

WebSphere MQ Everyplace V2.0.2

Índice

Trabalhando com o WebSphere

Message Broker	1
Cenários	1
Configuração da Ponte	3
Classes do Transformador de Ponte do MQe.	5
Classes do Objeto de Mensagem do MQe.	6
Gravando Aplicativos MQe para Direcionar Aplicativos MQ	8

Gravando Aplicativos de Publicação/Assinatura do MQe	9
Exemplos	10
Migrando Aplicativos MQe que Utilizam o Nó do MQe no Intermediário.	11
Utilizando o MQePubMsgObject	11

Trabalhando com o WebSphere Message Broker

Cenários

Os aplicativos MQe em gerenciadores de filas MQe diferentes comunicam-se uns com os outros através da troca de mensagens. No caso mais simples, os gerenciadores de fila de origem e de destino são diretamente conectados uns aos outros através de canais MQe ou indiretamente conectados com as mensagens entre eles passando pelos gerenciadores de fila MQe intermediários. Em todos os casos, a classe do objeto de mensagem MQe padrão `com.ibm.mqe.MQeMsgObject` é utilizada para criar essas mensagens ou algumas subclasses apropriadas (como no caso das mensagens administrativas). A situação torna-se mais complicada quando os gerenciadores de filas intermediários não são gerenciadores de filas MQe, mas sim gerenciadores do WebSphere MQ. Nesse caso, as mensagens podem ser consideradas como tunnelling através de uma rede MQ, sem que a passagem através da rede MQ tenha efeito sobre o aplicativo de envio ou de recebimento. O tunnelling requer a seguinte configuração:

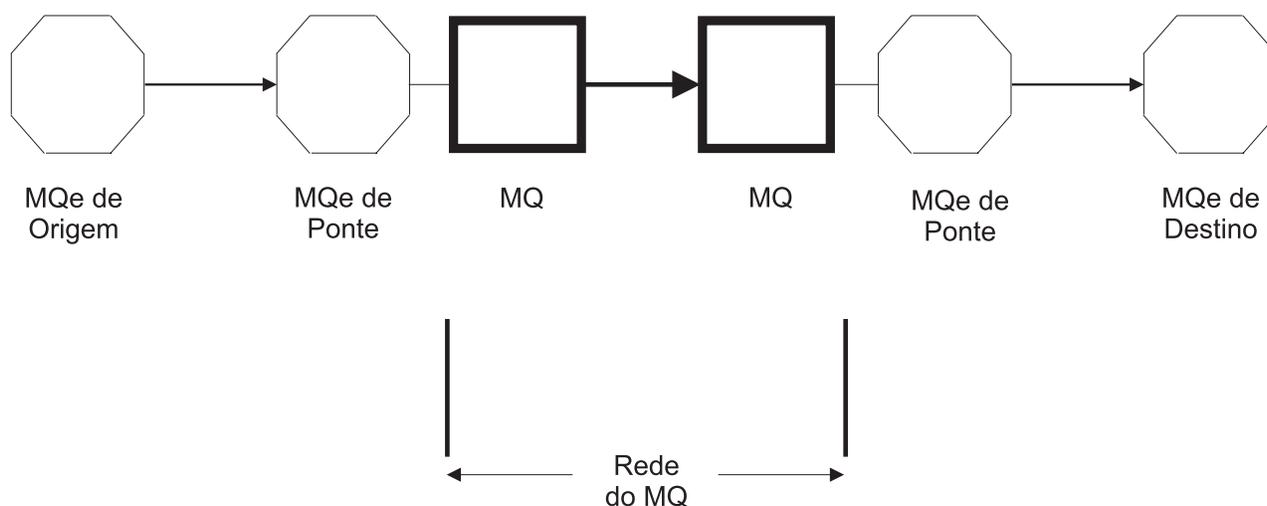


Figura 1. Tunnelling de Mensagens do MQe

O gerenciador de filas MQe que transmite as mensagens para um gerenciador de filas MQ é configurado como um gerenciador de filas de ponte. Do mesmo modo, o gerenciador de filas MQe que recebe mensagens de um gerenciador de filas MQ também é configurado como um gerenciador de filas de ponte. Os gerenciadores de filas de ponte às vezes são referidos como gerenciadores de filas de gateway. Os detalhes da configuração da ponte são descritos em outra parte; aqui é suficiente entender que uma única ponte configurada pode manipular as mensagens que se movem em ambas as direções (por exemplo, para e do MQ) e que os determinantes mais importantes em seu comportamento são: (a) o valor da propriedade da classe do transformador (definido na configuração da ponte) e (b) para as mensagens transmitidas do MQe para MQ, a classe da mensagem MQe. Em muitos casos, não é um requisito transportar mensagens através de uma rede MQ, mas sim utilizar mensagens de um aplicativo de origem MQe para direcionar um aplicativo de destino MQ, normalmente retornando os resultados em uma mensagem de resposta. A configuração envolvida tem a forma mostrada abaixo, embora o aplicativo de destino possa ser executado em qualquer gerenciador de filas MQ, incluindo o que está imediatamente conectado à ponte MQe:

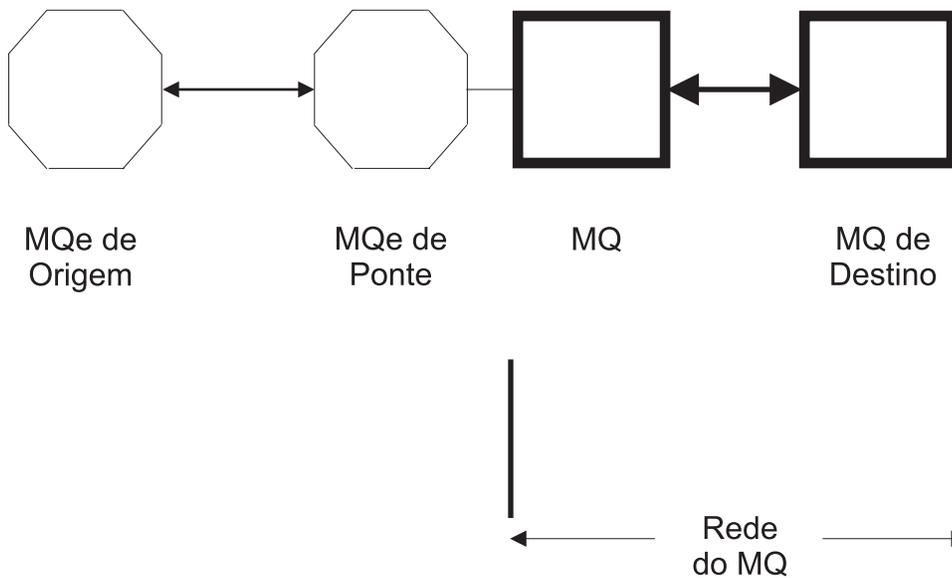


Figura 2. Direcionando Aplicativos MQ

Conforme foi dito anteriormente, os determinantes mais importantes no comportamento da ponte são:

- O valor da propriedade de classe do transformador (definido na configuração da ponte)
- Para mensagens sendo transmitidas do MQe para o MQ, a classe da mensagem MQe

A classe da mensagem utilizada pela origem deve permitir o controle total sobre a mensagem MQ gerada sendo recebida pelo aplicativo MQ. Geralmente, a classe `com.ibm.mqe.mqemqmessage.MQeMQMsgObject` é utilizada para essa finalidade. No entanto, se o JMS estiver sendo utilizado como o modelo de mensagem na origem e no destino, a classe de mensagem `com.ibm.mqe.MQeMsgObject` será utilizada, embora um conteúdo bastante específico esteja presente.

Um requisito adicional é suportar as necessidades dos aplicativos de publicação e assinatura, onde um intermediário IBM é o aplicativo MQ de destino e gerencia tanto as assinaturas quanto as publicações. A origem MQe tende a ser apenas uma das muitas entradas de origem para o intermediário. Existem duas configurações possíveis para implementar uma rede de publicação/assinatura, no entanto, somente uma é recomendada agora. Ela é:

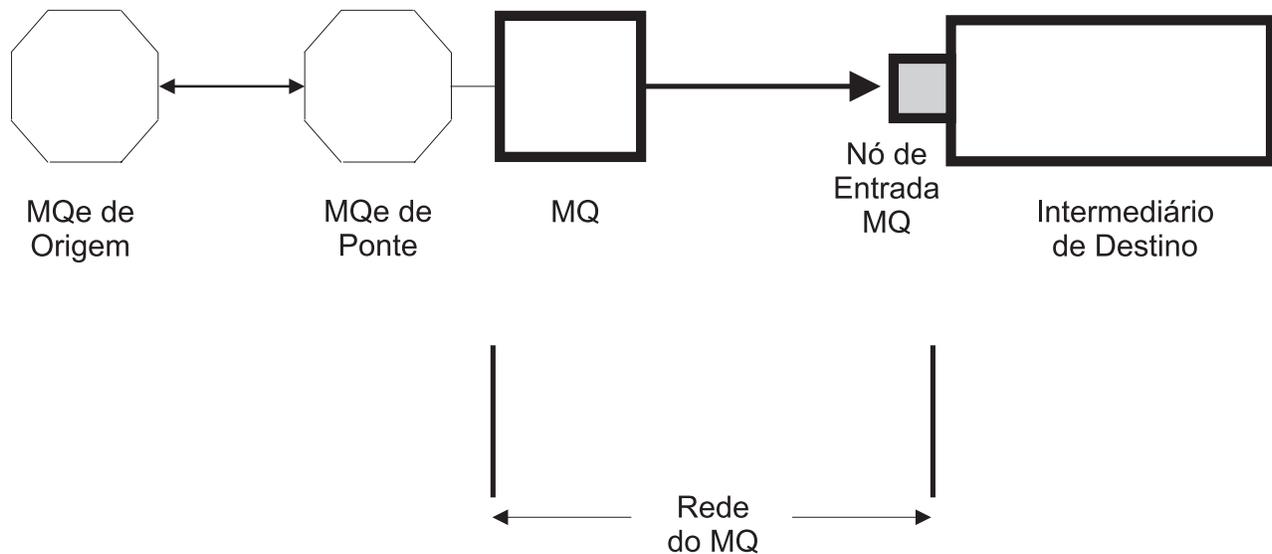


Figura 3. Publicação/Assinatura através da Ponte MQe

Mais uma vez, os determinantes mais importantes no comportamento da ponte são: (a) o valor da propriedade da classe do transformador (definido na configuração da ponte) e (b) para as mensagens transmitidas do MQe para MQ, a classe da mensagem MQe. Aqui, a escolha da mensagem utilizada pela origem deve permitir que uma mensagem MQ de publicação/assinatura seja enviada pela ponte e, subsequentemente, recebida pelo intermediário. A classe `com.ibm.mqe.mqemqmessage.MQePubSubMsgObject` é utilizada para essa finalidade. Para respostas vindas do intermediário, a classe do transformador da ponte detecta no conteúdo da mensagem MQ que essa é uma resposta de publicação/assinatura e, por essa razão, retorna uma mensagem de classe `com.ibm.mqe.mqemqmessage.MQePubSubMsgObject` de volta para a origem.

Nas versões anteriores do MQe, a seguinte configuração era suportada:



Figura 4. Publicação/Assinatura através do Nó de Entrada do Intermediário MQe

A origem enviou uma classe `com.ibm.broker.mqimqe.wrapper.MQeMbMsgObject` ou uma mensagem `com.ibm.mqe.MQeMsgObject` para o nó de entrada do MQe no intermediário. Essa configuração não é mais recomendada; os cenários de migração de implementações anteriores serão discutidos posteriormente nesta seção.

Configuração da Ponte

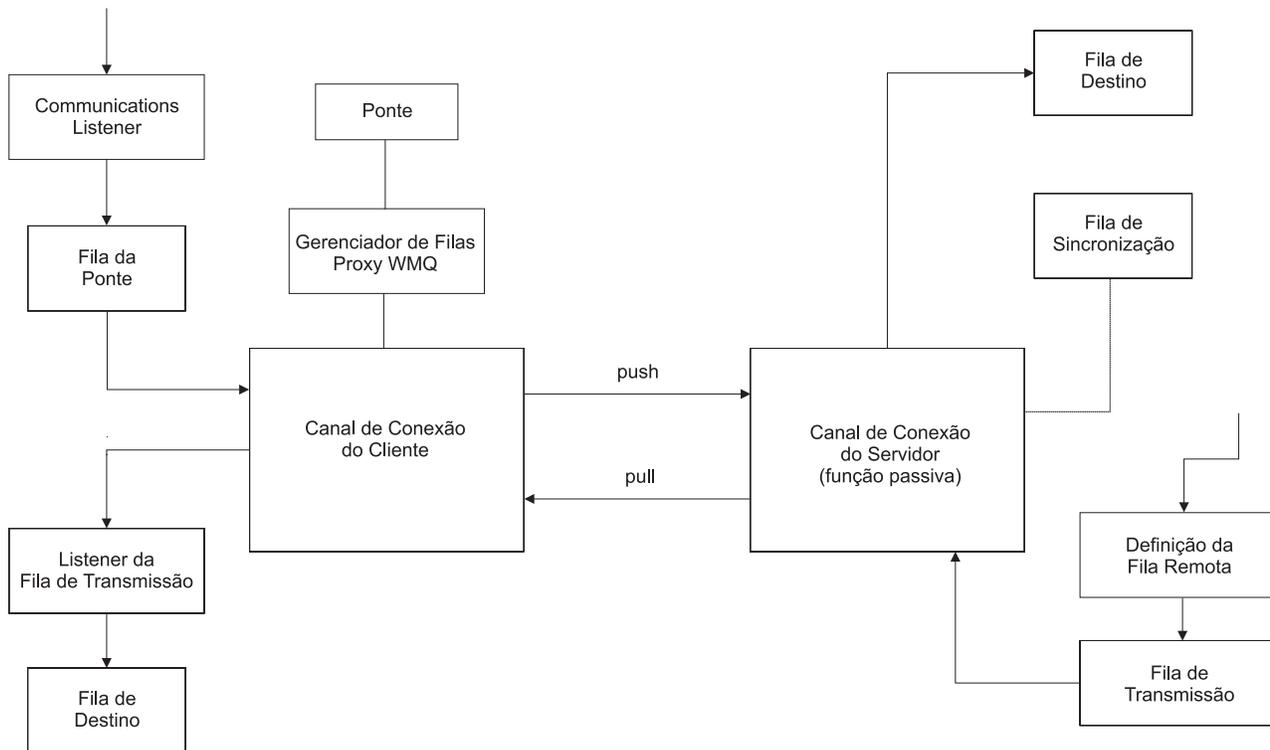
O seguinte resumo da ponte MQe é fornecido para auxiliar no entendimento das configurações e tarefas descritas neste documento.

A ponte MQe (ou gateway) é um gerenciador de filas MQe especialmente configurado que age como um link entre uma rede MQe e uma rede MQ. Para essas finalidades, um gerenciador de filas do WebSphere

Message Broker é considerado apenas um outro gerenciador de filas MQ na rede MQ. Uma ponte inclui os seguintes elementos que são vinculados através de um relacionamento hierárquico em forma de árvore:

- Ponte
- Proxy MQ
- Conexão do Cliente
- Listener

O diagrama a seguir mostra os objetos MQe e MQ requeridos para uma ponte e para o fluxo das mensagens.



Em cada nível na hierarquia, diversos elementos podem existir (por ex., diversos proxies MQ por ponte, diversas conexões do cliente por proxy MQ, etc.). O objeto da ponte reflete a hierarquia global e estabelece algum comportamento padrão. O proxy MQ identifica o gerenciador de filas MQ de destino; a conexão do cliente descreve os detalhes dos mecanismos através dos quais o MQe e o MQ se comunicam. Finalmente, o listener fornece as informações necessárias para mover mensagens do MQ para o MQe. Considera-se, então, que a ponte pode fornecer a capacidade de endereçamento no nível do gerenciador de filas entre as duas redes.

Um elemento da configuração da ponte envolve a especificação da classe do transformador a ser utilizada; esse transformador será chamado sempre que uma mensagem MQe precisar ser enviada para o MQ ou sempre que uma mensagem MQ tiver que ser enviada para o MQe.

Após uma ponte ter sido configurada é necessário, nesse mesmo gerenciador de filas MQe, incluir detalhes das filas MQ que não estão acessíveis a partir do MQe. Isso envolve a definição das filas de ponte e cada definição identifica uma fila de destino no MQ. As filas de pontes, então, fornecem capacidade de endereçamento no nível da fila entre as duas redes.

Classes do Transformador de Ponte do MQe

A classe do transformador de ponte controla a conversão de mensagens na ponte, isto é, as mensagens sendo recebidas de um gerenciador de filas MQe e enviadas para um gerenciador de filas MQ e o movimento das mensagens na direção inversa. Um número de transformadores é fornecido, no entanto, alguns deles são históricos e agora são obsoletos. Os transformadores recomendados são fornecidos abaixo, junto com seu relacionamento de classe:

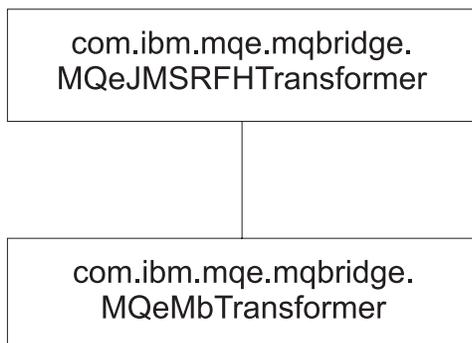


Figura 5. Hierarquia de Classe do Transformador de Ponte

Os transformadores recomendados por tarefa são mostrados em negrito na tabela abaixo; outros transformadores que podem ser utilizados (e que produzem resultados idênticos) são mostrados no tipo normal:

Tabela 1. Transformadores de Ponte MQe por Tarefa

Tarefa	Transformador
Tunnelling de Mensagens do MQe	com.ibm.mqe.mqbridge.MQeJMSRFHTransformer com.ibm.mqe.mqbridge.MQeBaseTransformer com.ibm.mqe.mqbridge.MQeGeneralRFHTransformer
Uso do JMS com MQe/MQ	com.ibm.mqe.mqbridge.MQeJMSRFHTransformer
Direcionando Aplicativos MQ	com.ibm.mqe.mqbridge.MQeJMSRFHTransformer com.ibm.mqe.mqbridge.MQeBaseTransformer com.ibm.mqe.mqbridge.MQeGeneralRFHTransformer
Publicação/Assinatura (através do MQePubSubMsgObject)	com.ibm.mqe.mqbridge.MQeJMSRFHTransformer
Publicação/Assinatura (através do MQeMbMsgObject)	com.ibm.mqe.mqbridge.MQeMbTransformer

Os recursos das classes do transformador recomendado são:

Tabela 2. Recursos da Classe do Transformador de Ponte

Classe do Transformador	Recursos (Conversão de MQe em MQ)
com.ibm.mqe.mqbridge.MQeJMSRFHTransformer	<p>As mensagens MQe de entrada da classe MQePubSubMsgObject serão convertidas como mensagens de publicação/assinatura se contiverem dados de publicação/assinatura para direcionar o intermediário; caso contrário, elas serão tratadas como mensagens MQeMQMsgObject.</p> <p>As mensagens da classe MQeMQMsgObject são processadas para configurar o MQMD e a carga útil da mensagem MQ de saída de modo que elas possam direcionar aplicativos MQ.</p> <p>Todas as outras mensagens são inspecionadas para verificar se elas possuem conteúdo JMS e, em caso afirmativo, elas são convertidas adequadamente; caso contrário, é efetuado tunnelling nas mensagens através da rede MQ.</p>
com.ibm.mqe.mqbridge.MQeMbTransformer	<p>Inclui suporte para a MQeMbMsgObject.</p> <p>Qualquer mensagem que chega da classe MQeMbMsgObject é convertida na MQePubSubMsgObject equivalente e, então, transmitida para o transformador de superclasse. Todas as outras classes de mensagens de entrada não são afetadas.</p>
Classe do Transformador	Recursos (Conversão de MQ em MQe)
com.ibm.mqe.mqbridge.MQeJMSRFHTransformer	<p>As mensagens MQ que chegam contendo mensagens MQe tunnelled são convertidas em uma mensagem da classe MQeMsgObject.</p> <p>As mensagens MQ contendo dados JMS são convertidas em mensagens MQeMsgObject com dados JMS.</p> <p>As mensagens MQ contendo respostas de um intermediário de publicação/assinatura são convertidas em mensagens MQePubSubMsgObject.</p> <p>Todas as outras mensagens são convertidas em mensagens MQeMQMsgObject.</p>
com.ibm.mqe.mqbridge.MQeMbTransformer	<p>Inclui suporte para a MQeMbMsgObject.</p> <p>Qualquer saída de mensagem MQe da classe MQePubSubMsgObject, gerada pelo transformador de superclasse, é convertida em uma mensagem MQeMbMsgObject equivalente. Todas as mensagens de saída de outras classes não são afetadas</p>

Classes do Objeto de Mensagem do MQe

As mensagens da origem MQe para a ponte (ou para o nó de entrada MQe no intermediário) são processadas de acordo com a classe de mensagem. A hierarquia de classe das classes de mensagem fornecidas pelo MQe é mostrada pelas caixas sombreadas no diagrama abaixo. Observe que todas as classes de mensagem são herdadas da classe com.ibm.mqe.MQeFields, o que fornece uma capacidade genérica de receber/enviar valores de dados nos objetos de mensagem.

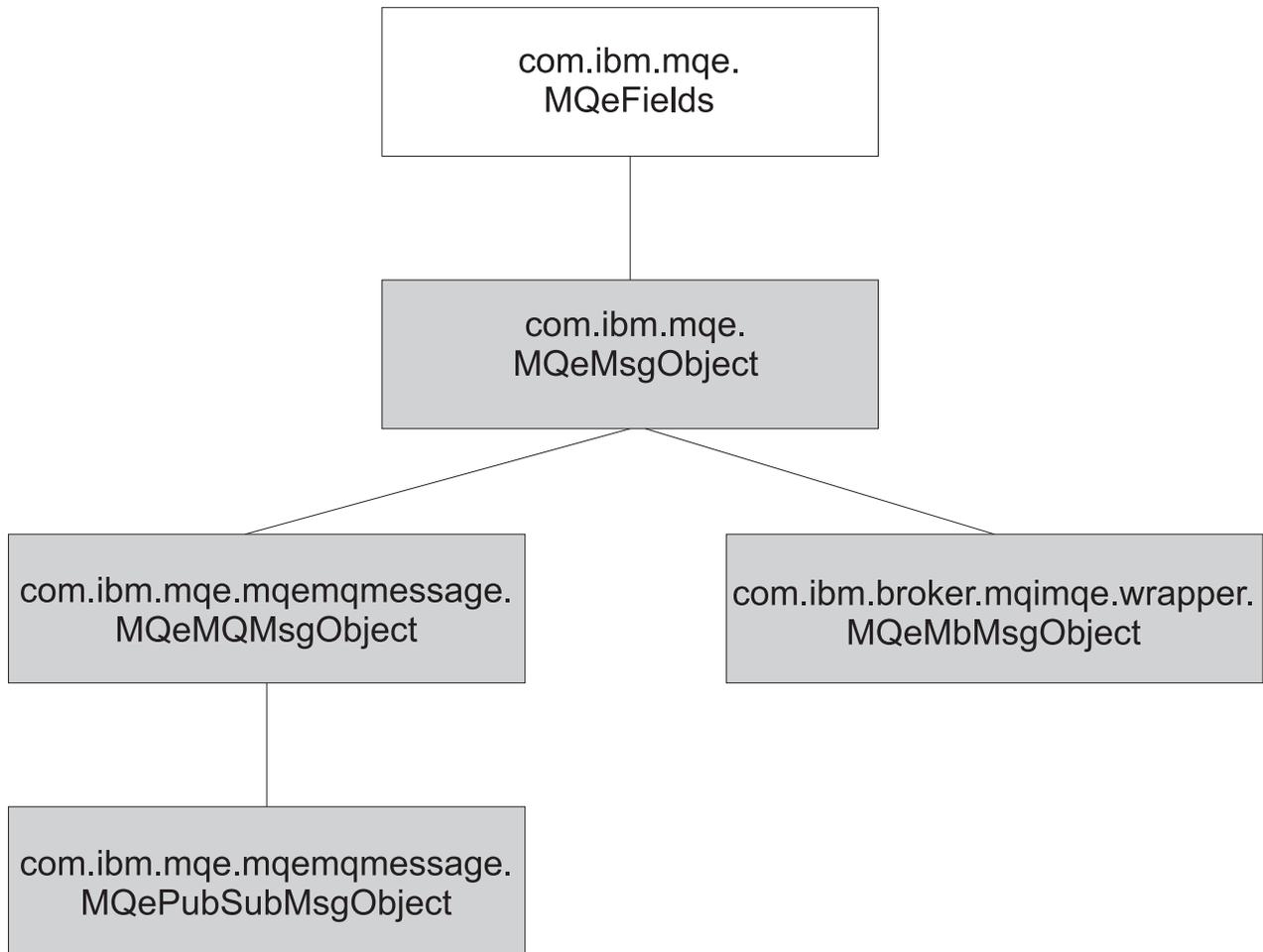


Figura 6. Hierarquia da Classe de Mensagem

Os recursos das várias classes são:

Tabela 3. Recursos da Classe de Mensagem MQe

Classe do Objeto de Mensagem	Recursos
com.ibm.mqe.MQeMsgObject	Classe de mensagem MQe de base. Métodos MQeFields utilizados para receber/enviar dados do usuário.
com.ibm.mqe.mqemqmessage.MQeMQMsgObject	Inclui a capacidade de configurar o conteúdo da MQMD e da carga útil da mensagem MQ que será gerada pela ponte.
com.ibm.mqe.mqemqmessage.MQePubSubMsgObject	Inclui métodos de publicação/assinatura específicos para direcionar o comportamento do intermediário.
com.ibm.broker.mqimqe.wrapper.MQeMbMsgObject	Inclui a capacidade de configurar os dados equivalentes aos mantidos na MQMD de uma mensagem MQ. Os métodos MQeFields que utilizam constantes definidas são utilizados para implementar a funcionalidade de publicação/assinatura.

As classes de mensagem recomendadas apropriadas para as várias tarefas são mostradas na tabela abaixo:

Tabela 4. Classes de Mensagem MQe por Tarefa

Tarefa	Classe do Objeto de Mensagem
Tunnelling de Mensagens do MQe	<code>com.ibm.mqe.MQeMsgObject</code> (ou qualquer subclasse apropriada para o aplicativo)
Uso do JMS com MQe/MQ	<code>com.ibm.mqe.MQeMsgObject</code> (com conteúdo especificamente formatado)
Direcionando Aplicativos MQ	<p><code>com.ibm.mqe.mqemqmessage.MQeMQMsgObject</code> (ou qualquer subclasse apropriada para o aplicativo)</p> <p>Nota: O conjunto de caracteres padrão para a <code>MQeMQMsgObject</code> é 1200. Quando utilizado com o Websphere Message Broker, ele pode ser alterado para 1208.</p> <p><code>com.ibm.mqe.mqemqmessage.MQePubSubMsgObject</code> (ou qualquer subclasse apropriada para o aplicativo)</p>
Publicação/Assinatura (através do <code>MQePubSubMsgObject</code>)	<code>com.ibm.mqe.mqemqmessage.MQePubSubMsgObject</code> (ou qualquer subclasse apropriada para o aplicativo)
Publicação/Assinatura (através do <code>MQeMbMsgObject</code>)	<code>com.ibm.broker.mqimqe.wrapper.MQeMbMsgObject</code>

Gravando Aplicativos MQe para Direcionar Aplicativos MQ

Existem dois aspectos para gravar um aplicativo MQe para direcionar um aplicativo MQ. São eles:

Configuração

- Configure uma ponte para fornecer acesso ao gerenciador de filas MQ de destino a partir da rede MQe, especificando a classe do transformador `com.ibm.mqe.mqbridge.MQeJMSRFHTransformer`. Também no gerenciador de filas de ponte, configure duas filas de ponte:
 - Uma para agir como um proxy para a fila de controle `SYSTEM.BROKER.CONTROL.QUEUE` do intermediário.
 - Uma para agir como um proxy para a fila de entrada do fluxo de mensagens apropriado.
- No gerenciador de filas de origem MQe, configure conexões para o gerenciador de filas de ponte MQe e o gerenciador de filas do aplicativo MQ de destino (sendo que essa segunda conexão é feita através do gerenciador de filas de ponte).
- No gerenciador de filas MQe de origem, configure uma fila de proxy remoto para a fila do aplicativo MQ de destino.
 - Uma para agir como um proxy para a fila de controle `SYSTEM.BROKER.CONTROL.QUEUE` do intermediário.
 - Uma para agir como um proxy para a fila de entrada do fluxo de mensagens apropriado.

Desenvolvimento do Aplicativo

- Construa uma mensagem MQe que será transformada pela ponte em uma mensagem MQ adequada para o intermediário. Essa mensagem terá um cabeçalho MQMD padrão e um cabeçalho RFH2, bem como uma carga útil apropriada.
- Envie as mensagens de assinatura e cancelamento de assinatura para a fila de controle do intermediário. Envie mensagens de publicação para a fila de entrada do fluxo de mensagens apropriado.

A mensagem deve ser da classe `com.ibm.mqe.mqemqmessage.MQePubSubMsgObject`. Caso seja necessário utilizar um MQ MQMD indicado, os valores serão definidos no objeto de mensagem (como

setCorrelationId, setMessageId, setExpiry, setFeedback, setReport, setUserId). As operações de publicação/assinatura são chamadas através dos métodos de publicação, assinatura e cancelamento de assinatura.

As respostas do intermediário serão recebidas como mensagens da mesma classe. O sucesso global de uma operação é fornecido pelo método getCompletionCode; informações mais detalhadas são fornecidas por getResponses. O intermediário nunca retorna os dados de publicação, mas as informações sobre a natureza do pedido original estão disponíveis através de getActionType, getDestQueueMgr, getDestQueueName, getRetention e getTopics.

Para as mensagens que foram construídas, os seguintes métodos podem ser utilizados para retornar informações sobre o pedido: getMessage e getMessageData.

Informações adicionais sobre os detalhes de configuração apropriados para uma ponte são fornecidas nas publicações do MQE, incluindo a manipulação de respostas de retorno do aplicativo MQ de destino.

Gravando Aplicativos de Publicação/Assinatura do MQE

Existem dois aspectos para gravar um aplicativo MQE para direcionar um aplicativo do intermediário MQ. São eles:

Configuração

1. Configure uma ponte para fornecer acesso ao gerenciador de filas MQ de destino a partir da rede MQE especificando a classe do transformador com.ibm.mqe.mqbridge.MQeJMSRFHTransformer. Também no gerenciador de filas de ponte, configure duas filas de ponte:
 - Uma para agir como um proxy para a fila de controle SYSTEM.BROKER.CONTROL.QUEUE do intermediário.
 - Uma para agir como um proxy para a fila de entrada do fluxo de mensagens apropriado.
2. No gerenciador de filas de origem MQE, configure conexões para o gerenciador de filas de ponte MQE e o gerenciador de filas do intermediário de destino (sendo que essa segunda conexão é feita através do gerenciador de filas de ponte).
3. No gerenciador de filas de origem MQE, configure duas filas de proxy remoto:
 - Uma para agir como um proxy para a fila de controle SYSTEM.BROKER.CONTROL.QUEUE do intermediário.
 - Uma para agir como um proxy para a fila de entrada do fluxo de mensagens apropriado.

Desenvolvimento do Aplicativo

1. Construa uma mensagem MQE que será transformada pela ponte em uma mensagem MQ adequada para o intermediário. Essa mensagem terá um cabeçalho MQMD padrão e um cabeçalho RFH2, bem como uma carga útil apropriada.
2. Envie as mensagens de assinatura e cancelamento de assinatura para a fila de controle do intermediário. Envie mensagens de publicação para a fila de entrada do fluxo de mensagens apropriado.

A mensagem deve ser da classe com.ibm.mqe.mqemqmessage.MQePubSubMsgObject. Caso seja necessário utilizar um WMQ MQMD indicado, os valores serão definidos no objeto de mensagem (por exemplo, setCorrelationId, setMessageId, setExpiry, setFeedback, setReport, setUserId e assim por diante). As operações de publicação/assinatura são chamadas através dos métodos de publicação, assinatura e cancelamento de assinatura.

As respostas do intermediário serão recebidas como mensagens da mesma classe. O sucesso global de uma operação é fornecido pelo método getCompletionCode; informações mais detalhadas são fornecidas

por `getResponses`. O intermediário nunca retorna os dados de publicação, mas as informações sobre a natureza do pedido original estão disponíveis através de `getActionType`, `getDestQueueMgr`, `getDestQueueName`, `getRetention` e `getTopics`.

Para as mensagens que foram construídas, os seguintes métodos podem ser utilizados para retornar informações sobre o pedido: `getMessage` e `getMessageData`.

Informações adicionais sobre os detalhes de configuração apropriados para uma ponte são fornecidas nas publicações do MQE, incluindo a manipulação de respostas de retorno do aplicativo MQ de destino.

Exemplos

Todos os exemplos a seguir são baseados naqueles citados no *MQI Programming Guide*, no Apêndice C, "MQSeries Everyplace Nodes". Um exemplo de um aplicativo de publicação:

```
try
{
    System.out.println("nome do QM local: " + myQMgr.getName());

    MQePubSubMsgObject mqeMsg = new MQePubSubMsgObject();
    mqeMsg.publish("Weather", true, "Hello");

    System.out.println("..Enviar mensagem para QM/fila: " +
        brokerQueueManager + "/" + flowInputQueue);
    myQM.putMessage(brokerQueueManager, flowInputQueue, mqeMsg, null, 0);
    System.out.println("Concluído");
}
catch (Exception e)
{
    e.printStackTrace();
    System.out.println("Falha! + e);
}
```

Um exemplo de um aplicativo de assinatura:

```
try
{
    System.out.println("nome do QM local: " + myQMgr.getName());

    MQePubSubMsgObject mqeMsg = new MQePubSubMsgObject();
    String[] topics = new String[]{"Topic1", "Topic2", "Topic3"};
    mqeMsg.subscribe(topics, "ServerQM1", "Caixa de Entrada");

    System.out.println("..Enviar mensagem para QM/fila: " +
        brokerQueueManager + "/" + controlQueue);
    myQM.putMessage(brokerQueueManager, "SYSTEM.BROKER.ADMIN.QUEUE", mqeMsg, null, 0);
    System.out.println("Concluído");
}
catch (Exception e)
{
    e.printStackTrace();
    System.out.println("Falha! + e);
}
```

Um exemplo de um aplicativo de cancelamento de assinatura:

```
try
{
    System.out.println("nome do QM local: " + myQMgr.getName());

    MQePubSubMsgObject mqeMsg = new MQePubSubMsgObject();
    String[] topics = new String[]{"Topic1", "Topic2", "Topic3"};
    mqeMsg.unsubscribe(topics, "ServerQM1", "Inbox");

    System.out.println("..Enviar mensagem para QM/fila: " +
        brokerQueueManager + "/" + controlQueue);
}
```

```

        myQM.putMessage(brokerQueueManager, "SYSTEM.BROKER.ADMIN.QUEUE", mqeMsg, null, 0);
        System.out.println("Concluído");
    }
    catch (Exception e)
    {
        e.printStackTrace();
        System.out.println("Falha! + e);
    }

```

Migrando Aplicativos MQe que Utilizam o Nó do MQe no Intermediário

A utilização do nó MQe no intermediário não é mais recomendada. Em vez disso, os aplicativos devem ser migrados para utilizar a ponte MQe; o intermediário então recebe uma mensagem MQ em vez de uma mensagem MQe. Existem três cenários de migração mostrados a seguir. Eles são listados na ordem, com o cenário preferencial primeiro – no entanto, o esforço necessário pode ser visto como inversamente proporcional à preferência. É recomendado que você leia todos os cenários com atenção e decida sua abordagem. Como todos os cenários requerem uma ponte MQe, você deverá fazer o download do MQe Server Support SupportPac, visto que ele ajudará muito na configuração da ponte. O MQe Server Support SupportPac está disponível a partir de: <http://www-306.ibm.com/software/integration/support/supportpacs/product.html>

Utilizando o MQePubMsgObject

Esta migração utiliza a ponte MQe no lugar do nó do intermediário MQe e evita a utilização de quaisquer classes MQe obsoletas.

O trabalho a seguir é requerido:

1. Modifique o fluxo de mensagens no intermediário de modo que os nós de entrada/saída MQ sejam utilizados no lugar dos nós de entrada/saída MQe.
2. Modifique o aplicativo cliente de modo que todas as referências à classe MQeMbMsgObject sejam removidas e substituídas pela utilização da classe MQePubSubMsgObject. O mapeamento entre essas duas classes e seus respectivos métodos é completamente óbvio; o MQePubSubMsgObject é mais fácil de ser utilizado para operações de publicação/assinatura porque ele contém métodos explícitos para essas funções. A configuração de quaisquer parâmetros MQ MQMD é muito semelhante em ambos os objetos, com pequenas diferenças nos nomes dos métodos e nos tipos de retorno.

Uma diferença importante é que o aplicativo cliente terá enviado todas as suas mensagens para a mesma fila (isto é, o nó de entrada MQe do intermediário); a inteligência no nó, então, determinou o destino correto. Agora, o aplicativo deve enviar todas as mensagens de fazer assinatura e cancelar assinatura à fila de controle SYSTEM.BROKER.CONTROL.QUEUE do intermediário, enquanto as mensagens de publicação devem ser enviadas à fila de entrada do fluxo de mensagens apropriado (esse fluxo conterá um nó de publicação).

Para obter um exemplo da conversão do aplicativo, compare os exemplos no *MQI Programming Guide*, no Apêndice C, "MQSeries Everyplace Nodes", com os equivalentes convertidos na seção "Gravando Aplicativos de Publicação/Assinatura do MQe" na página 9.

Nota: Há uma maneira mais simples de modificar o aplicativo, de forma que ele gere apenas mensagens da classe MQePubSubMsgObject – mas ela tem a desvantagem de perpetuar o uso da classe MQeMbMsgObject reprovada no aplicativo. Deixe a lógica inalterada até que a mensagem esteja prestes a ser enviada. Em seguida, utilize o novo método MQeMbMsgObject.toMQePubSubMsgObject que gera o objeto de mensagem equivalente na classe preferida. Esse objeto, é claro, deve ser enviado para a fila de destino apropriada de acordo com sua função de publicação/assinatura.

Da mesma forma para mensagens que estão sendo recebidas – um objeto de classe MQePubSubMsgObject pode ser convertido em seu objeto MQeMbMsgObject equivalente com o construtor MQeMbMsgObject(MQePubSubMsgObject).

3. Configure uma ponte MQe para transferir e transformar mensagens MQe da rede MQe para a rede MQ que serve de base para a topologia do intermediário. A classe do transformador da ponte deve ser configurada para ser `com.ibm.mqe.mqbridge.MQeJMSRFHTransformer`. Observe que duas filas de ponte também devem ser configuradas: uma para direcionar mensagens para a fila de controle `SYSTEM.BROKER.CONTROL.QUEUE` do intermediário; a outra para direcionar mensagens para a fila de entrada do fluxo de mensagens adequado.

Utilizando o MQeMbmMsgObject com a Ponte, Retendo um Fluxo de Mensagens Eficiente

Esta migração faz alterações mínimas no aplicativo, embora ainda utilize a ponte MQe no lugar do nó do intermediário MQe. A desvantagem é que ela requer a utilização das classes MQe obsoletas.

O trabalho a seguir é requerido:

1. Modifique o fluxo de mensagens no intermediário de modo que os nós de entrada MQ sejam utilizados no lugar dos nós de entrada MQe.
2. Modifique o aplicativo cliente de modo que as mensagens sejam enviadas através da ponte MQe. Todas as mensagens de fazer assinatura e cancelar assinatura são enviadas à fila de controle `SYSTEM.BROKER.CONTROL.QUEUE` do intermediário, enquanto as mensagens de publicação agora devem ser enviadas à fila de entrada do fluxo de mensagens apropriado (esse fluxo conterá um nó de publicação). Em todos os casos, o `com.ibm.broker.mqimqe.wrapper.MQeMbmMsgObject` continua sendo utilizado.
3. Configure uma ponte MQe para transferir e transformar mensagens MQe da rede MQe para a rede MQ que serve de base para a topologia do intermediário. A classe do transformador da ponte deve ser configurada para ser `com.ibm.mqe.mqbridge.MQeMbTransformer`. Observe que duas filas de ponte também devem ser configuradas: uma para direcionar mensagens para a fila de controle `SYSTEM.BROKER.CONTROL.QUEUE` do intermediário; a outra para direcionar mensagens para a fila de entrada do fluxo de mensagens adequado.

Utilizando o MQeMbmMsgObject com a Ponte e sem Alteração de Aplicativo

Esta migração não faz alterações no aplicativo, mas ainda utiliza a ponte MQe no lugar do nó do intermediário MQe. As desvantagens são que ela requer a utilização de classes MQe obsoletas e que ela também pode resultar no processamento excessivo dentro do fluxo de mensagens do intermediário.

O trabalho a seguir é requerido:

1. Modifique o fluxo de mensagens no intermediário de modo que os nós de entrada MQ sejam utilizados no lugar dos nós de entrada MQe. Inclua um nó de filtro de forma que (a) todas as mensagens de fazer assinatura e cancelar assinatura sejam enviadas à fila de controle `SYSTEM.BROKER.CONTROL.QUEUE` do intermediário e (b) as mensagens de publicação sejam enviadas à fila de entrada do fluxo de mensagens apropriado.

Um exemplo do ESQL necessário para o nó filtro:

```
BEGIN

    IF    InputRoot.MQRFH2.psc.Command = 'RegSub' OR
         InputRoot.MQRFH2.psc.Command = 'DeregSub' OR
         InputRoot.MQRFH2.psc.Command = 'ReqUpdate' THEN
        RETURN TRUE;
    END IF;

    IF    InputRoot.MQRFH2.psc.Command = 'Publish' OR
         InputRoot.MQRFH2.psc.Command = 'DeletePub' THEN
        RETURN FALSE;
    END IF;

    RETURN UNKNOWN;
END;
```

2. Configure uma ponte MQe para transferir e transformar mensagens MQe da rede MQe para a rede MQ que serve de base para a topologia do intermediário. A classe do transformador da ponte deve

ser configurada para ser `com.ibm.mqe.mqbridge.MQeMbTransformer`. Uma fila de ponte também só precisa ser configurada de modo que direcione todas as mensagens para a fila de entrada do fluxo de mensagens apropriado.

3. Se o modo de destino do nó foi configurado como Responder para a Fila e o gerenciador de filas de destino e a fila de destino forem diferentes da resposta para o gerenciador de filas e da resposta para a fila, você precisará configurar uma propriedade de sistema Java para impedir que os campos de destino sejam copiados na resposta para os campos no MD da mensagem MQ. A propriedade do sistema `"com.ibm.mqe.mqbridge.MQeMbTransformer.keepReplyTo"` pode ser configurada com qualquer valor. Para configurar utilizando um programa Java, inclua o argumento VM `-Dproperty=value`. Se você estiver utilizando o MQEExplorer ou MQEScript executáveis do MQEServerSupport SupportPac, será necessário criar um arquivo `.sp`. Consulte o manual *MQEServerSupport Installation Guide* para obter informações adicionais.