dados Db2 ou Oracle.

O Managed File Transfer é construído em IBM MQ, o que oferece entrega segura e única de mensagens entre aplicativos. Você pode tirar vantagem de vários recursos do IBM MQ. Por exemplo, você pode usar a compactação de canal para compactar os dados que envia entre agentes via canais do IBM MQ e usar os canais SSL para proteger os dados que você envia entre agentes. Os arquivos são transferidos de modo confiável e podem tolerar a falha da infra-estrutura por meio da qual a transferência será executada. Se tiver uma indisponibilidade da rede, a transferência de arquivos reinicia de onde ela parou, quando a conectividade é restabelecida.

Por consolidar a transferência de arquivos com sua rede IBM MQ existente, você pode evitar o gasto de recursos necessários para manter as duas infraestruturas separadas. Se você ainda não for um cliente do IBM MQ, ao criar uma rede do IBM MQ para suportar o Managed File Transfer, você construirá o backbone de uma implementação futura do SOA. Se você já for um cliente do IBM MQ, o Managed File Transfer poderá aproveitar sua infra-estrutura existente do IBM MQ, incluindo IBM MQ Internet Pass-Thru e IBM Integration Bus.

É possível aproveitar as soluções de alta disponibilidade do IBM MQ para melhorar a resiliência de sua configuração do Managed File Transfer. Se seus agentes usarem gerenciadores de filas de dados replicados (RDQMs), eles deverão ser configurados para usar o recurso de endereço IP flutuante. Isso significa que os agentes usam o mesmo endereço IP para se comunicar com qualquer uma das três instâncias do RDQM atualmente em execução e se reconectam automaticamente no failover (consulte [Alta disponibilidade do RDQM e Criando e excluindo um endereço IP flutuante](#)). Se você usar a solução do gerenciador de filas de várias instâncias, os aplicativos usarão um endereço IP diferente para se comunicar com cada instância, que é manipulada pela reconexão do cliente no failover (consulte [Gerenciadores de filas de várias instâncias e Canal e reconexão do cliente](#)).

O Managed File Transfer integra-se com vários outros produtos IBM:

IBM Integration Bus

Arquivos de processo que foram transferidos pelo Managed File Transfer como parte de um fluxo do IBM Integration Bus. Para obter mais informações, consulte [Trabalhando com MFT por meio do IBM Integration Bus](#).

IBM Sterling Connect:Direct

Transferir arquivos para e de uma rede Connect:Direct existente utilizando a ponte Managed File Transfer Connect:Direct. Para obter mais informações, consulte [A ponte Connect:Direct](#).

IBM Tivoli Composite Application Manager

O IBM Tivoli Composite Application Manager fornece um agente que você pode usar para monitorar informações que são publicadas para o gerenciador de filas de coordenação.

Conceitos relacionados

Opções do produto Managed File Transfer

[“Visão Geral da Topologia do MFT” na página 304](#)

Uma visão geral de como os agentes do Managed File Transfer são conectados ao gerenciador de filas de coordenação em uma rede do IBM MQ.

[“Como o MFT trabalha com o IBM MQ?” na página 304](#)

O Managed File Transfer interage de várias maneiras com o IBM MQ.

Como o MFT trabalha com o IBM MQ?

O Managed File Transfer interage de várias maneiras com o IBM MQ.

- O Managed File Transfer transfere arquivos entre os processos do agente, dividindo cada arquivo em uma ou mais mensagens e transmitindo as mensagens por meio da rede do IBM MQ.
- Os processos do agente movem os dados do arquivo usando mensagens não persistentes para minimizar o impacto sobre os logs do IBM MQ. Por se comunicarem entre si, os processos do agente regulam o fluxo de mensagens quem contêm dados de arquivo. Isso evita que mensagens contendo dados do arquivo se acumulem em filas de transmissão do IBM MQ e assegura que, se algumas das mensagens não persistentes não forem entregues, os dados do arquivo sejam enviados novamente.
- Os agentes do Managed File Transfer usam várias filas do IBM MQ. Para obter mais informações, consulte as [MFTfilas do sistema e o tópico do sistema](#).
- Embora algumas destas filas sejam estritamente para uso interno, um agente pode aceitar trabalho na forma de mensagens de comando especialmente formatadas, enviadas para uma fila específica da qual o agente pode ler. Os comandos da linha de comandos e o plug-in do IBM MQ Explorer enviam mensagens do IBM MQ para o agente para instruí-lo a executar a ação desejada. É possível gravar aplicativos IBM MQ que interagem com o agente dessa maneira. Para obter mais informações, consulte [Controlando o MFT ao colocar mensagens na fila de comandos do agente](#).
- Os agentes do Managed File Transfer enviam informações sobre seu estado e o progresso e resultado de transferências para um gerenciador de filas do MQ que foi designado como o gerenciador de filas de coordenação. Essas informações são publicadas pelo gerenciador de filas de coordenação e podem ser assinadas pelos aplicativos que desejam monitorar o progresso da transferência ou manter registros das transferências ocorridas. Os comandos da linha de comandos e o plug-in do IBM MQ Explorer podem usar as informações publicadas. É possível gravar aplicativos do IBM MQ que usam essas informações. Para obter mais informações sobre o tópico para o qual as informações são publicadas, consulte [SYSTEM.FTE tópico FTE](#)
- Os componentes chave do Managed File Transfer tiram vantagem do recurso de gerenciadores de filas do IBM MQ para armazenar e encaminhar mensagens. Isso significa que, em face de uma interrupção, as partes não afetadas da infra-estrutura podem continuar a transferir arquivos. Isso estende o gerenciador de filas de coordenação, em que uma combinação de armazenamento e encaminhamento e assinaturas duráveis permitem que o gerenciador de filas de coordenação seja capaz de ficar indisponível sem perder as principais informações sobre as transferências de arquivos realizadas.

Visão Geral da Topologia do MFT

Uma visão geral de como os agentes do Managed File Transfer são conectados ao gerenciador de filas de coordenação em uma rede do IBM MQ.

Os agentes do Managed File Transfer enviam e recebem os arquivos que são transferidos. Cada agente possui seu próprio conjunto de filas em seu gerenciador de filas associado e o agente conecta-se a esse gerenciador no modo de ligações ou cliente. Um agente também pode utilizar o gerenciador de filas de coordenação como seu gerenciador de filas.

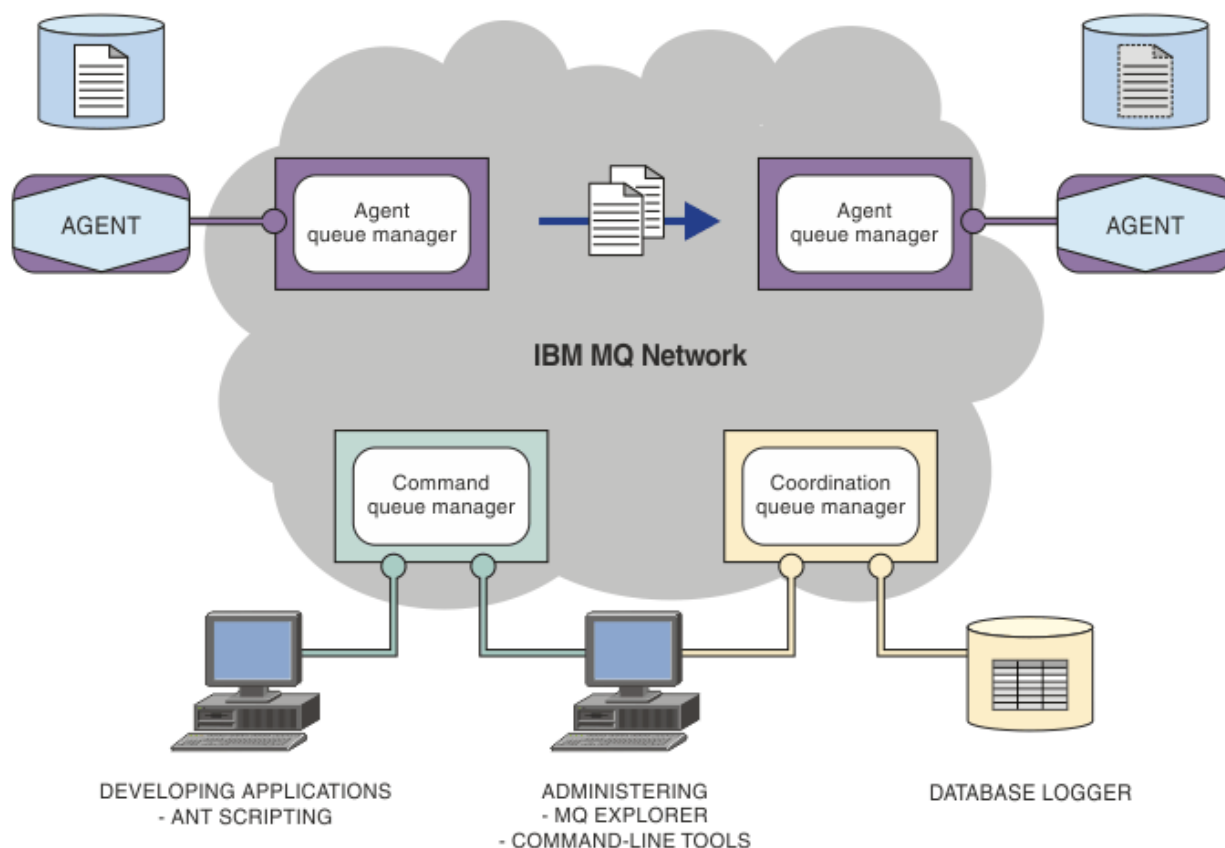
O gerenciador de filas de coordenação divulga informações de auditoria e transferência de arquivos. O gerenciador de filas de coordenação representa um único ponto para a coleta de informações do agente, de status de transferência e de auditoria de transferência. O gerenciador de filas de coordenação não precisa estar disponível para que as transferências ocorram. Se o gerenciador de filas de coordenação ficar temporariamente indisponível, as transferências continuarão normalmente. Mensagens de auditoria

e de status são armazenadas no gerenciador de filas do agente até que o gerenciador de filas de coordenação fique disponível e pode, então, ser processadas normalmente.

Os agentes registram-se com o gerenciador de filas de coordenação e publicam seus detalhes para esse gerenciador de filas. Estas informações do agente são usadas pelo plug-in do Managed File Transfer para ativar o início das transferências do IBM MQ Explorer. As informações do agente coletadas no gerenciador de filas de coordenação também são usadas pelos comandos para exibir as informações do agente e o status do agente.

As informações de auditoria de transferência e de status de transferência são publicadas no gerenciador de filas de coordenação. As informações do status de transferência e da auditoria de transferência são usadas pelo plug-in do Managed File Transfer para monitorar o progresso das transferências do IBM MQ Explorer. As informações de auditoria de transferência armazenadas no gerenciador de filas de coordenação podem ficar retidas para fornecer capacidade de auditoria.

O gerenciador de filas de comando é usado para se conectar à rede do IBM MQ e é o gerenciador de filas ao qual você se conecta ao emitir comandos do Managed File Transfer.



Conceitos relacionados

[“Managed File Transfer” na página 301](#)

O Managed File Transfer transfere arquivos entre sistemas de maneira gerenciada e auditável, independentemente do tamanho do arquivo ou do sistema operacional usado.

[“Como o MFT trabalha com o IBM MQ?” na página 304](#)

O Managed File Transfer interage de várias maneiras com o IBM MQ.

[Cenário do Managed File Transfer](#)

Visão Geral do MFT REST API

A REST API suporta determinados comandos do Managed File Transfer, incluindo a listagem de transferências e os detalhes sobre os agentes de transferência de arquivos.

No IBM MQ 9.1.0, a REST API inclui opções para listar todas as transferências atuais do Managed File Transfer e para consultar o status de agentes do Managed File Transfer. Para obter informações adicionais, consulte [Introdução ao REST API MFT](#).

IBM MQ Internet Pass-Thru

O IBM MQ Internet Pass-Thru (MQIPT) é um componente opcional do IBM MQ que pode ser usado para implementar soluções do sistema de mensagens entre sites remotos através da Internet.

A partir do IBM MQ 9.2.0, o MQIPT é um componente opcional do IBM MQ. Para obter os arquivos de instalação do MQIPT para o IBM MQ 9.3.x, acesse o [IBM Fix Central for IBM MQ](#). Antes do IBM MQ 9.2.0, o MQIPT estava disponível como um pacote de suporte.

Não é necessário executar IBM MQ 9.3 para usar MQIPT em IBM MQ 9.3. É possível usar o MQIPT para conectar qualquer versão suportada do IBM MQ e não é necessário instalar nenhum outro componente do IBM MQ na mesma versão do MQIPT.

Se você comprou a autorização do IBM MQ, será possível instalar quantas cópias forem necessárias do MQIPT. As instalações do MQIPT não são contadas com relação à sua autorização adquirida do IBM MQ. Para obter mais informações sobre o licenciamento do IBM MQ, consulte [Informações sobre licença do IBM MQ](#).

Nota: Esta documentação se relaciona com MQIPT em IBM MQ 9.3 Para a documentação do pacote de suporte do MQIPT (versão 2.1) no IBM Documentation, consulte [MQIPT \(SupportPac MS81\)](#) na documentação do IBM MQ 9.0.

Nota: Se você estiver usando MQIPT 2.1 ou anterior, será recomendado fazer upgrade para o MQIPT for IBM MQ 9.3, pois a data de término do suporte para o pacote de suporte MQIPT foi 30th de setembro de 2020.

O IBM MQ Internet Pass-Thru é executado como um serviço independente que pode receber e encaminhar fluxos de mensagens do IBM MQ, entre dois gerenciadores de filas do IBM MQ ou entre um cliente do IBM MQ e um gerenciador de filas do IBM MQ.

MQIPT ativa essa conexão quando o cliente e o servidor não estão na mesma rede física.

Uma ou mais instâncias do MQIPT podem ser colocadas no caminho de comunicação entre dois gerenciadores de filas IBM MQ, ou entre um cliente IBM MQ e um gerenciador de filas IBM MQ. As instâncias do MQIPT permitem que os dois sistemas IBM MQ troquem mensagens sem a necessidade de uma conexão TCP/IP direta entre eles. Isso é útil se a configuração de firewall proibir uma conexão TCP/IP direta entre os dois sistemas.

MQIPT atende em uma ou mais portas TCP/IP para conexões de entrada, que podem transportar mensagens normais do IBM MQ, IBM MQ mensagens ajustadas dentro de HTTP ou mensagens criptografadas usando Segurança da Camada de Transporte (TLS) ou Secure Sockets Layer (SSL). MQIPT pode manipular várias conexões simultâneas.

O canal IBM MQ que faz a solicitação de conexão TCP/IP inicial é chamado de *responsável pela chamada*, o canal ao qual ele está tentando se conectar de *respondente* e o gerenciador de filas com quem ele está finalmente tentando entrar em contato de *gerenciador de filas de destino*.

MQIPT retém dados na memória conforme ele a encaminha de sua origem para seu destino. Nenhum dado é salvo no disco (exceto para a memória paginada pelo disco pelo sistema operacional). O único horário MQIPT acessa o disco explicitamente é para ler seu arquivo de configuração e gravar registros de log de conexão e de rastreamento.

O intervalo completo de tipos de canais IBM MQ pode ser feito por meio de uma ou mais instâncias de MQIPT. A presença de MQIPT em um caminho de comunicação não tem efeito sobre as características funcionais dos componentes IBM MQ conectados, mas pode haver algum efeito no desempenho da transferência de mensagem.

MQIPT pode ser usado em conjunto com IBM MQ e IBM Integration Bus, conforme descrito em [“Possíveis configurações do MQIPT”](#) na página 310.

Para instalar o MQIPT, consulte [Instalando o MQIPT](#).

Tarefas relacionadas

[Configurando o IBM MQ Internet Pass-Thru](#)

[Administrando e configurando IBM MQ Internet Pass-Thru](#)

Referências relacionadas

[IBM MQ Internet Pass-Thru Referência de configuração](#)

Usos do MQIPT

Há vários usos potenciais para o IBM MQ Internet Pass-Thru (MQIPT).

O MQIPT pode ser usado como um concentrador de canal

Ao usar MQIPT desta maneira, os canais para ou a partir de vários hosts separados podem aparecer para um firewall como se eles fossem todos para ou a partir do host MQIPT. Isso torna mais fácil definir e gerenciar regras de filtragem de firewall.

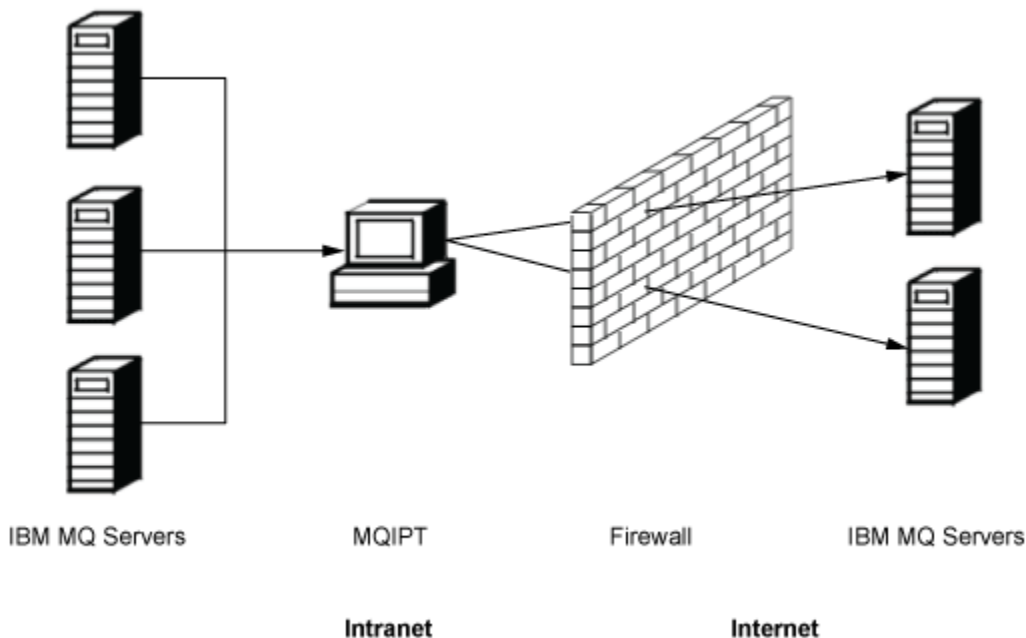


Figura 79. Exemplo de MQIPT como um concentrador de canal

MQIPT pode ser colocado em uma DMZ para fornecer um único ponto de acesso

Se MQIPT for colocado dentro de um firewall DMZ (uma configuração de firewall para proteção de redes locais), em um computador com um endereço de protocolo de Internet (IP) conhecido e confiável, MQIPT pode ser usado para atender as conexões de canal IBM MQ recebidas que ele pode, então, encaminhar para a intranet confiável; o firewall interno deve permitir que esse computador confiável faça conexões de entrada. Nessa configuração, o MQIPT evita que solicitações externas para acesso recebam os endereços IP verdadeiros dos computadores na intranet confiável. Desta maneira, MQIPT fornece um único ponto de acesso. Se necessário, o MQIPT pode ser configurado para aceitar conexões TLS e encaminhar dados para o destino usando uma conexão TLS separada, finalizando, portanto, a sessão TLS na DMZ

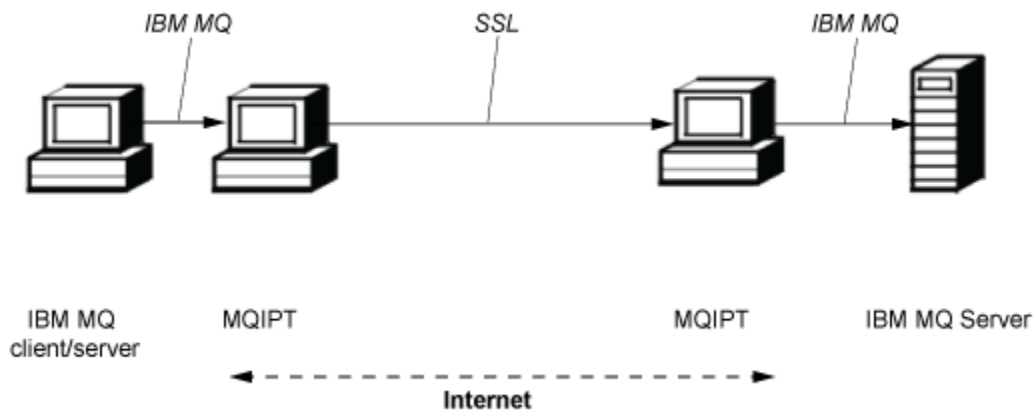


Figura 82. Exemplo de MQIPT e SSL/TLS

Como o MQIPT funciona

Em sua configuração mais simples, MQIPT age como um encaminhador de protocolo IBM MQ. Ele atende em uma porta TCP/IP e aceita solicitações de conexão de canais IBM MQ.

Se uma solicitação bem-formada for recebida, MQIPT estabelecerá uma conexão TCP/IP adicional entre si mesmo e o gerenciador de filas IBM MQ de destino. Em seguida, ele transmite todos os pacotes de protocolo que recebe de sua conexão de entrada para o gerenciador de filas de destino e retorna pacotes de protocolo do gerenciador de filas de destino de volta para a conexão de entrada original.

Nenhuma mudança para o protocolo IBM MQ (cliente / servidor ou gerenciador de filas para o gerenciador de filas) está envolvida porque nenhuma das finalizações está diretamente ciente da presença do intermediário. Novas versões do cliente IBM MQ ou código do servidor não são necessárias.

Para usar MQIPT, o canal responsável pela chamada deve ser configurado para usar o nome do host MQIPT e a porta, não o nome do host e a porta do gerenciador de filas de destino. Isso é definido com a propriedade **CONNNAME** do canal IBM MQ. MQIPT lê os dados recebidos e simplesmente transmite-o para o gerenciador de filas de destino. Outros campos de configuração, como o ID do usuário e a senha em um canal de cliente / servidor, são transmitidos de forma semelhante ao gerenciador de filas de destino.

Vários gerenciadores de filas

MQIPT pode ser usado para permitir o acesso a mais de um gerenciador de filas de destino. Para que isso funcione, deve haver um mecanismo para informar ao MQIPT a qual gerenciador de filas se conectar, portanto, o MQIPT usa o número da porta TCP/IP recebida para determinar a qual gerenciador de filas se conectar.

Portanto, é possível configurar MQIPT para atender em várias portas TCP/IP. Cada porta de atendimento é mapeada para um gerenciador de filas de destino por meio de uma *MQIPTrota*. É possível definir até 100 dessas rotas, que associam uma porta TCP/IP de atendimento com o nome do host e a porta do gerenciador de filas de destino. Isso significa que o nome do host (endereço IP) do gerenciador de filas de destino nunca é visível para o canal de origem. Cada rota pode manipular várias conexões entre sua porta de atendimento e de destino, cada conexão agindo independentemente.

Arquivo de configuração do MQIPT

O MQIPT usa um arquivo de configuração chamado `mqipt.conf`. Este arquivo contém definições de todas as rotas e suas propriedades associadas. Veja [Administrando e configurando o IBM MQ Internet Pass-Thru](#) para obter mais informações sobre o `mqipt.conf`.

Quando MQIPT for ativado, ele iniciará cada rota que está listada no arquivo de configuração. As mensagens são gravadas no console do sistema mostrando o status de cada rota. Quando a mensagem MQCPI078 é mostrada para uma rota, essa rota está pronta para aceitar solicitações de conexão.

Possíveis configurações do MQIPT

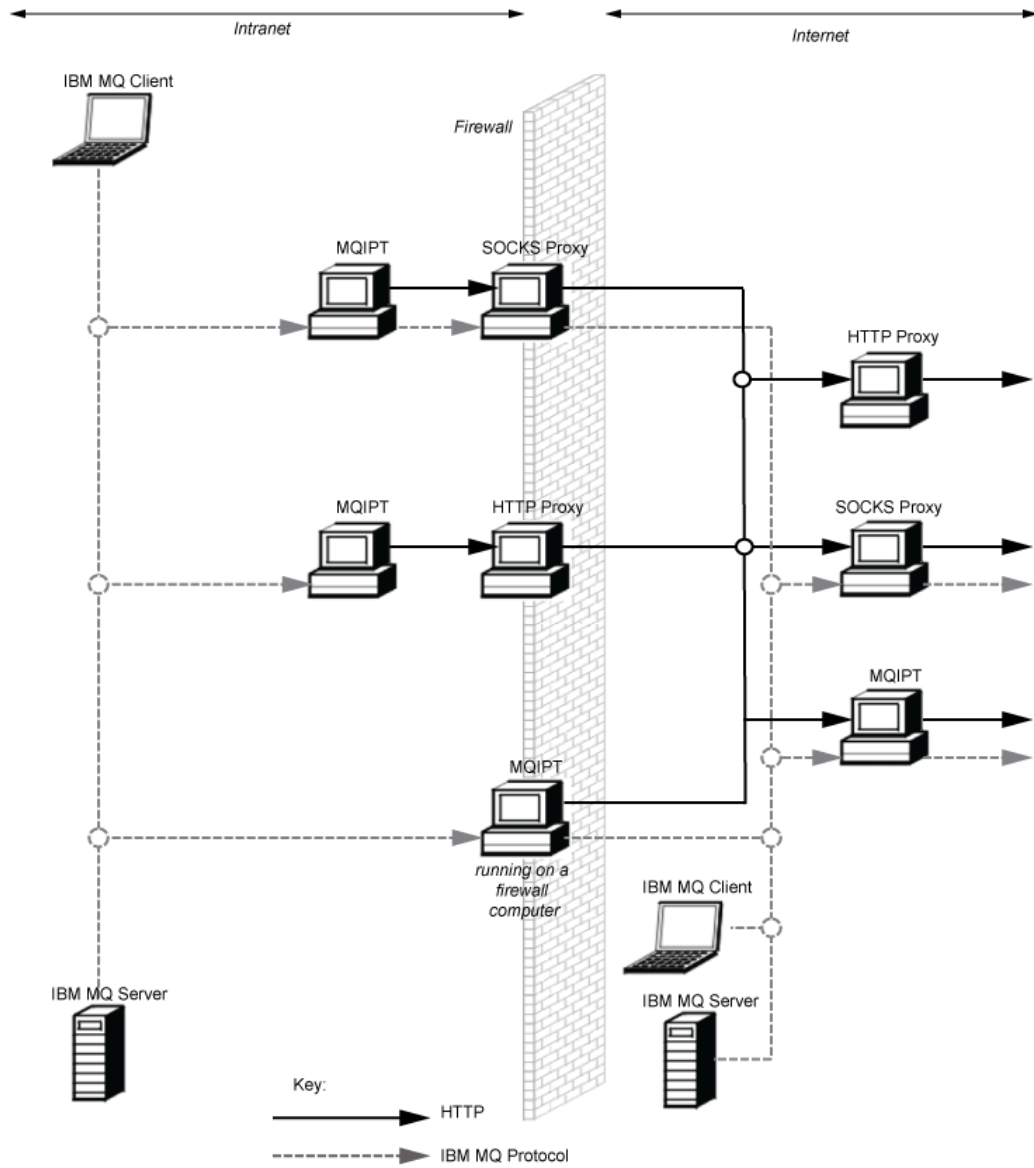
MQIPT pode ser usado em conjunto com IBM MQ e IBM Integration Bus.

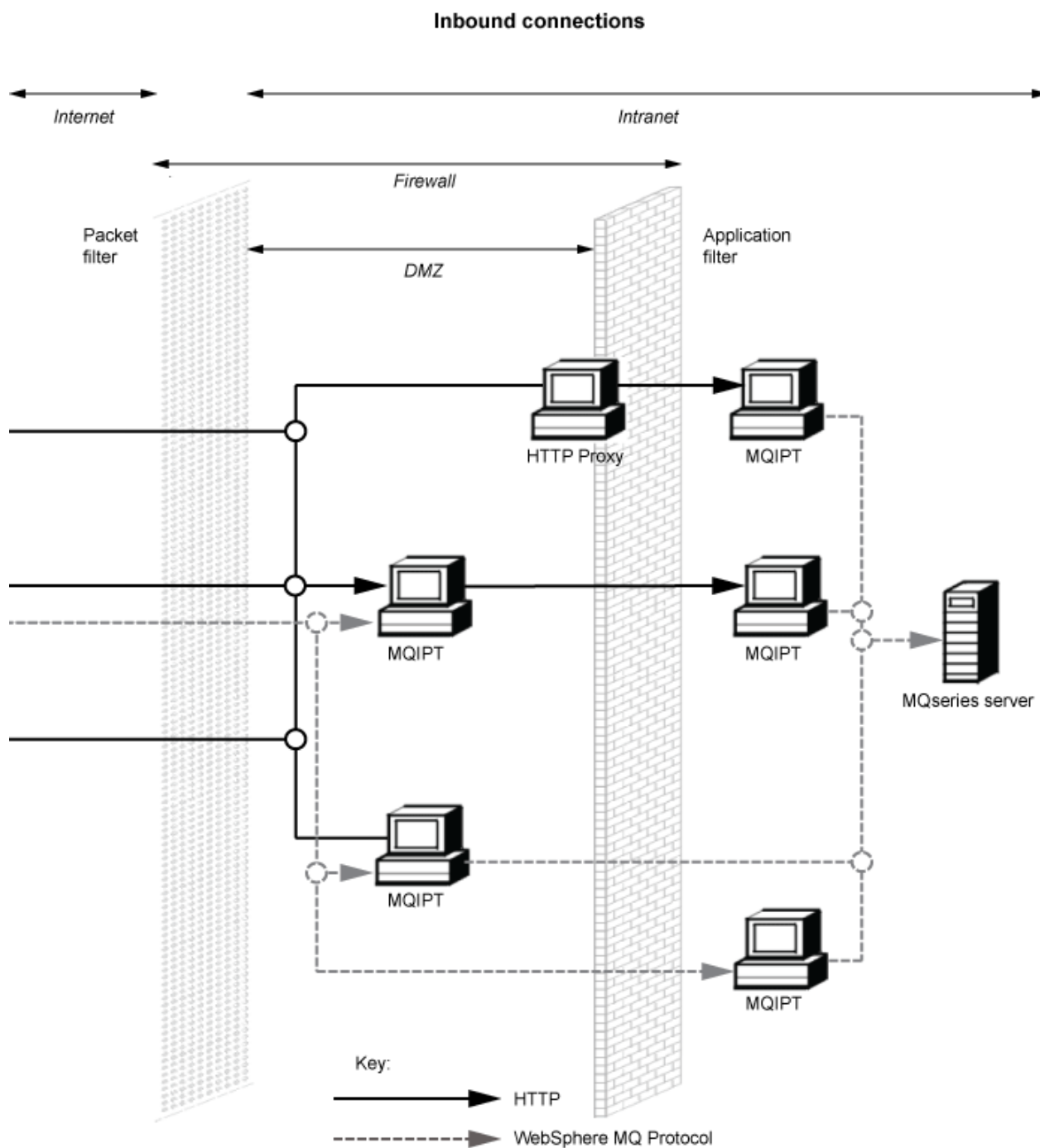
A figura de várias partes a seguir mostra muitas das configurações possíveis para o MQIPT em uma topologia do IBM MQ. Ele ilustra maneiras diferentes nas quais o MQIPT pode enviar mensagens. Ele mostra clientes e servidores em uma intranet, dentro de um firewall, e na Internet fora do firewall, transmitindo mensagens para MQIPT, proxy HTTP ou proxy SOCKS, o que os encaminha.

As mensagens são recebidas por um proxy MQIPT ou um proxy HTTP em um DMZ antes de passar a mensagem pelo firewall de entrada para um servidor.

Observe que o proxy HTTP, o proxy SOCKS e os computadores MQIPT no lado da intranet do firewall representam a possibilidade de vários computadores encadeados juntos na Internet. Por exemplo, um computador MQIPT pode se comunicar por meio de um ou mais computadores de proxy SOCKS ou HTTP, ou computadores adicionais MQIPT , antes de atingir seu destino.

Outbound connections





Configurações compatíveis

Cenários de conexão compatíveis em que um cliente IBM MQ ou gerenciador de filas se comunica com MQIPT. A mesma rota ou uma segunda rota do MQIPT é usada para se comunicar com um gerenciador de filas de destino.

Configurações compatíveis com uma única rota MQIPT

É possível usar uma única rota MQIPT para se comunicar com IBM MQ.

As colunas no [Tabela 26 na página 313](#) contêm as informações a seguir:

1. O protocolo usado entre IBM MQ e a rota MQIPT . A conexão pode ser criada por um cliente IBM MQ ou gerenciador de filas, e pode usar IBM MQ Formats and Protocols (FAP) ou um protocolo SSL/TLS.
2. O modo no qual a rota MQIPT opera. O formato da comunicação em toda a Internet entre MQIPT e IBM MQ é determinado pela configuração da rota MQIPT . Observe que, em que a tabela menciona SSL, também é possível usar TLS.
3. O protocolo usado entre a rota MQIPT e o gerenciador de filas de destino.

1. IBM MQ Protocolo de origem	2. Modo da rota MQIPT	3. IBM MQ Protocolo de destino
FAP	FAP-proxy (padrão)	FAP
	FAP-servidor e cliente SSL	SSL/TLS
SSL/TLS	SSL-proxy	SSL/TLS
	Servidor SSL e cliente FAP	FAP
	SSL-servidor e cliente SSL	SSL/TLS

Configurações compatíveis com mais de uma rota MQIPT

É possível optar por usar mais de uma rota, em uma ou mais instâncias de MQIPT, para se comunicar com IBM MQ.

As colunas no [Tabela 27 na página 314](#) contêm as informações a seguir:

1. O protocolo usado entre IBM MQ e a primeira rota MQIPT . A conexão pode ser criada por um cliente IBM MQ ou gerenciador de filas, e pode usar IBM MQ Formats and Protocols (FAP) ou um protocolo SSL/TLS.
2. O modo no qual a primeira rota MQIPT opera. O formato da comunicação em toda a Internet entre MQIPT e IBM MQ é determinado pela configuração da rota MQIPT . Observe que, em que a tabela menciona SSL, também é possível usar TLS.
3. O modo no qual a segunda rota MQIPT opera.
4. O protocolo usado entre a segunda rota MQIPT e o gerenciador de filas de destino.

Tabela 27. Configurações válidas com múltiplas instâncias do MQIPT

1. IBM MQ Protocolo de origem	2. Modo da primeira rota MQIPT	3. Modo da segunda rota MQIPT	4. IBM MQ Protocolo de destino
FAP (padrão)	FAP-proxy (padrão)	FAP-proxy (padrão)	FAP
	FAP-servidor e cliente SSL	SSL-proxy	SSL/TLS
		Servidor SSL e cliente FAP	FAP
		SSL-servidor e cliente SSL	SSL/TLS
	HTTP-cliente	HTTP-servidor e cliente SSL	SSL/TLS
	HTTPS-cliente	HTTPS-servidor e cliente SSL	SSL/TLS
	HTTP-cliente	HTTP-servidor	FAP
	HTTPS-cliente	HTTPS-servidor	FAP
SSL/TLS	SSL-proxy	SSL-proxy	SSL/TLS
		Servidor SSL e cliente FAP	FAP
		SSL-servidor e cliente SSL	SSL/TLS
	HTTP-cliente	HTTP-servidor	FAP
	HTTPS-cliente	HTTPS-servidor	SSL/TLS
	HTTP-cliente	HTTP-servidor e cliente SSL	FAP
	HTTPS-cliente	HTTPS-servidor e cliente SSL	SSL/TLS

Configurações de Canal Suportadas

Todos os tipos de canais IBM MQ são suportados, mas a configuração está restrita a conexões TCP/IP. Para um cliente IBM MQ ou gerenciador de filas, MQIPT aparece como se fosse o gerenciador de filas de destino. Em que a configuração do canal requer um host de destino e número de porta, o nome do host MQIPT e o número da porta do listener são especificados.

Canais cliente / servidor

MQIPT atende as solicitações de conexão do cliente recebidas e, em seguida, as encaminha usando os pacotes de protocolo HTTP Tunneling, SSL/TLS ou como padrão IBM MQ . Se MQIPT estiver usando o túnel HTTP ou SSL/TLS, ele os encaminha em uma conexão com um segundo MQIPT. Se não estiver usando o tunelamento HTTP, ele os encaminhará em uma conexão com o que ele vir como o gerenciador de filas de destino (embora isso possa, por sua vez, ser um MQIPT adicional). Quando o gerenciador de filas de destino tiver aceito a conexão do cliente, os pacotes serão retransmitidos entre o cliente e o servidor.

Emissor de cluster / canais receptores

Se MQIPT receber uma solicitação de entrada de um canal do emissor de clusters, ele assumirá que o gerenciador de filas tenha sido ativado para SOCKS e o endereço de destino verdadeiro será obtido durante o processo de handshaking SOCKS. Ele encaminha a solicitação para o próximo MQIPT ou gerenciador de filas de destino exatamente da mesma maneira que para canais de conexão do cliente. Isso também inclui canais de emissores de cluster definidos automaticamente.

Emissor/receptor

Se MQIPT receber uma solicitação recebida de um canal emissor, ele a encaminhará para o próximo gerenciador de filas de destino MQIPT ou de destino exatamente da mesma maneira que para canais de conexão do cliente. O gerenciador de filas de destino valida a solicitação recebida e inicia o canal receptor, caso seja apropriado. Todas as comunicações entre o canal emissor e receptor (incluindo os fluxos de segurança) são retransmitidas.

Solicitante-servidor

Essa combinação é manipulada da mesma maneira que as configurações anteriores. A validação da solicitação de conexão é executada pelo canal do servidor no gerenciador de filas de destino.

Solicitante-remetente

A configuração "retorno de chamada" poderia ser útil se os dois gerenciadores de filas não estiverem autorizados a estabelecer conexões diretas entre si, mas ambos podem se conectar ao MQIPT e aceitar conexões a partir dele.

Servidor / solicitante e servidor / receptor

Elas são manipuladas pelo MQIPT da mesma maneira que manipula a configuração `Sender/Receiver`.

Condições de término e falha do canal

Quando o MQIPT detecta o fechamento (normal ou anormal) de um canal IBM MQ, ele propaga o fechamento do canal. Se você fechar uma rota usando MQIPT, todos os canais que passarem por essa rota serão fechados.

MQIPT fornece um recurso de tempo limite inativo opcional. Se MQIPT detectar que um canal ficou inativo por um período de tempo excedendo o tempo limite, ele executará um encerramento imediato nas duas conexões em questão.

Os sistemas IBM MQ em qualquer extremidade do canal observam essas condições de encerramento anormais, como falhas de rede ou como finalização do canal pelo parceiro. O canal é, então, capaz de reiniciar e recuperar (se a falha ocorrer durante um período de protocolo em dúvida) como se MQIPT não estivesse sendo usado.

Segurança de Mensagens

O gerenciamento de filas distribuído do IBM MQ garante que as mensagens sejam entregues adequadamente. Esse ainda é o caso quando o MQIPT está presente entre as duas extremidades do canal. O MQIPT não armazena nenhum dado da mensagem ou participa do procedimento de ponto de sincronização que assegura a entrega de mensagem correta.

Ao usar mensagens IBM MQ rápidas, não persistentes, se a rota MQIPT falhar ou for reiniciado quando uma mensagem IBM MQ estiver em trânsito, a mensagem poderá ser perdida. Antes de reiniciar a rota, assegure-se de que todos os canais IBM MQ usando a rota MQIPT estejam inativos.

Para obter mais informações sobre a segurança de mensagens no IBM MQ, consulte [Segurança de mensagens](#).

Gerenciadores de filas de várias instâncias e alta disponibilidade

MQIPT pode ser usado com gerenciadores de filas de várias instâncias em ambientes de alta disponibilidade.

MQIPT não possui um estado persistente e, portanto, não há nenhum benefício na falha por meio de MQIPT para outro sistema. Em vez disso, tenha várias instâncias do MQIPT com arquivos de configuração do `mqipt.conf` idênticos em execução em sistemas diferentes. Monitore cada instância de MQIPT para disponibilidade e reinicie-a (no mesmo sistema), se necessário. Isso fornece um conjunto de instâncias MQIPT idênticas que podem ser usadas para rotear conexões. Em seguida, deve-se assegurar que IBM MQ possa rotear conexões para MQIPT e que MQIPT possa encaminhar essas conexões para o gerenciador de filas de destino.

Os canais IBM MQ de saída podem ser direcionados para uma instância disponível do MQIPT de várias maneiras, por exemplo:

- Use um balanceador de carga ou roteador de alta disponibilidade, como o IBM Network Dispatcher a partir do produto WebSphere Edge Components.
- Especifique vários nomes de conexão na definição de canal IBM MQ usando uma lista separada por vírgula. IBM MQ tenta, então, se conectar a cada endereço MQIPT, por vez, até localizar uma instância MQIPT disponível.

Você também deve direcionar conexões de MQIPT para o gerenciador de filas de destino. Se a configuração de alta disponibilidade assegurar que o endereço IP efetua failover com o gerenciador de filas de destino, nenhuma configuração especial do MQIPT será necessária: especifique o endereço IP de destino na propriedade de rota **Destination** e permita que a operação de failover mova o endereço IP com o gerenciador de filas.

No entanto, se o endereço IP do gerenciador de filas for alterado após um failover, você deverá organizar para MQIPT para encaminhar a conexão para o destino correto. Isso pode ser feito de uma de várias maneiras:

- Grave uma saída de roteamento que verifique qual endereço IP e número da porta estão acessíveis e, em seguida, substitua o destino de rota para cada conexão. Algumas saídas de roteamento de amostra são fornecidas com MQIPT; elas podem ser adaptadas para esse propósito.
- Use um balanceador de carga de alta disponibilidade para redirecionar a conexão.
- Defina diversas rotas MQIPT, uma para cada endereço IP e porta em que o gerenciador de filas pode estar em execução. Em seguida, direcione as conexões IBM MQ para as várias rotas MQIPT, por exemplo, listando todos os endereços IP de rota e números de porta em uma lista separada por vírgulas no nome da conexão do canal de saída.

Também é importante ajustar todos os componentes de ponta a ponta no caminho da rede:

1. As tentativas de conexão para sistemas indisponíveis devem falhar imediatamente para que as tentativas de reconexão possam se mover para o primeiro destino disponível.

Para rotas SSL do MQIPT, ajuste a rota **SSLClientConnectTimeout** para assegurar falha na conexão de prompt para destinos indisponíveis. Consulte a documentação do IBM MQ para obter detalhes dos parâmetros de ajuste IBM MQ. Além disso, consulte a documentação do sistema operacional para obter detalhes sobre o ajuste do TCP/IP para o sistema operacional. Em todos os casos, as tentativas de conexão com falha devem retornar rapidamente uma falha de rede (por exemplo, um pacote de reconfiguração TCP) ou deve passar o tempo limite sem atraso indevido.

2. As conexões ativas para um sistema com falha devem ser interrompidas imediatamente para que novas conexões possam ser estabelecidas.

Também é necessário considerar o impacto de um failover em um momento em que as conexões estão ativamente usando MQIPT. É provável que as conexões de rede sejam interrompidas durante um failover. Para aplicativos clientes, é possível usar o recurso de reconexão automática do cliente IBM MQ para restabelecer conexões interrompidas. Para canais de mensagens, é possível especificar um intervalo de nova tentativa curto para que o canal reconecte-se imediatamente. Consulte a documentação IBM MQ para obter mais informações sobre reconexão automática do cliente e a configuração de nova tentativa do canal de mensagem.

V 9.3.5 O IBM MQ Console e REST API

É possível usar IBM MQ Console e REST API para administrar IBM MQ e executar operações do sistema de mensagens usando HTTP.

- É possível usar o IBM MQ Console para executar tarefas básicas de administração de um navegador da web. Para obter mais informações, consulte [Administração Usando o IBM MQ Console](#).
- É possível usar a administrativa REST API para administrar objetos do IBM MQ, como gerenciadores de filas e filas e agentes do Managed File Transfer e transferências. Para obter mais informações, consulte [Administração Usando o REST API](#).

- É possível usar o messaging REST API para executar o sistema de mensagens simples ponto a ponto e de publicação Para obter mais informações, consulte [Sistema de mensagens que usa o REST API](#)

Opções de instalação

O IBM MQ Console e o REST API são executados em um servidor WebSphere Liberty , chamado mqweb No IBM MQ 9.3.5, é possível instalar o servidor mqweb como um componente opcional em uma instalação do IBM MQ ou como uma instalação independente do IBM MQ Web Server .

V 9.3.5

Linux

Instalação independente do IBM MQ Web Server

No IBM MQ 9.3.5, o servidor mqweb pode ser executado em uma instalação independente do IBM MQ Web Server. Uma instalação independente do IBM MQ Web Server permite instalar e executar o servidor mqweb em sistemas que são separados para suas instalações do IBM MQ . A instalação de um IBM MQ Web Server independente fornece maior flexibilidade sobre quais sistemas e o número de sistemas nos quais você escolhe executar seus servidores mqweb. Se necessário, várias instâncias do servidor mqweb podem ser executadas em máquinas diferentes para fornecer a escalabilidade e a disponibilidade necessárias.

Se você tiver comprado a autorização IBM MQ , será possível instalar quantas cópias forem necessárias do IBM MQ Web Server independente. As instalações do IBM MQ Web Server não são contadas com relação à sua autorização adquirida do IBM MQ. Para obter mais informações sobre o licenciamento do IBM MQ, consulte [Informações sobre licença do IBM MQ](#).

As restrições a seguir se aplicam em uma instalação independente do IBM MQ Web Server :

- O IBM MQ Console pode ser usado para administrar apenas os gerenciadores de filas remotas
- O messaging REST API pode ser usado apenas com gerenciadores de fila remotos..
- O administrative REST API não está disponível

O IBM MQ Web Server independente é suportado apenas em plataformas Linux ..

Para obter mais informações sobre como instalar o IBM MQ Web Server independente, consulte [Instalando o IBM MQ Web Server independente](#).

Componente opcional de uma instalação do IBM MQ

É possível optar por instalar o componente IBM MQ Console e REST API como parte de uma instalação do IBM MQ .

Todos os recursos IBM MQ Console e REST API estão disponíveis quando o servidor mqweb é executado em uma instalação do IBM MQ .

- O IBM MQ Console pode ser usado para administrar gerenciadores de fila locais e remotos..
- O messaging REST API pode ser usado com os gerenciadores de filas locais e remotos
- O administrative REST API pode ser usado para administrar gerenciadores de fila locais e remotos..

Para usar o componente IBM MQ Console e REST API , instale o componente a seguir como parte de sua instalação do IBM MQ :

- **AIX** No AIX, instale o conjunto de arquivos mqm.web.rte.
- **IBM i** No IBM i, instale o componente WEB.
- **Linux** No Linux, instale o componente MQSeriesWeb.
- **Windows** No Windows, instale o recurso Web Administration.
- **z/OS** No z/OS, instale o recurso IBM MQ for z/OS UNIX System Services Web Components.

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte seu representante local do IBM para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja nenhum direito de propriedade intelectual da IBM. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou aplicativos de patentes pendentes relativas aos assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum sobre tais patentes. É possível enviar pedidos de licença, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
U.S.A.

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. IBM pode aperfeiçoar e/ou alterar no produto(s) e/ou programa(s) descritos nesta publicação a qualquer momento sem aviso prévio.

Referências nestas informações a websites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Os licenciados deste programa que desejarem obter informações sobre este assunto com o propósito de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações trocadas, deverão entrar em contato com:

Av. Pasteur, 138-146
Av. Pasteur, 138-146

Botafogo
Rio de Janeiro, RJ
U.S.A.

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do IBM Customer Agreement, IBM Contrato de Licença do Programa Internacional ou qualquer contrato equivalente entre as partes.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disto, algumas medidas podem ter sido estimadas através de extrapolação. Os resultados reais podem variar. usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam somente metas e objetivos.

Essas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas.

Se estiver visualizando estas informações em formato eletrônico, as fotografias e ilustrações coloridas poderão não aparecer.

Informações sobre a Interface de Programação

As informações da interface de programação, se fornecidas, destinam-se a ajudá-lo a criar software aplicativo para uso com este programa.

Este manual contém informações sobre interfaces de programação desejadas que permitem que o cliente grave programas para obter os serviços do WebSphere MQ.

No entanto, estas informações também podem conter informações sobre diagnósticos, modificações e ajustes. As informações sobre diagnósticos, modificações e ajustes são fornecidas para ajudá-lo a depurar seu software aplicativo.

Importante: Não use essas informações de diagnóstico, modificação e ajuste como uma interface de programação, pois elas estão sujeitas a mudanças

Marcas comerciais

IBM, o logotipo IBM , ibm.com, são marcas registradas da IBM Corporation, registradas em várias jurisdições no mundo todo Uma lista atual de marcas registradas da IBM está disponível na Web em "Informações de copyright e marca registrada" www.ibm.com/legal/copytrade.shtml. 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: