

# WebSphere MQ Everyplace V2.0.2



---

# Índice

## **Configurando Comunicações . . . . . 1**

Considerações sobre o Sistema Operacional . . . . .	1
Atributos . . . . .	1
Mensagens . . . . .	1
Nomes de Fila e do Gerenciador de Filas . . . . .	2
Comunicações . . . . .	2
MQeTransporter . . . . .	3
MQeChannel . . . . .	3
Adaptadores de Comunicações . . . . .	4
Tamanho do Pacote . . . . .	4

Tempo Limite Sem Bloqueio . . . . .	5
Tempo Limite . . . . .	5
Novas Tentativas . . . . .	6
Versão do HTTP . . . . .	6
Tamanho do Grupo de Adaptadores – Apenas Adaptador UDP/IP . . . . .	6
Varição de Adaptadores– Apenas para Adaptador UDP/IP . . . . .	7
Uso de Valor Configurável de UDP/IP . . . . .	7
Lista Não Processada . . . . .	7



---

## Configurando Comunicações

Antes que qualquer configuração específica de comunicações possa ser realizada, várias decisões precisam ser tomadas do ponto de vista de design de aplicativos. É importante decidir, em ordem de prioridade, o que é importante para o aplicativo. Geralmente, isso será uma transposição entre desempenho, segurança e uso de largura de banda da rede.

Quando essas decisões forem tomadas, é importante criar um aplicativo de protótipo para realizar um teste prático. Esse teste deve utilizar dados de produção através da utilização de uma ferramenta sniffer de rede para consultar informações de uso de tempo e de rede.

---

## Considerações sobre o Sistema Operacional

O MQe utiliza soquetes TCP/IP para comunicações e arquivos para manter mensagens no armazenamento persistente. Em um sistema em que um grande número de mensagens está fluindo ou um grande número de clientes está se conectando (ou ambos), os limites do sistema operacional podem ser alcançados. Para certificar-se de que isso não causará problemas, o administrador do sistema pode garantir que haja descritores de arquivo e soquetes TCP/IP suficientes disponíveis. O método para realizar isso varia entre sistemas operacionais. Consulte a documentação do sistema operacional sobre como configurar esses limites.

---

## Atributos

Os atributos fornecem a capacidade de aplicar segurança e compactação a dados na rede MQe. Os atributos são configurados em uma fila específica (disponível apenas em Java) ou em uma mensagem específica. Um MQeAttribute pode conter um de cada item a seguir em qualquer combinação:

### MQeCompressor

Reduz o número de bytes na ligação; pode impactar o desempenho.

O MQe permite que o desenvolvedor de aplicativos selecione o melhor algoritmo de compactação para os dados que estão sendo utilizados pelo aplicativo. Em geral, o compactador é melhor selecionado a partir do teste prático utilizando dados que serão gerados pelo aplicativo na produção.

### MQeCryptor

Aumenta potencialmente o número de bytes na ligação; fornece segurança.

Normalmente, durante a criptografia de dados, a quantidade de dados em bytes é aumentada; isso não é exceção no MQe.

### MQeAuthenticator

Inclui bytes na ligação.

Utilizado para fornecer a autenticação do emissor da mensagem.

---

## Mensagens

Uma carga útil de mensagem MQe é mantida em um ou mais objetos MQeFields. Estes são contêineres com um tipo, nome e valor. O MQeFields pode ser recursivo, portanto, é possível para um desenvolvedor de aplicativos criar mensagens que contenham vários objetos MQeFields, dependendo do tipo de dados requerido. O objeto MQeFields é auto-explicativo, permitindo que o MQe dispense um objeto de cabeçalho estático para descrever a carga útil dos dados. Portanto, a quantidade de informações do MQe incluídas em uma mensagem depende em parte do número de objetos MQeFields no MQeMsgObject. A

melhor abordagem para minimizar o uso da rede é colocar todos os dados em um MQeMsgObject com o menor nome apropriado para o aplicativo. Isso produz dois efeitos. Primeiro, a quantidade de dados MQe na carga útil da mensagem é minimizada e, segundo, o tempo gasto para serializar e desserializar a mensagem também é minimizado. O aspecto negativo dessa abordagem é que seu aplicativo se torna responsável por analisar os dados na mensagem, em vez de poder utilizar vários objetos MQeFields.

Sempre que dados são enviados pela rede, bytes adicionais são incluídos pelo protocolo de rede. Por exemplo, TCP inclui 20 bytes; em seguida, IP inclui 20 bytes adicionais. O desenvolvedor de aplicativos pode minimizar este código extra criando menos mensagens grandes, em vez de enviar várias mensagens pequenas. Para obter informações adicionais, consulte as várias configurações de adaptador por toda esta seção.

Se suas mensagens MQe destinarem-se a uma fila MQe, será possível utilizar a API MQe MQeMultiMsgObject, que permite enviar várias mensagens para um único objeto. Este objeto permite que o desenvolvedor de aplicativos agrupe várias mensagens em uma grande mensagem com o suporte correspondente para recuperar as mensagens. O MQeMultiMsgObject remove a responsabilidade de analisar mensagens menores incorporadas em uma mensagem grande no aplicativo.

---

## Nomes de Fila e do Gerenciador de Filas

Os nomes das filas e dos gerenciadores de filas são enviados pela rede como parte das informações de MQe incluídas na carga útil da mensagem. Os nomes para estes recursos devem ser mantidos os mais curtos possível.

---

## Comunicações

Há vários objetos instanciados como parte das comunicações de MQe que, embora não expostos ao desenvolvedor de aplicativos, podem ser configurados para ajudar a otimizar o uso da rede. Muitos desses valores de configuração podem ser configurados utilizando as mensagens de administração. O diagrama a seguir mostra os objetos de comunicações de MQe.

## Fila e Gerenciador de Filas do MQe

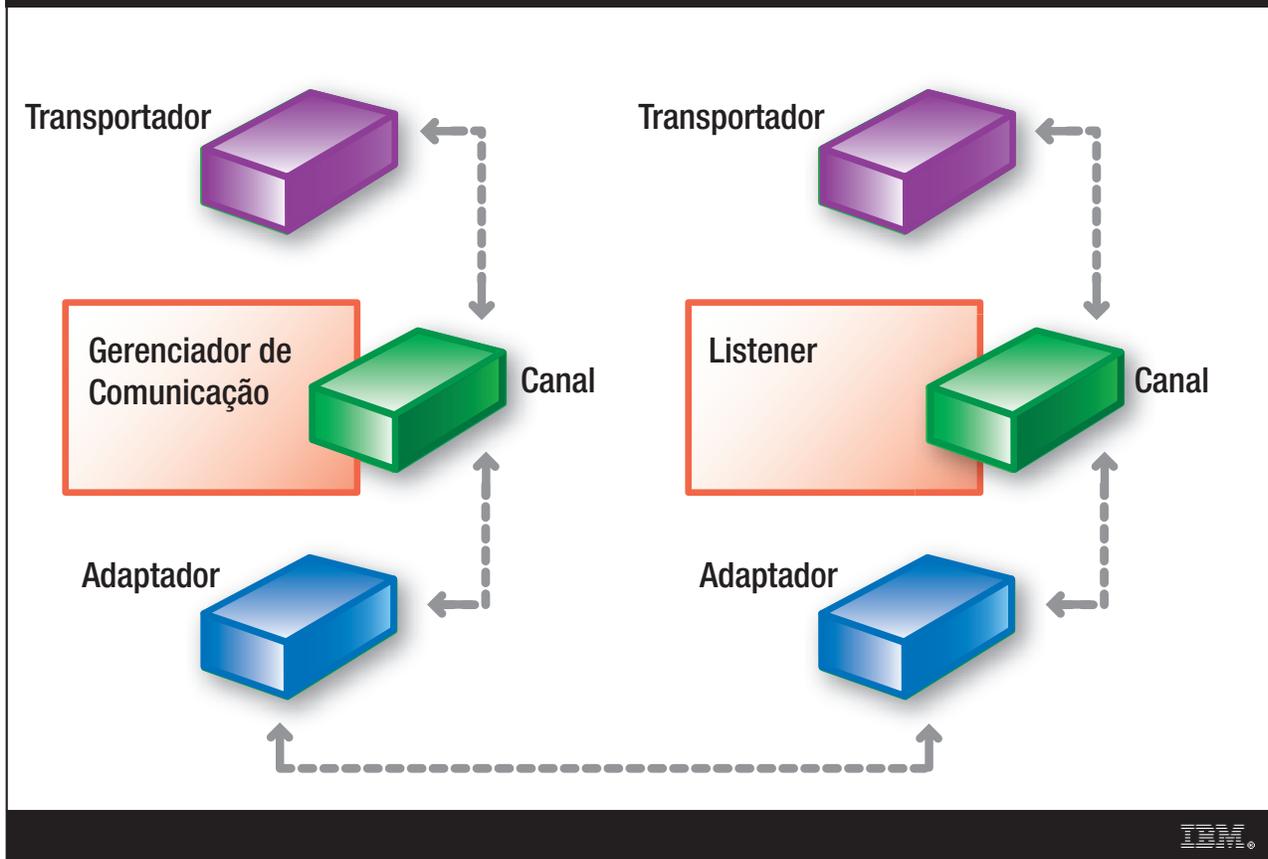


Figura 1. Fila e Gerenciador de Filas do MQe

### MQeTransporter

A classe MQeTransporter não está exposta à API. Esta classe é responsável por estabelecer um link de uma fila para outra. Há um relacionamento um-para-um entre MQeQueue e MQeTransporter. O MQeTransporter não é configurável e está incluído apenas para integralidade.

### MQeChannel

A classe MQeChannel não está exposta à API. Esta classe é responsável por estabelecer um link entre gerenciadores de filas. Conforme mostrado no diagrama, o objeto que contém uma referência a um canal aberto é alterado, dependendo se o canal é de entrada ou saída.

O valor de tempo limite do canal é fornecido em milissegundos. Um encadeamento de segundo plano é responsável pelo fechamento dos canais se eles ficarem inativos por um período de tempo configurado pelo valor de tempo limite do canal. É importante tornar os valores de tempo limite do cliente e do servidor compatíveis; de maneira ideal, o tempo limite do cliente deve ser alcançado antes do servidor. Isso permite que o cliente envie ao servidor o comando de fechamento quando o canal do cliente estiver fechando, assim, ambas as extremidades do canal serão fechadas. Se o servidor for fechado antes do cliente, tal comando não será enviado. Isso poderá levar a um dos seguintes resultados:

- O soquete do cliente poderá ser colocado no estado FINWAIT2 como resultado de estar sendo fechado quando o soquete *parceiro* não pode mais responder com a confirmação requerida. O estado pode ser mantido indefinidamente ou até o sistema operacional ser interrompido.

- O cliente deseja enviar outra mensagem e recebe um erro channel-ID do servidor. Em seguida, o cliente fecha o canal e precisa recriar um novo canal, consumindo, assim, tempo e recursos da rede.

O valor de tempo limite do canal precisa ser grande o suficiente para um cliente e um servidor concluírem mensagens de envio/recebimento mas não a ponto dos canais consumirem recursos valiosos do sistema.

Para canais de saída, o tempo limite do canal é configurado no gerenciador de filas. Isso deve ser feito assim que o gerenciador de filas for configurado. Utilize o `MQeQueueManagerAdminMsg` para configurar o tempo limite do canal no gerenciador de filas.

Para configurar o tempo limite do canal em canais de entrada, especifique o tempo limite do canal utilizando a mensagem de administração `MQeCommunicationsListenerAdminMsg` para criar ou alterar um listener.

Ao configurar o tempo limite do canal no listener ou no gerenciador de filas, o encadeamento subjacente responsável pelo tempo limite do canal não é notificado. Para finalidades de teste, pode ser conveniente parar e, em seguida, reiniciar o gerenciador de filas quando os tempos limite do canal tiverem sido enviados. Em um ambiente de produção, assim que o gerenciador de filas e o listener forem configurados, os valores serão mantidos no registro do MQe e, conseqüentemente, utilizados quando o gerenciador de filas for iniciado.

O valor padrão é 5 minutos.

## Adaptadores de Comunicações

Os adaptadores de comunicações fornecem o suporte a protocolo para envio de dados pela rede. Os adaptadores de comunicações possuem vários valores configuráveis; todos eles são configurados utilizando propriedades Java. As propriedades Java são configuradas no nível da JVM e acessadas pelo adaptador de comunicações quando ele é instanciado. O adaptador não visita essas propriedades novamente. Isso significa que, se as propriedades do nível da JVM forem alteradas, elas não terão efeito até que um novo adaptador seja criado.

Informações sobre essas propriedades também podem ser obtidas a partir do *Java Programming Reference Manual*, no tópico sobre a classe `com.ibm.mqe.adapters.MQeCommunicationsAdapter`.

## Tamanho do Pacote

Para configurar o valor do tamanho do pacote para o código-base Nativo, uma entrada é requerida no Windows Registry.

O tamanho do pacote possui o efeito mais profundo no número de pacotes enviados pela rede. Para determinar o melhor tamanho de pacote para sua rede, é requerido algum teste prático. Será necessária uma ferramenta de busca para isso.

Uma LAN Ethernet geralmente terá uma MTU (Maximum Transmission Unit) de 1500; no entanto, isso pode ser reduzido por um roteador. Para determinar a MTU efetiva de sua rede, é melhor forçar alguma fragmentação de IP; ou seja, forçar o IP a dividir os dados. Primeiramente, configure o tamanho do pacote para um valor maior que a MTU esperada de sua rede. Em seguida, envie mensagens de um gerenciador de filas MQe para outro de tamanho maior que o do pacote para que o adaptador tenha que dividir os dados. Ao olhar os resultados na ferramenta de busca, você deve ver alguma fragmentação de IP, que determinará a MTU efetiva da rede.

Depois de determinar a MTU da rede, será necessário configurar o tamanho do pacote permitido para os cabeçalhos de protocolo:

- TCP/IP com um comprimento de cabeçalho total de 40 bytes (20 bytes para IP e 20 bytes para TCP).

- UDP/IP com um comprimento de cabeçalho total de 44 bytes (12 bytes para MQe, 12 bytes para UDP e 20 bytes para IP). Ao criar os pacotes, o adaptador UDP/IP MQe levará em consideração suas próprias informações de cabeçalho, de forma que tudo o que deverá considerar são cabeçalhos UDP e IP, resultando em 32 bytes. Se a MTU da rede for 532, o tamanho do pacote deverá ser configurado como 500. Se o tamanho do pacote for menor que 30, o tamanho de pacote padrão será utilizado.
- HTTP é um pouco mais difícil de quantificar pois a quantidade de dados colocada no cabeçalho HTTP pelo MQe variará dependendo se um proxy está sendo utilizado ou se um servlet está sendo acessado e o comprimento dos dados sendo enviados. O mínimo incluído pelo MQe é de 116 bytes de dados que supõe que o comprimento se ajustará em 3 bytes para sua representação de cadeia. Bytes adicionais precisam ser permitidos se proxies ou servlets estiverem sendo utilizados.
- Ethernet possui um cabeçalho de 14 bytes.

**Java** Valor Padrão TCP/IP 4096, UDP/IP 500

Configurado utilizando a propriedade Java  
com.ibm.mqe.adapters.MQeCommunicationsAdapter.packetSize

**Nativo**

Valor Padrão 500

Configure utilizando o Registro do Windows: **HKEY\_LOCAL\_MACHINE** → **SOFTWARE** → **MQe** → **CurrentVersion** → **Communications** → **packetSize**. O valor é uma cadeia e deve representar o tamanho em bytes.

## Tempo Limite Sem Bloqueio

O tempo limite de não-bloqueio permite determinar quanto tempo um adaptador atenderá um soquete antes de verificar se o adaptador foi fechado. Esse fechamento provavelmente será emitido; como resultado de um tempo limite de canal ser alcançado ou o gerenciador de filas ser fechado. Este valor é utilizado em conjunto com o valor de Tempo Limite.

**Java** O valor padrão é 1 segundo.

Configurado utilizando a propriedade Java  
com.ibm.mqe.adapters.MQeCommunicationsAdapter.nonBlockingTimeout.

O valor deve representar o tempo em milissegundos.

**Nativo**

Não disponível.

## Tempo Limite

O adaptador tentará ler a partir de um soquete até que o valor de tempo limite seja atingido. Para permitir que o adaptador verifique um encerramento, devido ao tempo limite de canal ou ao encerramento do gerenciador de filas, o tempo limite do soquete efetivo será configurado como “Tempo Limite Sem Bloqueio”. O valor de tempo limite será, então, diminuído pelo tempo limite de não-bloqueio até que seja exaurido e, assim, utilizado como o tempo limite geral. Depois do valor de tempo limite ser alcançado, o valor de novas tentativas será diminuído conforme descrito abaixo.

O adaptador UDP/IP utiliza o valor de tempo limite para determinar a quantidade de tempo que o adaptador deve aguardar por uma confirmação. O tempo limite deve ser configurado como o tempo que um pacote gasta para fluir pela rede do inicializador até o replicador. É altamente sugerido que vários pings sejam executados com o tamanho de pacote apropriado para a rede específica. Por exemplo, se a + b for igual à quantidade de tempo levada pelo pacote para fazer a ida e volta, o MQe assume que a = b e o tempo limite deve ser definido para o valor a. O adaptador UDP/IP envia o valor do tempo limite para toda a rede ao configurar uma conversa; o valor máximo para o adaptador UDP/IP é 4294967295.

**Java** O valor padrão é TCP/IP 5 segundos, UDP/IP 10 segundos.

Configurado utilizando a propriedade Java `com.ibm.mqe.adapters.MQeCommunicationsAdapter.timeout`.

O valor deve representar o tempo em milissegundos.

#### **Nativo**

Valor Padrão 10 segundos.

Configure utilizando o Registro do Windows: **HKEY\_LOCAL\_MACHINE** → **SOFTWARE** → **MQe** → **CurrentVersion** → **Communications** → **socketTimeout**. O valor é uma cadeia e deve representar o tempo em milissegundos.

## **Novas Tentativas**

O número de novas tentativas é utilizado em conjunto com o valor “Tempo Limite” na página 5. Quando um adaptador esgotar o valor de tempo limite, o número de novas tentativas é diminuído e o adaptador, em seguida, recomeça o processo de nova tentativa. Este valor também é utilizado quando um adaptador tem um problema, por exemplo, na obtenção de um soquete assim como ao ler a partir de um soquete.

**Java** Valor Padrão 3

Configurado utilizando a propriedade Java `com.ibm.mqe.adapters.MQeCommunicationsAdapter.retries`

#### **Nativo**

Valor Padrão 3

Configure utilizando o Registro do Windows: **HKEY\_LOCAL\_MACHINE** → **SOFTWARE** → **MQe** → **CurrentVersion** → **Communications** → **retry**. O valor é uma cadeia e deve representar o número de novas tentativas.

## **Versão do HTTP**

Para uso com o `com.ibm.mqe.adapters.MqeTcpipHttpAdapter`. Para minimizar o número de soquetes criados e descartados, o que pode ser oneroso para alguns sistemas operacionais, é possível especificar a versão de HTTP a ser utilizada. Se o valor para esta propriedade for configurado como 1.1, os soquetes não serão fechados entre mensagens individuais, serão mantidos abertos durante a atividade do adaptador. O adaptador de atendimento utilizará a versão transmitida pelo cliente.

**Java** Valor Padrão 1.0

Configurado utilizando a propriedade Java `com.ibm.mqe.adapters.MQeCommunicationsAdapter.httpVersion`

O valor deve ser 1.0 ou 1.1

#### **Nativo**

Valor Padrão 1.0

Pode não estar configurado

## **Tamanho do Grupo de Adaptadores – Apenas Adaptador UDP/IP**

UDP/IP é um protocolo sem conexão e, portanto, é necessário que o MQe confirme quais pacotes foram recebidos. Para ajudar a minimizar o tráfego de rede, os pacotes fluem pela ligação em grupos. Depois de enviar vários pacotes em um grupo, o inicializador aguarda a confirmação do replicador antes de enviar um pedido para confirmação.

- Configurado utilizando a propriedade Java `com.ibm.mqe.adapters.MQeCommunicationsAdapter.groupSize`
- Valor Padrão 5
- Valor Máximo 255

## Variação de Adaptadores– Apenas para Adaptador UDP/IP

Esta variável é utilizada como uma constante em conjunto com o valor de tempo limite. O valor de variação é utilizado para fornecer um valor que reflita um possível comportamento diferente em uma rede. O valor é configurado em milissegundos.

- Configurado utilizando a propriedade Java `com.ibm.mqe.adapters.MQeCommunicationsAdapter.variance`
- Valor Padrão 1000
- Valor Máximo 4294967295

## Uso de Valor Configurável de UDP/IP

Conforme indicado anteriormente, o adaptador UDP/IP é particularmente sensível aos valores dos parâmetros configuráveis. As informações a seguir mostram como esses valores são utilizados internamente:

- Tempo para o replicador aguardar pelo próximo pacote de dados = tempo limite + variação
- Tempo para o replicador aguardar pelo próximo pacote de dados após enviar uma confirmação =  $(\text{tempo limite} + \text{variação}) \times 2$
- Tempo para o inicializador aguardar pela confirmação após enviar o último pacote no grupo =  $(\text{tempo limite} + \text{variação}) \times (\text{groupSize} + 1)$
- Tempo para o inicializador aguardar pela confirmação após enviar um pedido para confirmação =  $(\text{tempo limite} + \text{variação}) \times 2$

## Lista Não Processada

A lista não processada configurada no listener é 128. No momento, não há como alterar isso. Plataformas Windows ignorarão essa configuração se o sistema operacional não for um sistema operacional de servidor e definirão seu padrão como 5.