7.5

IBM Message Service Client for .NET

Nota

Antes de usar estas informações e o produto que elas suportam, leia as informações em “Avisos” na página 257.

Esta edição se aplica à versão 7 liberação 5 do IBM® WebSphere MQ e a todas as liberações e modificações subsequentes até que seja indicado de outra forma em novas edições.

Índice

Introdução ao Message Service Client for .NET

Message Service Client for .NET fornece uma interface de programação de aplicativos (API) chamada XMS que possui o mesmo conjunto de interfaces que o Serviço de Mensagens Java (JMS) API. O Message Service Client for .NET contém uma implementação totalmente gerenciada do XMS, que pode ser usada por qualquer linguagem compatível com .NET

O XMS suporta:

- sistema de mensagens ponto-a-ponto
- sistema de mensagens Publicação/Assinatura
- Entrega de Mensagem Síncrona
- Entrega de Mensagem Assíncrona

O aplicativo Um XMS pode trocar mensagens com os seguintes tipos de aplicativo:

- Aplicativo Um XMS
- Um aplicativo IBM WebSphere MQ classes para JMS
- Um aplicativo IBM WebSphere MQ nativo
- Um aplicativo JMS que está usando o provedor de sistemas de mensagens padrão do WebSphere

Um aplicativo XMS pode se conectar a, e usar os recursos do, qualquer um dos seguintes servidores de sistema de mensagens:

IBM WebSphere MQ gerenciador de filas
O aplicativo pode se conectar no modo de ligações ou de cliente.

WebSphere Servidor de Aplicação Serviço Integration Bus
O aplicativo pode usar uma conexão TCP/IP direta, ou pode usar HTTP sobre TCP/IP.

WebSphere Event Broker ou WebSphere Message Broker
As mensagens são transportadas entre o aplicativo e o broker usando WebSphere MQ Transporte em Tempo Real. As mensagens podem ser entregues para o aplicativo usando o WebSphere MQ Multicast Transport.

Ao conectar-se a um gerenciador de filas do IBM WebSphere MQ , o aplicativo um XMS pode usar IBM WebSphere MQ Transporte corporativo para se comunicar com o WebSphere Event Broker ou WebSphere Message Broker. Como alternativa, o aplicativo um XMS pode publicar e assinar conectando ao IBM WebSphere MQ.

Conceitos relacionados

“estilos do sistema de mensagens” na página 7
“O modelo de objeto XMS” na página 7
A API XMS é uma interface orientada a objetos. O modelo de objeto XMS é baseado no modelo de objeto JMS 1.1 .

“O modelo de mensagem XMS” na página 10
O modelo de mensagem XMS é o mesmo que o modelo de mensagem IBM WebSphere MQ classes para JMS .

Introdução ao Message Service Client for .NET

Message Service Client for .NET fornece uma interface de programação de aplicativos (API) chamada XMS que possui o mesmo conjunto de interfaces que o Serviço de Mensagens Java (JMS) API. O Message Service Client for .NET contém uma implementação totalmente gerenciada do XMS, que pode ser usada por qualquer linguagem compatível com .NET

O XMS suporta:

- sistema de mensagens ponto-a-ponto

- sistema de mensagens Publicação/Assinatura
- Entrega de Mensagem Síncrona
- Entrega de Mensagem Assíncrona

O aplicativo Um XMS pode trocar mensagens com os seguintes tipos de aplicativo:

- Aplicativo Um XMS
- Um aplicativo IBM WebSphere MQ classes para JMS
- Um aplicativo IBM WebSphere MQ nativo
- Um aplicativo JMS que está usando o provedor de sistemas de mensagens padrão do WebSphere

Um aplicativo XMS pode se conectar a, e usar os recursos do, qualquer um dos seguintes servidores de sistema de mensagens:

IBM WebSphere MQ gerenciador de filas
O aplicativo pode se conectar no modo de ligações ou de cliente.

WebSphere Servidor de Aplicação Serviço Integration Bus
O aplicativo pode usar uma conexão TCP/IP direta, ou pode usar HTTP sobre TCP/IP.

WebSphere Event Broker ou WebSphere Message Broker
As mensagens são transportadas entre o aplicativo e o broker usando WebSphere MQ Transporte em Tempo Real. As mensagens podem ser entregues para o aplicativo usando o WebSphere MQ Multicast Transport.

Ao conectar-se a um gerenciador de filas do IBM WebSphere MQ , o aplicativo um XMS pode usar IBM WebSphere MQ Transporte corporativo para se comunicar com o WebSphere Event Broker ou WebSphere Message Broker. Como alternativa, o aplicativo um XMS pode publicar e assinar conectando ao IBM WebSphere MQ.

Conceitos relacionados

“estilos do sistema de mensagens” na página 7

“O modelo de objeto XMS” na página 7
A API XMS é uma interface orientada a objetos. O modelo de objeto XMS é baseado no modelo de objeto JMS 1.1 .

“O modelo de mensagem XMS” na página 10
O modelo de mensagem XMS é o mesmo que o modelo de mensagem IBM WebSphere MQ classes para JMS .

O que Há de Novo Nesta Liberação

Há vários aprimoramentos nesta liberação do Message Service Client for .NET.

“Lendo e Gravando o Descritor de Mensagens a partir de um Aplicativo Message Service Client for .NET” na página 86
É possível acessar todos os campos do descritor de mensagens em uma mensagem IBM WebSphere MQ , exceto `StrucId` e `Versão`. O campo `BackoutCount` pode ser lido, mas não gravado. O acesso aos campos está disponível apenas ao conectar-se a um gerenciador de filas do IBM WebSphere MQ versão 6 e superior O acesso é controlado pelas propriedades de destino descritas posteriormente..

“Os aplicativos de amostra:” na página 19
Os aplicativos de amostra do XMS fornecem uma visão geral dos recursos comuns de cada API É possível usá-los para verificar a instalação e a configuração do servidor de sistema de mensagens e para verificar seus próprios aplicativos

Melhorias de desempenho
O desempenho do XMS .NET foi melhorado..

“Reconexão automática do cliente do IBM WebSphere MQ por XMS” na página 46
É possível configurar um cliente do WebSphere MQ V7.1 XMS IBM WebSphere MQ para se reconectar automaticamente após uma falha de rede, gerenciador de filas ou servidor.

“Transações XA IBM WebSphere MQ Gerenciadas por meio de XMS” na página 41
As transações XA do WebSphere MQ gerenciadas podem ser usadas por meio do XMS

GMO_CONVERT
Especificar um valor GMO_CONVERT em uma mensagem é opcional. Se um valor GMO_CONVERT for especificado, a conversão ocorrerá de acordo com o valor especificado.

estilos do sistema de mensagens

O XMS suporta os estilos ponto-a-ponto e Publicação/Assinatura do sistema de mensagens

Estilos de sistema de mensagens também são chamados de domínios de mensagens.

sistema de mensagens Ponto a ponto

Uma forma comum do sistema de mensagens ponto-a-ponto usa enfileiramento. No caso mais simples, um aplicativo envia uma mensagem para outro aplicativo identificando, implicitamente ou explicitamente, uma fila de destino. O sistema de mensagens subjacente e o sistema de enfileiramento recebem a mensagem do aplicativo de envio e roteia a mensagem para sua fila de destino. O aplicativo de recebimento pode então recuperar a mensagem a partir da fila.

Se o sistema de mensagens e o sistema de enfileiramento subjacentes contiverem o WebSphere Message Broker, o WebSphere Message Broker poderá replicar uma mensagem e rotear cópias da mensagem para filas diferentes. Como resultado, mais de um aplicativo pode receber a mensagem. WebSphere Message Broker também pode transformar uma mensagem e incluir dados a ele.

Uma característica chave do sistema de mensagens do ponto-a-ponto é que um aplicativo coloca uma mensagem em uma fila local quando ele envia uma mensagem O sistema de mensagens subjacente e o sistema de enfileiramento determinam para qual fila de destino a mensagem é enviada. O aplicativo de recebimento recupera a mensagem da fila de destino.

sistema de mensagens Publicar / assinar

No sistema de mensagens do Publicação/Assinatura , há dois tipos de aplicativo: publicador e assinante

Um *publicador* fornece informações sobre a forma de mensagens de publicação. Quando um publicador publica uma mensagem, ele especifica um tópico, que identifica o assunto das informações dentro da mensagem.

Um *assinante* é um consumidor das informações que são publicadas. Um assinante especifica os tópicos em que está interessado pela criação de assinaturas.

O sistema de publicação / assinatura recebe publicações de publicadores e assinaturas de assinantes. Ele roteia publicações para assinantes. Um assinante recebe publicações sobre apenas os tópicos para os quais ele subscreveu.

Uma característica principal do sistema de mensagens do Publicação/Assinatura é que um publicador identifica um tópico quando publica uma mensagem. Ele não identifica os assinantes. Se uma mensagem for publicada em um tópico para o qual não há assinantes, nenhum aplicativo receberá a mensagem.

Um aplicativo pode ser um publicador e um assinante.

O modelo de objeto XMS

A API XMS é uma interface orientada a objetos. O modelo de objeto XMS é baseado no modelo de objeto JMS 1.1 .

A lista a seguir resume as principais classes XMS ou tipos de objeto:

`ConnectionFactory`
Um objeto `ConnectionFactory` é encapsulado um conjunto de parâmetros para uma conexão. Um aplicativo usa um `ConnectionFactory` para criar uma conexão. Um aplicativo pode fornecer os parâmetros no tempo de execução e criar um objeto `ConnectionFactory` . Como alternativa, os parâmetros de conexão podem ser armazenados em um repositório de objetos administrados. Um

aplicativo pode recuperar um objeto a partir do repositório e criar um objeto `ConnectionFactory` a partir dele.

`Connection`
Um objeto `Connection` encapsula uma conexão ativa de um aplicativo para um servidor de sistema de mensagens. Um aplicativo usa uma conexão para criar sessões.

`Destination`
Um aplicativo envia mensagens ou recebe mensagens usando um objeto `Destination`. No domínio Publicação/Assinatura , um objeto `Destination` contém um tópico e, no domínio ponto-a-ponto , um objeto `Destination` contém uma fila. Um aplicativo pode fornecer os parâmetros para criar um objeto `Destination` no tempo de execução. Como alternativa, ele pode criar um objeto `Destination` a partir de uma definição de objeto que é armazenada no repositório de objetos administrados.

`Session`
Um objeto `Session` é um único contexto encadeado para enviar e receber mensagens. Um aplicativo usa um objeto `Session` para criar objetos `Message`, `MessageProducere` `MessageConsumer` .

`Message`
Um objeto `Message` contém o objeto `Message` que um aplicativo envia usando um objeto `MessageProducer` ou recebe usando um objeto `MessageConsumer`.

`MessageProducer`
Um objeto `MessageProducer` é usado por um aplicativo para enviar mensagens para um destino.

`MessageConsumer`
Um objeto `MessageConsumer` é usado por um aplicativo para receber mensagens enviadas para um destino.

Figura 1 na página 8 mostra esses objetos e seus relacionamentos.

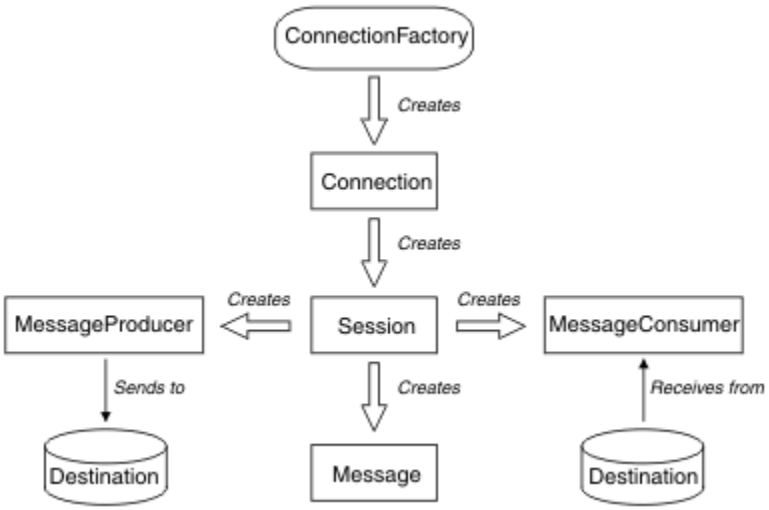

Figura 1. Os objetos do XMS e seus relacionamentos

Em .NET, as classes XMS são definidas como um conjunto de interfaces .NET . Quando você estiver codificando aplicativos do XMS .NET , precisará apenas das interfaces declaradas

O modelo de objeto XMS é baseado nas interfaces independentes de domínio descritas em *Java Message Service Specification, Versão 1.1*. Classes específicas de domínio, como `Topic`, `TopicPublishere` `TopicSubscriber`, não são fornecidas.

Atributos e Propriedades de Objetos

Um objeto XMS pode ter atributos e propriedades, que são características do objeto, que são implementados de diferentes maneiras

Atributos
Uma característica de objeto que está sempre presente e ocupa o armazenamento, mesmo que o atributo não tenha um valor. Nesse aspecto, um atributo é semelhante a um campo em uma estrutura de dados de comprimento fixo. Um recurso de diferenciação de atributos é que cada atributo possui seus próprios métodos para configurar e obter seu valor.

Propriedades
Uma propriedade de um objeto está presente e ocupa o armazenamento apenas depois que seu valor for configurado. Uma propriedade não pode ser excluída ou sua memória recuperada após seu valor ser configurado. É possível alterar seu valor. XMS fornece um conjunto de métodos genéricos para a configuração e obtenção de valores de propriedade.

Conceitos relacionados

Tipos primitivos XMS
O XMS fornece equivalentes dos oito tipos primitivos Java (byte, short, int, long, float, double, char e boolean). Isso permite a troca de mensagens entre XMS e JMS sem que os dados sejam perdidos ou corrompidos.

Conversão implícita de um valor de propriedade de um tipo de dados para outro
Quando um aplicativo obtém o valor de uma propriedade, o valor pode ser convertido por XMS em outro tipo de dados. Muitas regras controlam quais conversões são suportadas e como o XMS executa as conversões.

Referências relacionadas

Tipos de Dados para Elementos de Dados do Aplicativo
Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Objetos Administrados

Usando objetos administrados, é possível administrar as configurações de conexão usadas pelos aplicativos clientes a serem administradas a partir de um repositório central. Um aplicativo recupera definições de objeto do repositório central e as utiliza para criar objetos `ConnectionFactory` e `Destination` . Usando objetos administrados, é possível desacoplar aplicativos dos recursos usados no tempo de execução.

Por exemplo, os aplicativos XMS podem ser gravados e testados com objetos administrados que fazem referência a um conjunto de conexões e destinos em um ambiente de teste. Quando os aplicativos são implementados, os objetos administrados podem ser alterados para configurar os aplicativos para se referir a conexões e destinos no ambiente de produção.

XMS suporta dois tipos de objeto administrado:

- Um objeto `ConnectionFactory` , que é usado pelos aplicativos para fazer a conexão inicial com o servidor.
- Um objeto `Destination` , que é usado pelos aplicativos para especificar o destino para mensagens que estão sendo enviadas e a origem de mensagens que estão sendo recebidas. Um destino é um tópico ou uma fila no servidor para o qual um aplicativo se conecta.

A ferramenta de administração **`JMSAdmin`** é fornecida com IBM WebSphere MQ. Ele é usado para criar e gerenciar objetos administrados para o em um repositório central de objetos administrado

Os objetos administrados no repositório podem ser usados pelos aplicativos IBM WebSphere MQ classes para JMS e XMS . Aplicativos XMS podem usar os objetos `ConnectionFactory` e `Destination` para se conectar a um IBM WebSphere MQ gerenciador de filas. Um administrador pode alterar as definições de objeto mantidas no repositório sem afetar o código do aplicativo.

O diagrama a seguir mostra como um aplicativo XMS geralmente usa objetos administrados.

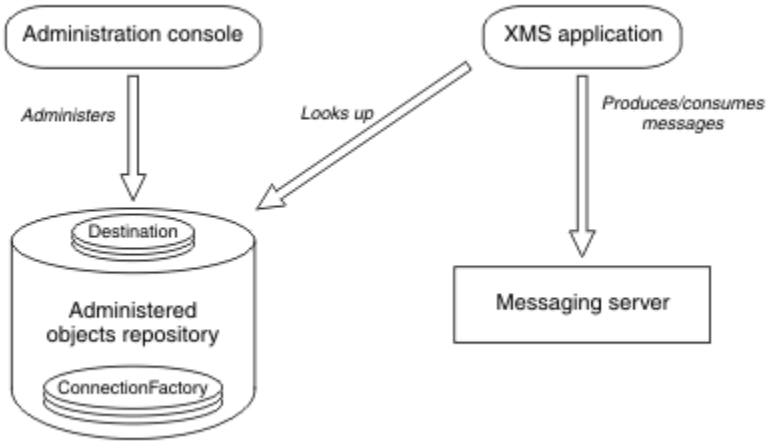

Figura 2. Uso típico de objetos administrados por um aplicativo XMS

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados

O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Tarefas relacionadas

Criando Objetos Administrados

As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

O modelo de mensagem XMS

O modelo de mensagem XMS é o mesmo que o modelo de mensagem IBM WebSphere MQ classes para JMS .

Em particular, XMS implementa os campos de cabeçalho da mesma mensagem e as propriedades de mensagem que o IBM WebSphere MQ classes para JMS implementa:

- Campos de cabeçalho do JMS Esses campos possuem nomes que se iniciam com o prefixo JMS.
- Propriedades definidas pelo JMS Esses campos possuem propriedades cujos nomes se iniciam com o prefixo JMSX.
- Propriedades definidas IBM . Esses campos possuem propriedades cujos nomes começam com o prefixo JMS_IBM_.

Como resultado, os aplicativos XMS podem trocar mensagens com aplicativos IBM WebSphere MQ classes para JMS . Em cada mensagem, alguns dos campos de cabeçalho e propriedades são configurados pelo aplicativo e outros são configurados por XMS ou IBM WebSphere MQ classes para JMS. Alguns dos campos configurados por XMS ou IBM WebSphere MQ classes para JMS são configurados quando a mensagem é enviada e outros quando ela é recebida. Os campos de cabeçalho e propriedades são propagados com uma mensagem por meio de um servidor de sistema de mensagens quando apropriado. Elas são disponibilizadas para qualquer aplicativo que receba a mensagem.

Evitando que os aplicativos usem a versão mais recente do XMS

Por padrão, quando uma versão mais recente do XMS é instalada, os aplicativos que usam a versão anterior alternam automaticamente para a versão mais recente sem precisar recompilar.

Sobre esta tarefa

O recurso de coexistência de várias versões assegura que a instalação de uma versão mais recente do XMS não sobrescreva a versão XMS anterior. Em vez disso, várias instâncias de montagens de XMS .NET semelhantes coexistem no Global Assembly Cache (GAC), mas possuem números de versão diferentes. Internamente, o GAC usa um arquivo de políticas para rotear as chamadas de aplicativo para a versão mais recente de XMS. Os aplicativos são executados sem uma necessidade de recompilação e podem usar novos recursos disponíveis na versão XMS .NET mais recente.

No entanto, se um aplicativo for necessário para usar a versão mais antiga do XMS , configure o atributo `publisherpolicy` como no no arquivo de configuração do aplicativo

Nota: Um arquivo de configuração de aplicativo é um arquivo com um nome que consiste no nome do programa executável para o qual o arquivo está relacionado, com o sufixo .config. Por exemplo, o arquivo de configuração de aplicativo para text.exe teria o nome text.exe.config.

A qualquer momento, no entanto, todos os aplicativos de um sistema usam a mesma versão do XMS .NET.

Configurando o Ambiente do Servidor de Mensagens

Este seçãocapítulo descreve como configurar o ambiente do servidor de sistema de mensagens para permitir que aplicativos XMS se conectem a um servidor.

Para aplicativos que se conectam a um gerenciador de filas do IBM WebSphere MQ , o cliente IBM WebSphere MQ (ou gerenciador de filas para o modo de ligações) é necessário.

Atualmente, não há pré-requisitos para aplicativos que usam uma conexão em tempo real com um broker.

Você deve configurar o ambiente do servidor de sistema de mensagens antes de executar quaisquer aplicativos XMS , incluindo os aplicativos de amostra fornecidos com XMS.

Esse seçãocapítulo contém o seguinte tópicosseções:

Tarefas relacionadas

Instalando Message Service Client for .NET usando o assistente de instalação
A instalação usa um instalador MSI InstallShield X/Windows. Duas opções de configuração estão disponíveis, para que seja possível escolher uma instalação completa ou customizada.

Configurando o gerenciador e o broker para um aplicativo que se conecta a um gerenciador de filas do IBM WebSphere MQ

Esta seção assume que você está usando o IBM WebSphere MQ version 7.0 Para poder executar um aplicativo que se conecta a um IBM WebSphere MQ gerenciador de filas, deve-se configurar o gerenciador de filas. Para um aplicativo Publicação/Assinatura , alguma configuração adicional será necessária se você estiver usando a interface de publicação / assinatura enfileirado.

Antes de começar

XMS funciona com o WebSphere Message Broker versão 6.1 ou anterior.

Antes de iniciar esta tarefa, você executa as seguintes etapas:

- Certifique-se de que seu aplicativo tenha acesso a um gerenciador de filas em execução.
- Se seu aplicativo for um aplicativo de publicação / assinatura e usar a interface de publicação / assinatura enfileirada, certifique-se de que o atributo "PSMODE" esteja configurado como "ENABLED" no gerenciador de filas.
- Certifique-se de que seu aplicativo usa um connection factory cujas propriedades são configuradas corretamente para se conectar ao gerenciador de filas. Se o seu aplicativo for um aplicativo de publicação / assinatura, certifique-se de que as propriedades apropriadas do connection factory estejam configuradas para o uso do broker. Para obter mais informações sobre as propriedades de um connection factory, “Propriedades de ConnectionFactory” na página 185

Sobre esta tarefa

Você configura o gerenciador de filas e o broker para executar aplicativos XMS da mesma maneira que você configura o gerenciador de filas e a interface de publicação / assinatura enfileirada para executar aplicativos JMS do WebSphere MQ As etapas a seguir resumem o que você precisa fazer.

Procedimento

1. No gerenciador de filas, crie as filas de que seu aplicativo precisa

   Para obter informações sobre como criar filas, consulte o tópico *Definindo filas* na *WebSphere MQ documentação do produto*.

   Se o seu aplicativo for um aplicativo Publicação/Assinatura e usar a interface de publicação / assinatura Enfileirada que precisa de acesso às filas do sistema IBM WebSphere MQ classes para JMS , aguarde até a Etapa 4a antes de criar as filas
2. Conceda ao ID do usuário associado ao aplicativo a autoridade para se conectar ao gerenciador de filase a autoridade apropriada para acessar as filas.

   Para obter informações sobre autorização, consulte a seção *Segurança* da *IBM WebSphere MQ documentação do produto*.. Se seu aplicativo se conectar ao gerenciador de filas no modo cliente, consulte também o tópico *Clientes* na *IBM WebSphere MQ documentação do produto*.
3. Se seu aplicativo se conectar ao gerenciador de filas no modo cliente, certifique-se de que um canal de conexão do servidor esteja definido no gerenciador de filas e que um listener seja iniciado.

   Para obter informações sobre como fazer isso, consulte o tópico *Clientes* na *IBM WebSphere MQ documentação do produto*..

   Não é necessário executar essa etapa para cada aplicativo que se conecta ao gerenciador de filas Uma definição de canal de conexão do servidor e um listener podem suportar todos os aplicativos que se conectam no modo cliente.
4. Se seu aplicativo for um aplicativo Publicação/Assinatura e usar a interface de publicação / assinatura enfileirada, execute as etapas a seguir.
   a) No gerenciador de filas, crie as filas do sistema IBM WebSphere MQ classes para JMS executando o script de comandos MQSC fornecidos com IBM WebSphere MQ. Certifique-se de que o ID do usuário associado ao Message Broker do WebSphere tenha a autoridade para acessar as filas.

      Para obter informações sobre onde localizar o script e como executá-lo, consulte o tópico *Usando Java™* na documentação do produto *WebSphere MQ*..

      Execute esta etapa apenas uma vez para o gerenciador de filas O mesmo conjunto de filas do sistema IBM WebSphere MQ classes para JMS pode suportar todos os aplicativos XMS e IBM WebSphere MQ classes para JMS que se conectam ao gerenciador de filas..
   b) Conceda ao ID do usuário associado ao seu aplicativo a autoridade para acessar as filas do sistema IBM WebSphere MQ classes para JMS .

Para obter informações sobre quais autoridades o ID do usuário precisa, consulte o tópico *Usando Java* na *IBM WebSphere MQ documentação do produto*.

c) Para um broker de WebSphere Event Broker ou WebSphere Message Broker, crie e implemente um fluxo de mensagens para atender a fila na qual os aplicativos enviam mensagens que eles publicam.

O fluxo de mensagens básico inclui um nó de processamento de mensagens MQInput para ler as mensagens publicadas e um nó de processamento de mensagem de publicação para publicar as mensagens.

Para obter informações sobre como criar e implementar um fluxo de mensagens, consulte a documentação do produto WebSphere Event Broker ou WebSphere Message Broker .

Você não precisa executar esta etapa se um fluxo de mensagens adequado já estiver implementado no intermediário.

Resultados

Agora é possível iniciar seu aplicativo.

Tarefas relacionadas

Configurando um broker para um aplicativo que usa uma conexão em tempo real com um broker
Antes de poder executar um aplicativo que usa uma conexão em tempo real com um broker, você deve configurar esse broker.

Configurando o barramento de integração de serviços para um aplicativo que se conecta a um WebSphere Serviço Integration Bus
Antes de poder executar um aplicativo que se conecta a um WebSphere Serviço Integration Bus, deve-se configurar o barramento de integração de serviços da mesma maneira que o barramento de integração de serviços para executar aplicativos JMS que usam o provedor de sistemas de mensagens padrão.

Instalando Message Service Client for .NET usando o assistente de instalação
A instalação usa um instalador MSI InstallShield X/Windows. Duas opções de configuração estão disponíveis, para que seja possível escolher uma instalação completa ou customizada.

Referências relacionadas

Pré-requisitos para aplicativos XMS se conectando ao IBM WebSphere MQ
Alguns pré-requisitos se aplicam se seu aplicativo XMS se conectar ao IBM WebSphere MQ.

Configurando um broker para um aplicativo que usa uma conexão em tempo real com um broker

Antes de poder executar um aplicativo que usa uma conexão em tempo real com um broker, você deve configurar esse broker.

Antes de começar

XMS funciona com o WebSphere Message Broker versão 6.1 ou anterior.

Antes de iniciar esta tarefa, você executa as seguintes etapas:

- Certifique-se de que seu aplicativo tenha acesso a um broker que está em execução.
- Certifique-se de que seu aplicativo usa um connection factory cujas propriedades são configuradas apropriadamente para uma conexão em tempo real com um broker. Para obter mais informações sobre as propriedades de um connection factory, consulte “Propriedades de ConnectionFactory” na página 185

Sobre esta tarefa

Você configura um broker para executar aplicativos XMS da mesma maneira que você configura um broker para executar aplicativos IBM WebSphere MQ classes para JMS . As etapas a seguir resumem o que você precisa fazer, mas para obter mais detalhes, consulte a documentação do produto WebSphere Event Brokerou WebSphere Message Broker :

Procedimento

1. Crie e implemente um fluxo de mensagens para ler mensagens a partir da porta TCP/IP na qual um broker está atendendo e publique as mensagens.

   Você pode fazer isso de uma das seguintes maneiras:

   - Crie um fluxo de mensagens que contenha um nó de processamento de mensagens **Real-timeOptimizedFlow** .
   - Crie um fluxo de mensagens que contenha um nó de processamento de mensagens **Real-timeInput** e um nó de processamento de mensagem de publicação.

   Você deve configurar o nó **Real-timeOptimizedFlow** ou **Real-timeInput** para atender na porta usada para conexões em tempo real. Em XMS, o número da porta padrão para conexões em tempo real é 1506.

   Você não precisa executar esta etapa se um fluxo de mensagens adequado já estiver implementado no intermediário.
2. Se precisar que as mensagens sejam entregues ao seu aplicativo usando o WebSphere MQ Multicast Transport, configure o broker para ativar o multicast. Configure os tópicos que devem ser ativados multicast, especificando uma qualidade de serviço confiável para esses tópicos que requerem multicast confiável.
3. Se o seu aplicativo fornecer um ID de usuário e uma senha quando ele se conectar ao intermediário e você desejar que o intermediário autentique seu aplicativo utilizando essas informações, configure o servidor de nome de usuário e o intermediário para autenticação de senha simples de telnet.

Resultados

Agora é possível iniciar seu aplicativo.

Tarefas relacionadas

Configurando o gerenciador e o broker para um aplicativo que se conecta a um gerenciador de filas do IBM WebSphere MQ
Esta seção assume que você está usando o IBM WebSphere MQ version 7.0 Para poder executar um aplicativo que se conecta a um IBM WebSphere MQ gerenciador de filas, deve-se configurar o gerenciador de filas. Para um aplicativo Publicação/Assinatura , alguma configuração adicional será necessária se você estiver usando a interface de publicação / assinatura enfileirado.

Configurando o barramento de integração de serviços para um aplicativo que se conecta a um WebSphere Serviço Integration Bus
Antes de poder executar um aplicativo que se conecta a um WebSphere Serviço Integration Bus, deve-se configurar o barramento de integração de serviços da mesma maneira que o barramento de integração de serviços para executar aplicativos JMS que usam o provedor de sistemas de mensagens padrão.

Instalando Message Service Client for .NET usando o assistente de instalação
A instalação usa um instalador MSI InstallShield X/Windows. Duas opções de configuração estão disponíveis, para que seja possível escolher uma instalação completa ou customizada.

Referências relacionadas

Pré-requisitos para aplicativos XMS se conectando ao IBM WebSphere MQ
Alguns pré-requisitos se aplicam se seu aplicativo XMS se conectar ao IBM WebSphere MQ.

Configurando o barramento de integração de serviços para um aplicativo que se conecta a um WebSphere Serviço Integration Bus

Antes de poder executar um aplicativo que se conecta a um WebSphere Serviço Integration Bus, deve-se configurar o barramento de integração de serviços da mesma maneira que o barramento de integração de serviços para executar aplicativos JMS que usam o provedor de sistemas de mensagens padrão.

Antes de começar

Antes de iniciar esta tarefa, você deve executar as seguintes etapas:

- Certifique-se de que um barramento do sistema de mensagens seja criado e que o servidor seja incluído no barramento como um membro do barramento.
- Certifique-se de que seu aplicativo tenha acesso a um barramento de integração de serviços que contenha pelo menos um mecanismo do sistema de mensagens em execução.
- Se a operação HTTP, for necessária, um canal de transporte de entrada do mecanismo do sistema de mensagens HTTP deverá ser definido. Por padrão, os canais para SSL e TCP são definidos durante a instalação do servidor.
- Certifique-se de que seu aplicativo use um connection factory cujas propriedades estão configuradas apropriadamente para se conectar ao barramento de integração de serviços usando um servidor de autoinicialização As informações mínimas necessárias são:
  - O terminal do provedor, que descreve o local e o protocolo a serem usados ao negociar uma conexão com o servidor de sistema de mensagens (ou seja, por meio do servidor de auto-inicialização). Em sua forma mais simples, para um servidor instalado com as configurações padrão, o terminal de fornecimento pode ser configurado como o nome do host do servidor
  - O nome do barramento por meio do qual as mensagens são enviadas.

  Para obter mais informações sobre as propriedades de um connection factory, consulte "Propriedades de ConnectionFactory" na página 185

Sobre esta tarefa

Quaisquer espaços de fila ou de tópico que você precisa devem ser definidos. Por padrão, um espaço de tópico chamado Default.Topic.Space é definido durante a instalação do servidor, mas, se você precisar de espaços de tópico adicionais, deverá criar esses espaços de tópico você mesmo. Você não precisa predefinir tópicos individuais dentro de um espaço de tópico, uma vez que o servidor instancia esses tópicos individuais dinamicamente conforme necessário.

As etapas a seguir resumem o que você precisa fazer.

Procedimento

1. Crie as filas que seu aplicativo precisa para o sistema de mensagens ponto-a-ponto .
2. Crie quaisquer espaços de tópico adicionais que seu aplicativo precisa para o sistema de mensagens do Publicação/Assinatura

Resultados

Agora é possível iniciar seu aplicativo.

Tarefas relacionadas

Configurando o gerenciador e o broker para um aplicativo que se conecta a um gerenciador de filas do IBM WebSphere MQ
Esta seção assume que você está usando o IBM WebSphere MQ version 7.0 Para poder executar um aplicativo que se conecta a um IBM WebSphere MQ gerenciador de filas, deve-se configurar o gerenciador de filas. Para um aplicativo Publicação/Assinatura , alguma configuração adicional será necessária se você estiver usando a interface de publicação / assinatura enfileirado.

Configurando um broker para um aplicativo que usa uma conexão em tempo real com um broker
Antes de poder executar um aplicativo que usa uma conexão em tempo real com um broker, você deve configurar esse broker.

Instalando Message Service Client for .NET usando o assistente de instalação
A instalação usa um instalador MSI InstallShield X/Windows. Duas opções de configuração estão disponíveis, para que seja possível escolher uma instalação completa ou customizada.

Referências relacionadas

Pré-requisitos para aplicativos XMS se conectando ao IBM WebSphere MQ
Alguns pré-requisitos se aplicam se seu aplicativo XMS se conectar ao IBM WebSphere MQ.

Instalando Message Service Client for .NET usando o assistente de instalação

A instalação usa um instalador MSI InstallShield X/Windows. Duas opções de configuração estão disponíveis, para que seja possível escolher uma instalação completa ou customizada.

Sobre esta tarefa

Para instalar o Message Service Client for .NET em Windows, siga este procedimento

Procedimento

1. Se estiver instalando a partir de um SupportPac , conclua as etapas a seguir, caso contrário, continue diretamente na etapa “2” na página 16.
   a) On Windows, efetue logon como um administrador.
   b) Execute o instalador do dotNETClientsetup.exe.
2. Aguarde a abertura do assistente de instalação e exiba a mensagem a seguir:

   ```
   Welcome to IBM Message Service Client for .NET installation wizard
   ```

   Clique em **Avançar**.

   O assistente pode solicitar a leitura do contrato de licença.
3. Se você for solicitado a ler o contrato de licença e aceitar os termos do contrato de licença, clique em **Eu aceito os termos no contrato de licença**e, em seguida, clique em **Avançar**.

   O assistente de instalação solicita que você escolha o tipo de configuração que melhor se adapte às suas necessidades
4. Selecione o tipo de configuração necessário:
   - Para instalar todos os recursos do programa e instalá-los no diretório de instalação padrão, clique em **Concluir**..
   - Para escolher quais recursos deseja instalar e especificar onde eles estão instalados, clique em **Customizado**.
5. Clique em **Avançar**.

   Se você selecionar a opção de instalação completa, o assistente de instalação exibe uma mensagem de que está pronto para iniciar a instalação, conforme descrito na etapa “8” na página 16. Se você selecionar a opção de instalação customizada, o assistente de instalação solicitará que você selecione os recursos que deseja instalar e você deverá concluir a etapa “6” na página 16 e a etapa “7” na página 16 antes de avançar para a etapa “8” na página 16
6. Apenas para uma instalação customizada, clique em um ícone na lista de recursos para especificar quaisquer mudanças em como você deseja que os recursos do Message Service Client for .NET sejam instalados Se você não quiser instalar o Message Service Client for .NET no diretório sugerido, escolha outro diretório.

   Se você optar por instalar o Message Service Client for .NET em um diretório que não existe atualmente, o assistente de instalação criará o diretório para você.

   Se desejar desenvolver aplicativos XMS , assegure-se de que o recurso **Ferramentas e Amostras de Desenvolvimento** esteja selecionado. Esse recurso fornece os aplicativos de amostra e as bibliotecas e quaisquer outros arquivos necessários para compilar aplicativos .NET . Se você não selecionar esse recurso, apenas os arquivos necessários para executar aplicativos XMS serão instalados..
7. Se você estiver usando a opção de instalação customizada, clique em **Avançar** após selecionar as opções necessárias conforme descrito na etapa “6” na página 16.

   O assistente de instalação exibe uma mensagem de que ele está pronto para iniciar a instalação
8. Clique em **Instalar** para iniciar a instalação.

O assistente de instalação exibe uma barra mostrando o progresso da instalação.. Aguarde a conclusão da barra de progresso. Quando a instalação for concluída com êxito, a janela exibirá a seguinte mensagem:

```
The installation wizard has successfully installed IBM Message Service Client
for .NET. Click Finish to exit the wizard.
```

9. Clique em **Concluir** para fechar o assistente de instalação.

Resultados

Você instalou com sucesso o Message Service Client for .NET, que está pronto para ser usado

Como proceder a seguir

Antes de executar quaisquer aplicativos XMS , incluindo os aplicativos de amostra fornecidos com o XMS, você deve configurar o ambiente do servidor de sistema de mensagens, para obter detalhes, consulte: “Configurando o Ambiente do Servidor de Mensagens” na página 11.

Conceitos relacionados

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Configurando o Ambiente do Servidor de Mensagens
Este seçãocapítulo descreve como configurar o ambiente do servidor de sistema de mensagens para permitir que aplicativos XMS se conectem a um servidor.

Usando os aplicativos de amostra XMS
Use os aplicativos de amostra fornecidos com o XMS para verificar sua instalação e configuração do servidor de sistema de mensagens e para ajudá-lo a construir seus aplicativos. As amostras fornecem uma visão geral dos recursos comuns de cada API.

Tarefas relacionadas

Configurando o gerenciador e o broker para um aplicativo que se conecta a um gerenciador de filas do IBM WebSphere MQ
Esta seção assume que você está usando o IBM WebSphere MQ version 7.0 Para poder executar um aplicativo que se conecta a um IBM WebSphere MQ gerenciador de filas, deve-se configurar o gerenciador de filas. Para um aplicativo Publicação/Assinatura , alguma configuração adicional será necessária se você estiver usando a interface de publicação / assinatura enfileirado.

Configurando um broker para um aplicativo que usa uma conexão em tempo real com um broker
Antes de poder executar um aplicativo que usa uma conexão em tempo real com um broker, você deve configurar esse broker.

Configurando o barramento de integração de serviços para um aplicativo que se conecta a um WebSphere Serviço Integration Bus
Antes de poder executar um aplicativo que se conecta a um WebSphere Serviço Integration Bus, deve-se configurar o barramento de integração de serviços da mesma maneira que o barramento de integração de serviços para executar aplicativos JMS que usam o provedor de sistemas de mensagens padrão.

Referências relacionadas

Pré-requisitos para aplicativos XMS se conectando ao IBM WebSphere MQ
Alguns pré-requisitos se aplicam se seu aplicativo XMS se conectar ao IBM WebSphere MQ.

Pré-requisitos para aplicativos XMS se conectando ao IBM WebSphere MQ

Alguns pré-requisitos se aplicam se seu aplicativo XMS se conectar ao IBM WebSphere MQ.

Para aplicativos que se conectam a um gerenciador de filas do IBM WebSphere MQ , deve-se instalar as bibliotecas do cliente IBM WebSphere MQ apropriadas na máquina usada para executar o aplicativo XMS . Essas bibliotecas são pré-instaladas nas máquinas com um gerenciador de fila local

Para clienteXMS para .NET, use as bibliotecas do cliente enviadas com IBM WebSphere MQ Versão 7.0.1.0 ou posterior. Essas são as classes *IBM WebSphere MQ para .NET*. Eles ativam conexões do modo cliente para IBM WebSphere MQ Versão 7.0, IBM WebSphere MQ Versão 6.0e IBM WebSphere MQ Versão 5.3 gerenciadores de filas e conexões do modo de ligação para um gerenciador de fila local, se ele também for Versão 7.0.1.0 ou posterior

O Pacote Redistribuível do Microsoft .NET Framework Versão 2.0 deve ser instalado no computador no qual o XMS deve ser instalado Se esse pacote não estiver disponível, a instalação do XMS falhará Em seguida, é necessário sair do procedimento de instalação, instalar o Microsoft .NET Framework Versão 2.0 Pacote Redistribuível em seu computador e executar novamente o procedimento de instalação

No site de download da Microsoft, é necessário procurar dotnetfx.exe para Microsoft .NET Framework Versão 2.0 Pacote Redistribuível (x86) e NetFx64.exe para Microsoft .NET Framework versão 2.0 Pacote Redistribuível (x64), o que for aplicável.

Conceitos relacionados

"Configurando o Ambiente do Servidor de Mensagens" na página 11
Este seçãocapítulo descreve como configurar o ambiente do servidor de sistema de mensagens para permitir que aplicativos XMS se conectem a um servidor.

Tarefas relacionadas

Configurando o gerenciador e o broker para um aplicativo que se conecta a um gerenciador de filas do IBM WebSphere MQ
Esta seção assume que você está usando o IBM WebSphere MQ version 7.0 Para poder executar um aplicativo que se conecta a um IBM WebSphere MQ gerenciador de filas, deve-se configurar o gerenciador de filas. Para um aplicativo Publicação/Assinatura , alguma configuração adicional será necessária se você estiver usando a interface de publicação / assinatura enfileirado.

Configurando um broker para um aplicativo que usa uma conexão em tempo real com um broker
Antes de poder executar um aplicativo que usa uma conexão em tempo real com um broker, você deve configurar esse broker.

Configurando o barramento de integração de serviços para um aplicativo que se conecta a um WebSphere Serviço Integration Bus
Antes de poder executar um aplicativo que se conecta a um WebSphere Serviço Integration Bus, deve-se configurar o barramento de integração de serviços da mesma maneira que o barramento de integração de serviços para executar aplicativos JMS que usam o provedor de sistemas de mensagens padrão.

Instalando Message Service Client for .NET usando o assistente de instalação
A instalação usa um instalador MSI InstallShield X/Windows. Duas opções de configuração estão disponíveis, para que seja possível escolher uma instalação completa ou customizada.

Usando os aplicativos de amostra XMS

Use os aplicativos de amostra fornecidos com o XMS para verificar sua instalação e configuração do servidor de sistema de mensagens e para ajudá-lo a construir seus aplicativos. As amostras fornecem uma visão geral dos recursos comuns de cada API.

Conceitos relacionados

"Os aplicativos de amostra:" na página 19
Os aplicativos de amostra fornecem uma visão geral dos recursos comuns de cada API. É possível usá-los para verificar a sua instalação e o servidor de sistema de mensagens configurado e para ajudar a construir os seus próprios aplicativos.

Tarefas relacionadas

Instalando Message Service Client for .NET usando o assistente de instalação
A instalação usa um instalador MSI InstallShield X/Windows. Duas opções de configuração estão disponíveis, para que seja possível escolher uma instalação completa ou customizada.

"Executando os Aplicativos de Amostra" na página 20
É possível executar os aplicativos de amostra do .NET interativamente no modo simples ou avançado ou não interativamente usando arquivos de resposta gerados automaticamente ou customizados.

"Construindo os aplicativos de amostra .NET" na página 21
Quando você constrói um aplicativo .NET de amostra, uma versão executável de sua amostra escolhida é criada.

Os aplicativos de amostra:

Os aplicativos de amostra fornecem uma visão geral dos recursos comuns de cada API. É possível usá-los para verificar a sua instalação e o servidor de sistema de mensagens configurado e para ajudar a construir os seus próprios aplicativos.

Se você precisar de ajuda para criar seus próprios aplicativos, será possível usar os aplicativos de amostra como um ponto de início. Tanto a origem quanto uma versão compilada são fornecidas para cada aplicativo. Revise o código-fonte de amostra e identifique as etapas principais para criar cada objeto necessário para seu aplicativo (ConnectionFactory, Conexão, Sessão, Destino, e um Produtor, ou um Consumidor, ou ambos) e para configurar quaisquer propriedades específicas que sejam necessárias para especificar como você deseja que seu aplicativo funcione. Para obter mais informações, consulte "Gravando aplicativos do XMS" na página 22. As amostras estão sujeitas a mudanças em liberações futuras de XMS.

A tabela a seguir mostra os três conjuntos de aplicativos de amostra (um para cada API) que são fornecidos com XMS.

Nome da amostra	Descrição
SampleConsumerCS	Um aplicativo consumidor de mensagens que obtém mensagens de uma fila ou assina um tópico.
SampleProducerCS	Um aplicativo de produtor de mensagem que produz mensagens para uma fila ou em um tópico.
SampleConfigCS	Um aplicativo de configuração que pode ser usado para criar um repositório de objeto administrado que se baseia em arquivo. O aplicativo contém um connection factory e um destino para suas configurações de conexão específicas. Esse repositório de objeto administrado pode, então, ser usado com cada um dos aplicativos de consumidor e de produtor de amostra.

As amostras que suportam as mesmas funções nas várias APIs possuem diferenças sintácticas.

- Os aplicativos de consumidor e de produtor de mensagens de amostra suportam as seguintes funções:
  - Conexões com IBM WebSphere MQ, WebSphere Event Broker, WebSphere Message Broker (usando uma conexão em tempo real com um broker) e um WebSphere Serviço Integration Bus
  - Consultas do Repositório de Objeto Administrado Usando a Interface de Contexto Inicial
  - Conexões com filas (IBM WebSphere MQ e WebSphere Serviço Integration Bus) e tópicos (IBM WebSphere MQ, conexão em tempo real para um broker e WebSphere Serviço Integration Bus)
  - Mensagens de base, de byte, de mapa, de objeto, de fluxo e de texto
- O aplicativo consumidor de mensagens de amostra suporta modos de recebimento síncronos e assíncronos e instruções SQL Selector.
- O aplicativo do produtor de mensagem de amostra suporta modos de entrega persistentes e não persistentes.

Modos de funcionamento

As amostras podem operar em um dos dois modos:

Modo Simples
É possível executar as amostras com a entrada mínima do usuário.

Modo Avançado
É possível customizar mais finamente a maneira na qual as amostras operam.

Todas as amostras são compatíveis e podem, portanto, operar por meio de linguagens.

Onde Localizar as Amostras

Para descobrir onde os aplicativos de amostra para Message Service Client for .NET estão instalados, consulte *Diretórios instalados em Windows (.NET)* na documentação do produto on-line do IBM IBM WebSphere MQ .

Conceitos relacionados

“Construindo seus próprios aplicativos” na página 45
Você constrói seus próprios aplicativos, como você constrói os aplicativos de amostra.

Tarefas relacionadas

Executando os Aplicativos de Amostra
É possível executar os aplicativos de amostra do .NET interativamente no modo simples ou avançado ou não interativamente usando arquivos de resposta gerados automaticamente ou customizados.

Construindo os aplicativos de amostra .NET
Quando você constrói um aplicativo .NET de amostra, uma versão executável de sua amostra escolhida é criada.

“Executando os Aplicativos de Amostra” na página 20
É possível executar os aplicativos de amostra do .NET interativamente no modo simples ou avançado ou não interativamente usando arquivos de resposta gerados automaticamente ou customizados.

“Construindo os aplicativos de amostra .NET” na página 21
Quando você constrói um aplicativo .NET de amostra, uma versão executável de sua amostra escolhida é criada.

Executando os Aplicativos de Amostra

É possível executar os aplicativos de amostra do .NET interativamente no modo simples ou avançado ou não interativamente usando arquivos de resposta gerados automaticamente ou customizados.

Antes de começar

Antes de executar qualquer um dos aplicativos de amostra fornecidos, você deve primeiro configurar o ambiente do servidor de sistema de mensagens para que os aplicativos possam se conectar a um servidor. Consulte “Configurando o Ambiente do Servidor de Mensagens” na página 11.

Procedimento

Para executar um aplicativo de amostra .NET, conclua as etapas a seguir:

Sugestão: Quando você estiver executando um aplicativo de amostra, digite ? em qualquer momento para obter ajuda sobre o que fazer em seguida.

1. Selecione o modo no qual você deseja executar o aplicativo de amostra.

   Digite `Advanced` ou `Simple`.
2. Responda às perguntas.

   Para selecionar o valor padrão, que é mostrado entre os colchetes no final da pergunta, pressione Enter. Para selecionar um valor diferente, digite o valor apropriado e pressione Enter.

   Aqui está uma pergunta de exemplo:

   ```
   Enter connection type [wpm]:
   ```

Nesse caso, o valor padrão é `wpm` (conexão com um WebSphere Serviço Integration Bus).

Resultados

Quando você executa os aplicativos de amostra, os arquivos de resposta são gerados automaticamente no diretório de trabalho atual. Os nomes de arquivo de resposta estão no formato *connection_type*-*sample_type*`.rsp`; por exemplo, `wpm-producer.rsp`. Se necessário, é possível usar o arquivo de resposta gerado para executar novamente o aplicativo de amostra com as mesmas opções, de modo que não tenha que inserir as opções novamente.

Conceitos relacionados

Os aplicativos de amostra:
Os aplicativos de amostra fornecem uma visão geral dos recursos comuns de cada API. É possível usá-los para verificar a sua instalação e o servidor de sistema de mensagens configurado e para ajudar a construir os seus próprios aplicativos.

"Os aplicativos de amostra:" na página 19
Os aplicativos de amostra fornecem uma visão geral dos recursos comuns de cada API. É possível usá-los para verificar a sua instalação e o servidor de sistema de mensagens configurado e para ajudar a construir os seus próprios aplicativos.

Tarefas relacionadas

Construindo os aplicativos de amostra .NET
Quando você constrói um aplicativo .NET de amostra, uma versão executável de sua amostra escolhida é criada.

"Construindo os aplicativos de amostra .NET" na página 21
Quando você constrói um aplicativo .NET de amostra, uma versão executável de sua amostra escolhida é criada.

Construindo os aplicativos de amostra .NET

Quando você constrói um aplicativo .NET de amostra, uma versão executável de sua amostra escolhida é criada.

Antes de começar

Instale o compilador apropriado. Consulte *Instalando o Message Service Client for .NET* na documentação on-line do produto IBM IBM WebSphere MQ . Esta tarefa supõe que você tenha o Visual Studio 2005 instalado e que esteja familiarizado com seu uso.

Procedimento

Para construir um aplicativo de amostra .NET , conclua as etapas a seguir:

1. Clique no arquivo de solução `Samples.sln` fornecido com as amostras do .NET.
2. Clique com o botão direito do mouse na solução `Amostras` na janela Explorador de Solução e selecione **Construir Solução**.

Resultados

Um programa executável é criado na subpasta apropriada da amostra, seja `bin/Debug` ou `bin/Release`, dependendo da configuração que você escolheu. Esse programa possui o mesmo nome que a pasta, com um sufixo CS. Por exemplo, se você estiver construindo a versão C# do aplicativo de amostra do produtor de mensagens, `SampleProducerCS.exe` será criado na pasta `SampleProducer`.

Conceitos relacionados

Os aplicativos de amostra:
Os aplicativos de amostra fornecem uma visão geral dos recursos comuns de cada API. É possível usá-los para verificar a sua instalação e o servidor de sistema de mensagens configurado e para ajudar a construir os seus próprios aplicativos.

"Os aplicativos de amostra:" na página 19

Os aplicativos de amostra fornecem uma visão geral dos recursos comuns de cada API. É possível usá-los para verificar a sua instalação e o servidor de sistema de mensagens configurado e para ajudar a construir os seus próprios aplicativos.

"Construindo seus próprios aplicativos" na página 45
Você constrói seus próprios aplicativos, como você constrói os aplicativos de amostra.

Tarefas relacionadas

Executando os Aplicativos de Amostra
É possível executar os aplicativos de amostra do .NET interativamente no modo simples ou avançado ou não interativamente usando arquivos de resposta gerados automaticamente ou customizados.

"Executando os Aplicativos de Amostra" na página 20
É possível executar os aplicativos de amostra do .NET interativamente no modo simples ou avançado ou não interativamente usando arquivos de resposta gerados automaticamente ou customizados.

Desenvolvendo aplicativos do XMS

Este seçãocapítulo fornece informações que você pode achar úteis ao gravar aplicativos XMS

As informações neste seçãocapítulo aplicam-se a aplicativos .NET

Para obter informações sobre como gravar aplicativos XMS , consulte os tópicos a seguir:

Gravando aplicativos do XMS

Este seçãocapítulo fornece informações para ajudá-lo na gravação de aplicativos XMS

Este seçãocapítulo contém conceitos gerais para gravar aplicativos XMS . Consulte também "Gravando aplicativos XMS .NET" na página 46 para obter informações específicas para criar aplicativos .NET.

Esse seçãocapítulo contém o seguinte tópicosseções:

Referências relacionadas

Interfaces do .NET
Este tópicoseção documenta as interface de classe .NET e suas propriedades e métodos.

O modelo de encadeamento

As regras gerais controlam como um aplicativo multiencadeado pode usar objetos XMS .

- Apenas objetos dos tipos a seguir podem ser usados simultaneamente em encadeamentos diferentes:

  - ConnectionFactory
  - Conexão
  - ConnectionMetaData
  - Destino
- Um objeto de Sessão pode ser usado em apenas um único encadeamento em um determinado momento.

Exceções a essas regras são indicadas por entradas rotuladas "Contexto de encadeamento" nas definições de interface dos métodos em os capítulos de referência da API"Clientes de serviço de mensagens para referência .NET" na página 92.

Condições de erro que podem ser manipuladas no tempo de execução

Os códigos de retorno de chamadas de API são condições de erro que podem ser manipulados no tempo de execução A maneira na qual você lida com esse tipo de erro depende se você está usando a API C ou C++

Como detectar erros no tempo de execução

Se um aplicativo chamar uma função da API C e a chamada falhar, uma resposta com um código de retorno diferente de XMS_OK será retornado com um bloco de erro XMS contendo mais informações sobre o motivo da falha.

A API C++ lança uma exceção quando um método é usado..

Um aplicativo usa um listener de exceção para ser notificado assincronamente de um problema com uma conexão. O listener de exceção é fornecido e inicializado usando a API XMS C ou C++ .

Como manipular erros no tempo de execução

Algumas condições de erro são uma indicação de que algum recurso está indisponível, e a ação que um aplicativo pode executar depende da função XMS que o aplicativo está chamando.. Por exemplo, se uma conexão falhar ao se conectar ao servidor, o aplicativo poderá desejar tentar novamente periodicamente até que uma conexão seja feita Um bloco ou exceção de erro XMS pode não conter informações suficientes para determinar qual ação tomar e, nessas situações, geralmente há um bloco ou exceção de erro vinculado que contém informações de diagnóstico mais específicas.

Na API C, sempre teste uma resposta com um código de retorno diferente de XMS_OK e sempre transmita um bloco de erro na chamada da API. A ação tomada geralmente depende de qual função da API é o aplicativo usando.

Na API C++ , sempre inclua chamadas para métodos em um bloco try e, para capturar todos os tipos de exceção XMS , especifique a classe Exception na construção catch.

O listener de exceção é um caminho de condição de erro assíncrono que pode ser iniciado a qualquer momento.. Quando a função do listener de exceção é iniciada, em seu próprio encadeamento, ela geralmente é uma indicação de uma falha mais grave do que uma condição de erro normal da API do XMS . Qualquer ação apropriada pode ser executada mas deve-se ter cuidado para seguir as regras para o modelo de encadeamento XMS , conforme descrito em "O modelo de encadeamento" na página 22

ConnectionFactories e objetos de Conexão

Um objeto ConnectionFactory fornece um modelo que um aplicativo usa para criar um objeto Connection. O aplicativo usa o objeto Connection para criar um objeto Session.

Para .NET, o aplicativo um XMS usa primeiro um objeto XMSFactoryFactory para obter uma referência a um objeto ConnectionFactory que seja adequado ao tipo de protocolo necessário. Esse objeto ConnectionFactory pode, então, produzir conexões somente para esse tipo de protocolo.

O aplicativo Um XMS pode criar várias conexões e um aplicativo multiencadeado pode usar um único objeto de Conexão simultaneamente em vários encadeamentos Um objeto Connection encapsula uma conexão de comunicações entre um aplicativo e um servidor de sistema de mensagens.

Uma conexão atende a vários propósitos:

- Quando um aplicativo cria uma conexão, o aplicativo pode ser autenticado.
- Um aplicativo pode associar um identificador de cliente exclusivo a uma conexão. O identificador de cliente é usado para suportar assinaturas duráveis no domínio do Publicação/Assinatura . O identificador de cliente pode ser configurado de duas maneiras:

  A maneira preferida de designar um identificador de cliente de conexões é configurar em um objeto ConnectionFactory específico do cliente usando propriedades e designá-lo de forma transparente à conexão criada.

  Uma maneira alternativa de designar um identificador de cliente é usar um valor específico do provedor que é configurado no objeto Connection. Esse valor não substitui o identificador que foi configurado administrativamente. Ele é fornecido para o caso em que não existe nenhum identificador administrativamente especificado. Se um identificador especificado administrativamente existir, uma tentativa de substituí-la com um valor específico do provedor fará com que uma exceção seja lançada. Se um aplicativo configurar explicitamente um identificador, ele deverá fazer isso imediatamente após a criação da conexão e antes de qualquer outra ação na conexão ser tomada; caso contrário, uma exceção será lançada.

O aplicativo Um XMS geralmente cria uma conexão, uma ou mais sessões e vários produtores de mensagens e consumidores de mensagens.

A criação de uma conexão é relativamente cara em termos de recursos do sistema, porque envolve o estabelecimento de uma conexão de comunicação e pode também envolver a autenticação do aplicativo.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Referências relacionadas

IConnectionFactory (para a interface .NET)
Um aplicativo usa um connection factory para criar uma conexão.

Propriedades de ConnectionFactory
Uma visão geral das propriedades do objeto ConnectionFactory , com links para informações de referência mais detalhadas.

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Modo iniciado e interrompido da conexão

Uma conexão pode operar no modo iniciado ou interrompido.

Quando um aplicativo cria uma conexão, a conexão fica no modo interrompido. Quando a conexão está no modo interrompido, o aplicativo pode inicializar as sessões e pode enviar mensagens, mas não pode recebê-las, de forma síncrona ou assíncrona.

Um aplicativo pode iniciar uma conexão chamando o método `Start Connection` .. Quando a conexão está no modo iniciado, o aplicativo pode enviar e receber mensagens. O aplicativo pode, então, parar e reiniciar a conexão chamando os métodos Stop Connection e `Start Connection` .

Conceitos relacionados

Fechamento da conexão
Um aplicativo fecha uma conexão chamando o método Fechar Conexão.

Manipulação de Exceção

Se um aplicativo usar uma conexão apenas para consumir mensagens de forma assíncrona, ele aprenderá sobre um problema com a conexão somente usando um listener de exceções

Conexão com um Barramento de Integração de Serviços do WebSphere
O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

Fechamento da conexão

Um aplicativo fecha uma conexão chamando o método Fechar Conexão.

Quando um aplicativo fecha uma conexão, XMS executa as ações a seguir:

- Ele fecha todas as sessões associadas à conexão e exclui determinados objetos associados a essas sessões. Para obter mais informações sobre quais objetos são excluídos, consulte "Exclusão de objeto" na página 40. Ao mesmo tempo, XMS retrocede quaisquer transações atualmente em andamento dentro das sessões.
- Ele encerra a conexão de comunicações com o servidor de sistema de mensagens.
- Ele libera a memória e outros recursos internos usados pela conexão.

O XMS não reconhece o recebimento de nenhuma mensagem que ele tenha falhado em reconhecer durante uma sessão antes de fechar a conexão. Para obter mais informações sobre como reconhecer o recebimento de mensagens, consulte "Confirmação da mensagem.." na página 28.

Conceitos relacionados

Modo iniciado e interrompido da conexão
Uma conexão pode operar no modo iniciado ou interrompido.

Manipulação de Exceção
Se um aplicativo usar uma conexão apenas para consumir mensagens de forma assíncrona, ele aprenderá sobre um problema com a conexão somente usando um listener de exceções

Conexão com um Barramento de Integração de Serviços do WebSphere
O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

Manipulação de Exceção

Se um aplicativo usar uma conexão apenas para consumir mensagens de forma assíncrona, ele aprenderá sobre um problema com a conexão somente usando um listener de exceções

As exceções XMS.NET são todas derivadas de System.Exception. Para obter mais informações, consulte "Manipulação de erros em .NET" na página 50.

Conceitos relacionados

Modo iniciado e interrompido da conexão
Uma conexão pode operar no modo iniciado ou interrompido.

Fechamento da conexão
Um aplicativo fecha uma conexão chamando o método Fechar Conexão.

Conexão com um Barramento de Integração de Serviços do WebSphere
O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

Conexão com um Barramento de Integração de Serviços do WebSphere

O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

O protocolo HTTP pode ser usado em situações em que uma conexão TCP/IP direta não é possível. Uma situação comum é quando se comunica através de um firewall, como quando duas empresas trocam mensagens. Usar HTTP para se comunicar por meio de um firewall geralmente é referido como *tunelamento HTTP.* O tunelamento HTTP, no entanto, é inerentemente mais lento do que o uso de uma conexão TCP/IP direta, porque os cabeçalhos de HTTP incluem significativamente a quantidade de dados que são transferidos e porque o protocolo HTTP requer mais fluxos de comunicação do que o TCP/IP.

Para criar uma conexão TCP/IP, um aplicativo pode usar um connection factory cuja propriedade XMSC_WPM_TARGET_TRANSPORT_CHAIN esteja configurada como XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC. Esse é o valor padrão da propriedade. Se a conexão for criada com sucesso, a propriedade XMSC_WPM_CONNECTION_PROTOCOL da conexão será configurada como XMSC_WPM_CP_TCP.

Para criar uma conexão que utiliza HTTP, um aplicativo deve usar um connection factory cuja propriedade XMSC_WPM_TARGET_TRANSPORT_CHAIN esteja configurada para o nome de uma cadeia de transporte de entrada, que está configurada para usar um canal de transporte HTTP. Se a conexão for criada com sucesso, a propriedade XMSC_WPM_CONNECTION_PROTOCOL da conexão será configurada como XMSC_WPM_CP_HTTP. Para obter informações sobre como configurar as cadeias de transporte, consulte Cadeias de transporte na duplicação do WebSphere Application Server

Um aplicativo possui uma opção semelhante de protocolos de comunicação ao se conectar a um servidor de auto-inicialização. A propriedade XMSC_WPM_PROVIDER_ENDPOINTS de um connection factory é uma sequência de um ou mais endereços de terminal de servidores de autoinicialização. O componente de cadeia de transporte de auto-inicialização de cada endereço de terminal pode ser XMSC_WPM_BOOTSTRAP_TCP, para uma conexão TCP/IP com um servidor de auto-inicialização ou XMSC_WPM_BOOTSTRAP_HTTP, para uma conexão que usa HTTP.

Conceitos relacionados

Modo iniciado e interrompido da conexão
Uma conexão pode operar no modo iniciado ou interrompido.

Fechamento da conexão
Um aplicativo fecha uma conexão chamando o método Fechar Conexão.

Manipulação de Exceção
Se um aplicativo usar uma conexão apenas para consumir mensagens de forma assíncrona, ele aprenderá sobre um problema com a conexão somente usando um listener de exceções

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Referências relacionadas

IConnectionFactory (para a interface .NET)
Um aplicativo usa um connection factory para criar uma conexão.

Propriedades de ConnectionFactory
Uma visão geral das propriedades do objeto ConnectionFactory , com links para informações de referência mais detalhadas.

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Sessões

Uma sessão é um único contexto encadeado para enviar e receber mensagens.

Um aplicativo pode usar uma sessão para criar mensagens, produtores de mensagens, consumidores de mensagens, navegadores de filas e destinos temporários. Um aplicativo também pode usar uma sessão para executar transações locais.

Um aplicativo pode criar várias sessões, em que cada sessão produz e consome mensagens independentemente das outras sessões. Se dois consumidores de mensagem em sessões separadas (ou

mesmo na mesma sessão) assinam o mesmo tópico, cada um recebe uma cópia de qualquer mensagem publicada nesse tópico.

Diferente de um objeto Connection, um objeto Session não pode ser usado simultaneamente em encadeamentos diferentes. Apenas o método Fechar Sessão de um objeto de Sessão pode ser chamado a partir de um encadeamento diferente daquele que o objeto Session está usando no momento. O método Fechar Sessão termina uma sessão e libera os recursos do sistema alocados para a sessão.

Se um aplicativo precisar processar mensagens simultaneamente em mais de um encadeamento, o aplicativo deverá criar uma sessão em cada encadeamento e, em seguida, usar essa sessão para qualquer operação de envio ou recebimento dentro desse encadeamento.

Sessões transacionadas

Os aplicativos XMS podem executar transações locais. Uma *transação local* é aquela que envolve mudanças apenas para os recursos do gerenciador de filas ou do barramento de integração de serviços ao qual o aplicativo está conectado.

As informações neste tópicoseção serão relevantes somente se um aplicativo se conectar a um gerenciador de fila do IBM WebSphere MQ ou a um barramento de integração de serviços do WebSphere As informações não são relevantes para uma conexão em tempo real com um broker.

Para executar transações locais, um aplicativo deve primeiro criar uma sessão transacionada ao chamar o método Create Session de um objeto Connection, especificando como um parâmetro que a sessão é transacionada. Em seguida, todas as mensagens enviadas e recebidas dentro da sessão são agrupadas em uma sequência de transações. Uma transação terminará quando o aplicativo confirmar ou recuperar as mensagens que ele enviou e recebeu desde o início da transação.

Para confirmar uma transação, um aplicativo chama o método Commit do objeto Session. Quando uma transação for confirmada, todas as mensagens enviadas dentro da transação se tornarão disponíveis para entrega para outros aplicativos e todas as mensagens recebidas dentro da transação serão confirmadas para que o servidor de sistema de mensagens não tente entregá-las ao aplicativo novamente. No domínio ponto-a-ponto , o servidor de mensagens também remove as mensagens recebidas de suas filas.

Para retroceder uma transação, um aplicativo chama o método Rollback do objeto Session. Quando uma transação for retrocedida, todas as mensagens enviadas dentro da transação serão descartadas pelo servidor de sistema de mensagens e todas as mensagens recebidas dentro da transação se tornarão disponíveis para entrega novamente. No domínio ponto-a-ponto , as mensagens que foram recebidas são colocadas novamente em suas filas e se tornam visíveis para outros aplicativos novamente.

Uma nova transação é iniciada automaticamente quando um aplicativo cria uma sessão transacionada ou chama o método Commit ou Rollback. Portanto, uma sessão transacionada sempre possui uma transação ativa.

Quando um aplicativo fechar uma sessão transacionada, um retrocesso implícito ocorrerá. Quando um aplicativo fechar uma conexão, um retrocesso implícito ocorrerá para todas as sessões transacionadas da conexão.

Uma transação é completamente contida dentro de uma sessão transacionada. Uma transação não pode abranger as sessões. Isso significa que não é possível para um aplicativo enviar e receber mensagens em duas ou mais sessões transacionadas e, em seguida, confirmar ou retroceder todas estas ações como uma única transação.

Conceitos relacionados

Confirmação da mensagem..
Cada sessão que não é transacionada tem um modo de reconhecimento que determina como as mensagens recebidas pelo aplicativo são confirmados Três modos de confirmação estão disponíveis, e a escolha do modo de confirmação afeta o design do aplicativo

Entrega de Mensagem Assíncrona
XMS usa um encadeamento para manipular todas as entregas de mensagens assíncronas para uma sessão. Isso significa que apenas uma função de listener de mensagem ou um método `onMessage()` pode ser executado de cada vez.

Entrega de Mensagem Síncrona
As mensagens são entregues de forma síncrona para um aplicativo se o aplicativo usar os métodos Receive de MessageConsumer objetos.

Modo de entrega de mensagens
O XMS suporta dois modos de entrega de mensagens

Confirmação da mensagem..

Cada sessão que não é transacionada tem um modo de reconhecimento que determina como as mensagens recebidas pelo aplicativo são confirmados Três modos de confirmação estão disponíveis, e a escolha do modo de confirmação afeta o design do aplicativo

As informações neste tópicoseção serão relevantes apenas se um aplicativo se conectar a um gerenciador de fila do IBM WebSphere MQ ou a um WebSphere Serviço Integration Bus As informações não são relevantes para uma conexão em tempo real com um broker.

XMS usa o mesmo mecanismo para reconhecer o recebimento de mensagens que o JMS utiliza.

Se uma sessão não for transacionada, a maneira como as mensagens recebidas pelo aplicativo são reconhecidas será determinada pelo modo de confirmação da sessão. Os três modos de reconhecimento são descritos nos seguintes parágrafos:

XMSC_AUTO_ACKNOWLEDGE
A sessão confirma automaticamente cada mensagem recebida pelo aplicativo.

Se as mensagens forem entregues de forma síncrona para o aplicativo, a sessão confirmará o recebimento de uma mensagem toda vez que uma chamada Receive for concluída com sucesso.

Se o aplicativo receber uma mensagem com êxito, mas uma falha impedir a ocorrência de confirmação, a mensagem se tornará disponível para entrega novamente O aplicativo deve, portanto, ser capaz de manipular uma mensagem que seja entregue novamente.

XMSC_DUPS_OK_ACKNOWLEDGE
A sessão confirma as mensagens recebidas pelo aplicativo em momentos que ele seleciona.

O uso desse modo de confirmação reduz a quantidade de trabalho que a sessão deve executar, mas uma falha que impede o reconhecimento da mensagem pode resultar em mais de uma mensagem ficar disponível para entrega novamente. O aplicativo deve, portanto, ser capaz de manipular mensagens que são entregues novamente.

XMSC_CLIENT_ACKNOWLEDGE
O aplicativo confirma as mensagens que ele recebe chamando o método Acknowledge da classe Message.

O aplicativo pode confirmar o recebimento de cada mensagem individualmente ou receber um lote de mensagens e chamar o método Acknowledge apenas para a última mensagem que ele recebe. Quando o método Acknowledge for chamado, todas as mensagens recebidas desde a última vez que o método foi chamado serão confirmadas.

Em conjunto com qualquer um desses modos de confirmação, um aplicativo pode parar e reiniciar a entrega de mensagens em uma sessão chamando o método Recover da classe Session. Mensagens cujo recebimento tenha sido anteriormente não reconhecido são entregues novamente. No entanto, elas não podem ser entregues na mesma sequência em que foram entregues anteriormente. No entretanto, mensagens de prioridade superior podem ter chegado e algumas das mensagens originais podem ter expirado. No domínio ponto a ponto, algumas das mensagens originais podem ter sido consumidas por outro aplicativo.

Um aplicativo pode determinar se uma mensagem está sendo entregue novamente examinando o conteúdo do campo de cabeçalho JMSRedelivered da mensagem. O aplicativo faz isso chamando o método Get JMSRedelivered da classe Message.

Conceitos relacionados
Sessões transacionadas

Os aplicativos XMS podem executar transações locais. Uma *transação local* é aquela que envolve mudanças apenas para os recursos do gerenciador de filas ou do barramento de integração de serviços ao qual o aplicativo está conectado.

Entrega de Mensagem Assíncrona
XMS usa um encadeamento para manipular todas as entregas de mensagens assíncronas para uma sessão. Isso significa que apenas uma função de listener de mensagem ou um método onMessage() pode ser executado de cada vez.

Entrega de Mensagem Síncrona
As mensagens são entregues de forma síncrona para um aplicativo se o aplicativo usar os métodos Receive de MessageConsumer objetos.

Modo de entrega de mensagens
O XMS suporta dois modos de entrega de mensagens

Entrega de Mensagem Assíncrona

XMS usa um encadeamento para manipular todas as entregas de mensagens assíncronas para uma sessão. Isso significa que apenas uma função de listener de mensagem ou um método onMessage() pode ser executado de cada vez.

Se mais de um consumidor de mensagens em uma sessão estiver recebendo mensagens de forma assíncrona, e uma função de listener de mensagem ou um método onMessage() estiver entregando uma mensagem para um consumidor de mensagens, qualquer outro consumidor de mensagens que estiver aguardando a mesma mensagem deverá continuar aguardando. Outras mensagens que estão aguardando para serem entregues para a sessão também devem continuar aguardando.

Se um aplicativo requerer a entrega simultânea de mensagens, crie mais de uma sessão para que XMS use mais de um encadeamento para manipular a entrega de mensagens assíncronas. Dessa maneira, mais de uma função do listener de mensagem ou método onMessage() pode ser executado simultaneamente.

Uma sessão não é feita assíncrona, atribuindo um listener de mensagem a um consumidor. Uma sessão se torna assíncrona somente quando o método Connection.Start é chamado. Todas as chamadas síncronas são permitidas até que o método Connection.Start seja chamado. A entrega de mensagens para os consumidores iniciam quando o Connection.Start é chamado.

Se as chamadas síncronas, como a criação de um consumidor ou um produtor, devem ser feitas em uma sessão assíncrona, o Connection.Stop deve ser chamado. Uma sessão pode ser continuada chamando o método Connection.Start para iniciar a entrega de mensagens. A única exceção a isso é o encadeamento de entrega de mensagens da sessão, que é aquele que entrega mensagens para a função de retorno de chamada. Esse encadeamento pode fazer qualquer chamada na sessão (exceto uma chamada Fechar) na função de retorno de chamada de mensagem.

Nota: No modo Não Gerenciado, a chamada MQDISC em uma função de retorno de chamada não é suportada pelo WMQ.NET Portanto, o aplicativo cliente não pode Criar ou Fechar sessões dentro do retorno de chamada MessageListener no modo de recebimento Assíncrono. Crie e despose a sessão fora do método MessageListener .

Conceitos relacionados

Sessões transacionadas
Os aplicativos XMS podem executar transações locais. Uma *transação local* é aquela que envolve mudanças apenas para os recursos do gerenciador de filas ou do barramento de integração de serviços ao qual o aplicativo está conectado.

Confirmação da mensagem..
Cada sessão que não é transacionada tem um modo de reconhecimento que determina como as mensagens recebidas pelo aplicativo são confirmados Três modos de confirmação estão disponíveis, e a escolha do modo de confirmação afeta o design do aplicativo

Entrega de Mensagem Síncrona
As mensagens são entregues de forma síncrona para um aplicativo se o aplicativo usar os métodos Receive de MessageConsumer objetos.

Modo de entrega de mensagens
O XMS suporta dois modos de entrega de mensagens

Entrega de Mensagem Síncrona

As mensagens são entregues de forma síncrona para um aplicativo se o aplicativo usar os métodos Receive de MessageConsumer objetos.

Usando os métodos Receive, um aplicativo pode esperar um período de tempo especificado para uma mensagem ou pode esperar indefinidamente. Como alternativa, se um aplicativo não desejar aguardar por uma mensagem, ele poderá usar o método Receive with No Wait.

Conceitos relacionados

Sessões transacionadas
Os aplicativos XMS podem executar transações locais. Uma *transação local* é aquela que envolve mudanças apenas para os recursos do gerenciador de filas ou do barramento de integração de serviços ao qual o aplicativo está conectado.

Confirmação da mensagem..
Cada sessão que não é transacionada tem um modo de reconhecimento que determina como as mensagens recebidas pelo aplicativo são confirmados Três modos de confirmação estão disponíveis, e a escolha do modo de confirmação afeta o design do aplicativo

Entrega de Mensagem Assíncrona
XMS usa um encadeamento para manipular todas as entregas de mensagens assíncronas para uma sessão. Isso significa que apenas uma função de listener de mensagem ou um método `onMessage()` pode ser executado de cada vez.

Modo de entrega de mensagens
O XMS suporta dois modos de entrega de mensagens

Modo de entrega de mensagens

O XMS suporta dois modos de entrega de mensagens

- Mensagens *persistentes* são entregues uma vez. Um servidor de sistema de mensagens toma precauções especiais, como a criação de log das mensagens, para assegurar que as mensagens persistentes não sejam perdidas em trânsito, mesmo no caso de uma falha.
- Mensagens *não persistentes* são entregues no máximo uma vez. Mensagens não persistentes são menos confiáveis do que as mensagens persistentes porque elas podem ser perdidas em trânsito no caso de uma falha.

A opção de modo de entrega é uma troca entre confiabilidade e desempenho. Mensagens não persistentes são geralmente transportadas mais rapidamente do que as mensagens persistentes.

Conceitos relacionados

Sessões transacionadas
Os aplicativos XMS podem executar transações locais. Uma *transação local* é aquela que envolve mudanças apenas para os recursos do gerenciador de filas ou do barramento de integração de serviços ao qual o aplicativo está conectado.

Confirmação da mensagem..
Cada sessão que não é transacionada tem um modo de reconhecimento que determina como as mensagens recebidas pelo aplicativo são confirmados Três modos de confirmação estão disponíveis, e a escolha do modo de confirmação afeta o design do aplicativo

Entrega de Mensagem Assíncrona
XMS usa um encadeamento para manipular todas as entregas de mensagens assíncronas para uma sessão. Isso significa que apenas uma função de listener de mensagem ou um método `onMessage()` pode ser executado de cada vez.

Entrega de Mensagem Síncrona
As mensagens são entregues de forma síncrona para um aplicativo se o aplicativo usar os métodos Receive de MessageConsumer objetos.

Destinos

Um aplicativo XMS usa um objeto de Destino para especificar o destino das mensagens que estão sendo enviadas e a origem de mensagens que estão sendo recebidas.

Um aplicativo XMS pode criar um objeto Destination no tempo de execução ou obter um destino predefinido a partir do repositório de objetos administrados.

Como com um ConnectionFactory, a maneira mais flexível para um aplicativo XMS para especificar um destino é defini-la como um objeto administrado. Usando essa abordagem, os aplicativos gravados em C, C++, .NET idiomas e Javapodem compartilhar definições do destino. As propriedades de objetos de Destino administrados podem ser mudadas sem alterar qualquer código.

Para aplicativos .NET , você cria um destino usando o método CreateTopic ou CreateQueue. Esses dois métodos estão disponíveis em ambos os objetos ISession e XMSFactoryFactory na API .NET . Para obter mais informações, consulte o "Destinos em .NET" na página 48 e o "IDestination" na página 111.

Referências relacionadas

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Identificadores de recursos uniformes do tópico

O URI (Identificador Uniforme de Recursos (URI) do tópico especifica o nome do tópico; ele também pode especificar uma ou mais propriedades para ele.

O URI para um tópico inicia com o tópico de sequência: //, seguido pelo nome do tópico e (opcional) uma lista de pares nome-valor que configuram as propriedades do tópico restantes. Um nome de tópico não pode estar vazio.

Aqui está um exemplo em um fragmento de código .NET :

```
topic = session.CreateTopic("topic://Sport/Football/Results?multicast=7");
```

Para obter mais informações sobre as propriedades de um assunto, incluindo o nome e os valores válidos que podem ser usados em um URI, consulte "Propriedades de Destino" na página 192.

Ao especificar um URI de tópico para uso em uma assinatura, os curingas podem ser usados. A sintaxe para esses curingas depende do tipo de conexão e da versão do broker; as opções a seguir estão disponíveis:

- IBM WebSphere MQ V7.0 gerenciador de filas com formato curinga no nível de caractere
- IBM WebSphere MQ V7.0 gerenciador de filas com formato curinga de nível de Tópico
- IBM WebSphere MQ V6.0 gerenciador de filas com o broker V1 (IBM WebSphere MQ V6.0 Publicação/ Assinatura)
- IBM WebSphere MQ V6.0 com ou com conexão em tempo real com o broker V2 (WebSphere Event Broker ou WebSphere Message Broker)
- WebSphere barramento de integração de serviços

IBM WebSphere MQ V7.0 gerenciador de filas com formato curinga no nível de caractere

IBM WebSphere MQ V7.0 gerenciador de filas com o formato curinga de nível de caractere usa os seguintes caracteres curinga:

* para 0 ou mais caracteres
? para 1 caractere

% para um caractere de escape

Tabela 1 na página 32 fornece alguns exemplos de como usar esse esquema curinga.

Tabela 1. URIs de exemplo usando o esquema curinga de nível de caractere para o gerenciador de filas IBM WebSphere MQ V7.0

Identificador Uniforme de Recursos	**Correspondentes**	**Examples**
"topic://Sport *Results"	Todos os tópicos iniciados com "Sport" e terminam em "Results"	"topic://SportsResults" e "topic://Sport/Hockey/National/Div3/Results"
"topic://Sport?Results"	Todos os tópicos iniciando com "Sport" seguido por um caractere único, seguido por "Resultados"	"topic://SportsResults" e "topic://SportXResults"
"topic://Sport/ * ball*/Div? / Results/*/???"	tópicos	"topic://Sport/Football/Div1/Results/2002/Nov" e "topic://Sport/Netball/National/Div3/Results/02/Jan"

IBM WebSphere MQ V7.0 gerenciador de filas com formato curinga de nível de Tópico

IBM WebSphere MQ V7.0 gerenciador de filas com o formato curinga de nível de Tópico usa os seguintes caracteres curinga:

para corresponder a diversos níveis
+ para corresponder a um único nível

Tabela 2 na página 32 fornece alguns exemplos de como usar esse esquema curinga.

Tabela 2. URIs de exemplo usando esquema curinga de nível de tópico para o gerenciador de filas IBM WebSphere MQ V7.0

Identificador Uniforme de Recursos	**Correspondentes**	**Examples**
"topic://Sport/ + / Results"	Todos os Tópicos com um Nome de Nível Hierárquico Único entre Esporte e Resultados	"topic://Sport/Football/Results" e "topic://Sport/Ju-Jitsu/Results"
"topic://Sport/#/Results"	Todos os tópicos iniciando com "Sport/" e terminando em "/Results"	"topic://Sport/Football/Results" e "topic://Sport/Hockey/National/Div3/Results"
" topic://Sport/Football/#"	Todos os tópicos iniciando com "Sport/Football/"	"topic://Sport/Football/Results" e "topic://Sport/Football/TeamNews/Signings/Managerial"

IBM WebSphere MQ V6.0 gerenciador de filas com o broker V1

IBM WebSphere MQ V6.0 gerenciador de filas com o broker V1 usa os seguintes caracteres curinga:

* para 0 ou mais caracteres
? para 1 caractere
% para um caractere de escape

Tabela 1 na página 32 fornece alguns exemplos de como usar esse esquema curinga.

IBM WebSphere MQ V6.0 com ou com conexão em tempo real com um broker V2

IBM WebSphere MQ V6.0 com, ou conexão em tempo real com, um broker V2 usa os seguintes caracteres curinga:

para corresponder a diversos níveis
+ para corresponder a um único nível

Tabela 2 na página 32 fornece alguns exemplos de como usar esse esquema curinga.

WebSphere barramento de integração de serviços

O WebSphere barramento de integração de serviços usa os seguintes caracteres curinga:

* para corresponder a qualquer caractere em um nível na hierarquia
// para corresponder a 0 ou mais níveis
//. para combinar com 0 ou mais níveis (no final de uma expressão Tópico)

Tabela 3 na página 33 fornece alguns exemplos de como usar esse esquema curinga.

Tabela 3. URIs de exemplo usando esquema curinga para o WebSphere barramento de integração de serviços

Identificador Uniforme de Recursos	**Correspondentes**	**Examples**
"topic://Sport/ * ball/ Results"	Todos os tópicos com um nome de nível hierárquico único terminando em "ball" entre Esporte e Resultados	"topic://Sport/Football/Results" e "topic:// Sport/Netball/Results"
"topic://Sport// Results"	Todos os tópicos iniciando com "Sport/" e terminando em "/Results"	"topic://Sport/Football/Results" e "topic:// Sport/Hockey/National/Div3/Results"
"topic://Sport/ Football//."	Todos os tópicos iniciando com "Sport/ Football/"	"topic://Sport/Football/Results" e "topic://Sport/Football/TeamNews/ Signings/Managerial"
"topic://Sport/ * ball// Results//."	tópicos	"topic://Sport/Football/Results" e "topic://Sport/Netball/National/Div3/ Results/2002/November"

Conceitos relacionados

Identificadores uniformes de recursos da fila
O URI para uma fila especifica o nome da fila; ele também pode especificar uma ou mais propriedades da fila.

Destinos Temporários
Os aplicativos XMS podem criar e usar destinos temporários.

Curinga de destino
O XMS fornece suporte para curingas de destino, assegurando que os curingas possam ser transmitidos para o local no qual eles são necessários para correspondência Há um esquema curinga diferente para cada tipo de servidor com o qual XMS pode trabalhar.

Referências relacionadas

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Identificadores uniformes de recursos da fila

O URI para uma fila especifica o nome da fila; ele também pode especificar uma ou mais propriedades da fila.

O URI para uma fila começa com a fila de sequência: //, seguido pelo nome da fila; ele também pode incluir uma lista de pares nome-valor que configuram as propriedades da fila restantes..

Para as filas IBM WebSphere MQ (mas não para filas do provedor de mensagens padrão WebSphere Servidor de Aplicação), o gerenciador de filas no qual a fila reside pode ser especificado antes da fila, com um / separando o nome do gerenciador de filas do nome da fila.

Se um gerenciador de filas for especificado, ele deverá ser aquele para o qual o XMS está diretamente conectado para a conexão usando essa fila ou deve ser acessível a partir desta fila. Os gerenciadores de filas remotas são suportados apenas para recuperar mensagens das filas, não para colocar mensagens nas filas. Para obter detalhes completos, consulte a documentação do gerenciador de filas IBM WebSphere MQ .

Se nenhum gerenciador de filas for especificado, então, o extra / separador será opcional e sua presença ou ausência não fará diferença para a definição da fila.

As seguintes definições de fila são todas equivalentes para uma fila IBM WebSphere MQ chamada QB em um gerenciador de filas chamado QM_A, para o qual XMS está diretamente conectado:

```
queue://QB
queue:///QB
queue://QM_A/QB
```

Conceitos relacionados

Identificadores de recursos uniformes do tópico
O URI (Identificador Uniforme de Recursos (URI) do tópico especifica o nome do tópico; ele também pode especificar uma ou mais propriedades para ele.

Destinos Temporários
Os aplicativos XMS podem criar e usar destinos temporários.

Curinga de destino
O XMS fornece suporte para curingas de destino, assegurando que os curingas possam ser transmitidos para o local no qual eles são necessários para correspondência Há um esquema curinga diferente para cada tipo de servidor com o qual XMS pode trabalhar.

Referências relacionadas

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Destinos Temporários

Os aplicativos XMS podem criar e usar destinos temporários.

Um aplicativo geralmente usa um destino temporário para receber respostas para solicitar mensagens. Para especificar o destino no qual uma resposta a uma mensagem de solicitação deve ser enviada, um aplicativo chama o método Set JMSReplyTo do objeto de mensagem que representa a mensagem de solicitação. O destino especificado na chamada pode ser um destino temporário.

Embora uma sessão seja usada para criar um destino temporário, o escopo de um destino temporário é, na verdade, a conexão que foi usada para criar a sessão. Qualquer uma das sessões da conexão pode criar produtores de mensagens e consumidores de mensagens para o destino temporário. O destino temporário permanece até que seja explicitamente excluído ou que a conexão termine, o que ocorrer primeiro.

Quando um aplicativo cria uma fila temporária, uma fila é criada no servidor de sistema de mensagens para o qual o aplicativo está conectado. Se o aplicativo estiver conectado a um gerenciador de filas, uma fila dinâmica será criada a partir da fila modelo cujo nome é especificado pela propriedade XMSC_WMQ_TEMPORARY_MODEL e o prefixo usado para formar o nome da fila dinâmica será especificado pela propriedade XMSC_WMQ_TEMP_Q_PREFIX . Se o aplicativo estiver conectado a um barramento de integração de serviços, uma fila temporária será criada no barramento e o prefixo usado para formar o nome da fila temporária será especificado pela propriedade XMSC_WPM_TEMP_Q_PREFIX .

Quando um aplicativo que está conectado a um barramento de integração de serviços cria um tópico temporário, o prefixo usado para formar o nome do tópico temporário é especificado pela propriedade XMSC_WPM_TEMP_TOPIC_PREFIX .

Conceitos relacionados

Identificadores de recursos uniformes do tópico
O URI (Identificador Uniforme de Recursos (URI) do tópico especifica o nome do tópico; ele também pode especificar uma ou mais propriedades para ele.

Identificadores uniformes de recursos da fila
O URI para uma fila especifica o nome da fila; ele também pode especificar uma ou mais propriedades da fila.

Curinga de destino
O XMS fornece suporte para curingas de destino, assegurando que os curingas possam ser transmitidos para o local no qual eles são necessários para correspondência Há um esquema curinga diferente para cada tipo de servidor com o qual XMS pode trabalhar.

Referências relacionadas

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Curinga de destino

O XMS fornece suporte para curingas de destino, assegurando que os curingas possam ser transmitidos para o local no qual eles são necessários para correspondência Há um esquema curinga diferente para cada tipo de servidor com o qual XMS pode trabalhar.

Os esquemas são:

Tipo de conexão	Esquema curinga	Descrição
Gerenciador de filas IBM WebSphere MQ	* ? %	0 ou mais caracteres 1 caractere Caractere de escape
Conexão de tempo real com um broker	# +	Corresponder vários níveis Corresponder um único nível
WebSphere Serviço Integration Bus	* // //.	Corresponder quaisquer caracteres em um nível na hierarquia Corresponder 0 ou mais níveis Corresponder 0 ou mais níveis (no final de uma expressão de Tópico)

Consulte também Nomes de tópicos e uso de caracteres curingas em expressões de tópico na documentação do WebSphere Application Server .

Conceitos relacionados

Identificadores de recursos uniformes do tópico
O URI (Identificador Uniforme de Recursos (URI) do tópico especifica o nome do tópico; ele também pode especificar uma ou mais propriedades para ele.

Identificadores uniformes de recursos da fila
O URI para uma fila especifica o nome da fila; ele também pode especificar uma ou mais propriedades da fila.

Destinos Temporários
Os aplicativos XMS podem criar e usar destinos temporários.

Referências relacionadas

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Produtores de mensagens

Em XMS, um produtor de mensagens pode ser criado com um destino válido ou sem destino associado. Ao criar um produtor de mensagem com um destino nulo, um destino válido precisa ser especificado ao enviar uma mensagem.

Produtores de mensagens sem destino associado

Em XMS .NET, um produtor de mensagem pode ser criado com um destino nulo.

Para criar um produtor de mensagem sem destino associado ao usar a API .NET, NULL deve ser transmitido como um parâmetro no método `CreateProducer()` do objeto ISession (por exemplo, `session.CreateProducer(null)`). No entanto, um destino válido deve ser especificado quando a mensagem for enviada

Produtores de mensagens com destino associado

Nesse cenário, o produtor da mensagem é criado usando um destino válido. Durante a operação de envio, o destino não precisa ser especificado.

Consumidores de mensagens

Os consumidores de mensagens podem ser classificados como assinantes duráveis e não duráveis e consumidores de mensagens síncrona e assíncrona.

Assinantes duráveis

Um assinante durável é um consumidor de mensagens que recebe todas as mensagens publicadas em um tópico, incluindo mensagens publicadas enquanto o assinante está inativo.

As informações neste tópicoseção serão relevantes somente se um aplicativo se conectar a um gerenciador de fila do IBM WebSphere MQ ou a um barramento de integração de serviços doWebSphere As informações não são relevantes para uma conexão em tempo real com um broker.

Para criar um assinante durável para um tópico, um aplicativo chama o método Criar Assinante Durável de um objeto de Sessão, especificando como parâmetros um nome que identifica a assinatura durável e um objeto de Destino que representa o tópico. O aplicativo pode criar um assinante durável com ou sem um seletor de mensagem, e pode especificar se o assinante durável deve receber mensagens publicadas por sua própria conexão.

A sessão usada para criar um assinante durável deve ter um identificador de cliente associado. O identificador de cliente é o mesmo que aquele associado à conexão que é usada para criar a sessão; ela é especificada conforme descrito em “ConnectionFactories e objetos de Conexão” na página 23.

O nome que identifica a assinatura durável deve ser exclusivo dentro do identificador de cliente e, portanto, o identificador de cliente faz parte do identificador completo e exclusivo da assinatura durável. O servidor de sistema de mensagens mantém um registro da assinatura durável e assegura que todas as mensagens publicadas no tópico sejam retidas até que sejam reconhecidas pelo assinante durável ou que elas expirem.

O servidor de sistema de mensagens continua a manter o registro da assinatura durável mesmo depois que o assinante durável for fechado. Para reutilizar uma assinatura durável que foi criada anteriormente, um aplicativo deve criar um assinante durável especificando o mesmo nome de assinatura e usando uma sessão com o mesmo identificador de cliente, como aqueles associados à assinatura durável. Apenas uma sessão de cada vez pode ter um assinante permanente para uma assinatura durável específica.

O escopo de uma assinatura durável é o servidor de sistema de mensagens que está mantendo um registro da assinatura. Se dois aplicativos conectados a diferentes servidores de sistema de mensagens criarem um assinante durável usando o mesmo nome de assinatura e o identificador de cliente, duas assinaturas duráveis completamente independentes serão criadas.

Para excluir uma assinatura durável, um aplicativo chama o método Unsubscribe de um objeto Session, especificando como um parâmetro o nome que identifica a assinatura durável. O identificador de cliente associado à sessão deve ser o mesmo que aquele associado à assinatura durável. O servidor de mensagens exclui o registro da assinatura durável que ele está mantendo e não envia mais mensagens para o assinante durável.

Para alterar uma assinatura existente, um aplicativo pode criar um assinante durável usando o mesmo nome de assinatura e identificador de cliente, mas especificando um tópico diferente ou seletor de mensagem (ou ambos). Alterar uma assinatura durável é equivalente a excluir a assinatura e criar uma nova.

Para um aplicativo que se conecta ao gerenciador de filas do IBM WebSphere MQ V7.0 , o XMS gerencia as filas de assinantes. Portanto, o aplicativo não é necessário para especificar uma fila de assinantes. O XMS ignorará a fila de assinantes, se especificada.

No entanto, para um aplicativo que se conecta ao gerenciador de filas do IBM WebSphere MQ V6.0 , cada assinante durável deve ter uma fila de assinantes designada Para especificar o nome da fila de assinantes para um tópico, configure a propriedade XMSC_WMQ_DUR_SUBQ do objeto de Destino que representa o tópico. A fila de assinantes padrão é SYSTEM.JMS.D.SUBSCRIBER.QUEUE.

Os assinantes duráveis que se conectar aos gerenciadores de filas do IBM WebSphere MQ V6.0 podem compartilhar uma única fila de assinantes ou cada assinante durável pode recuperar suas mensagens a partir de sua própria fila de assinantes exclusivos. Para uma discussão sobre qual abordagem adotar para seu aplicativo, consulte *IBM WebSphere MQ Usando Java*.

Observe que não é possível alterar a fila de assinantes para uma assinatura durável. A única maneira de alterar a fila de assinantes é excluir a assinatura e criar uma nova.

Para um aplicativo que se conecta a um barramento de integração de serviços, cada assinante durável deve ter um home de assinatura durável designado. Para especificar o início da assinatura durável para todos os assinantes permanentes que usam a mesma conexão, configure a propriedade XMSC_WPM_DUR_SUB_HOME do objeto ConnectionFactory usado para criar a conexão. Para especificar o início da assinatura durável para um tópico individual, configure a propriedade XMSC_WPM_DUR_SUB_HOME do objeto Destination que representa o tópico. Um lar de assinaturas duráveis deve ser especificado para uma conexão antes que um aplicativo possa criar um assinante durável que use a conexão. Qualquer valor especificado para um destino substitui o valor especificado para a conexão.

Assinantes Não Duráveis

Um assinante não durável é um consumidor de mensagens que recebe apenas mensagens que são publicadas enquanto o assinante está ativo. As mensagens entregues enquanto o assinante está inativo são perdidas.

As informações neste tópicoseção são relevantes apenas quando você estiver usando o gerenciador de filas do Publicação/Assinatura messaging over IBM WebSphere MQ V6.0 .

Se os objetos de consumidor não forem excluídos antes ou durante o fechamento da conexão, as mensagens poderão ser deixadas nas filas do broker para assinantes que não estiverem mais ativos

Nessa situação, as mensagens podem ser limpas nessas filas usando o utilitário de Limpeza fornecido com o IBM WebSphere MQ Classes for JMS. Os detalhes de como usar esse utilitário são fornecidos em *IBM WebSphere MQ Usando Java.* Também pode ser necessário aumentar a profundidade da fila de assinantes se houver grandes números de mensagens deixadas nessa fila.

Consumidores de mensagens síncrona

O consumidor de mensagens síncronas recebe as mensagens de uma fila de forma síncrona

Um consumidor de mensagens síncrona recebe uma mensagem de cada vez. Quando o método `Receive(wait interval)` é usado; a chamada aguarda apenas um período de tempo especificado em milissegundos para uma mensagem ou até o consumidor de mensagens ser fechado.

Se o método ReceiveNoWait () for usado, o consumidor de mensagens síncrona receberá mensagens sem nenhum atraso; se a próxima mensagem estiver disponível, ela será recebida imediatamente, caso contrário, um ponteiro para um objeto de Mensagem nulo será retornado.

Consumidores de mensagens assíncronas

O consumidor de mensagens assíncronas recebe uma mensagem de uma fila de forma assíncrona O listener de mensagem registrado pelo aplicativo é chamado sempre que uma nova mensagem está disponível na fila.

Mensagens Suspeitas

Uma mensagem suspeita é aquela que não pode ser processada por um aplicativo MDB de recebimento. Se uma mensagem suspeita for encontrada, o objeto XMS MessageConsumer poderá reenfileirá-lo de acordo com duas propriedades de fila, BOQNAME e BOTHRESH

Em algumas circunstâncias, uma mensagem entregue para um MDB pode ser recuperada em uma fila do IBM WebSphere MQ . Isso pode acontecer, por exemplo, quando uma mensagem é entregue dentro de uma unidade de trabalho que é, subsequentemente, retrocedida. Uma mensagem que é retrocedida é geralmente entregue novamente, mas uma mensagem mal formatada pode repetidamente fazer com que um MDB falhe e, portanto, não pode ser entregue. Essa mensagem é chamada de uma mensagem suspeita. É possível configurar IBM WebSphere MQ para que a mensagem suspeita seja automaticamente transferida para outra fila para investigação adicional ou seja descartada. Para obter informações sobre como configurar o IBM WebSphere MQ dessa maneira, consulte Manipulando mensagens suspeitas no ASF

Às vezes, uma mensagem mal formatada incorretamente chega em uma fila. Nesse contexto, mal formatada significa que o aplicativo de recebimento não pode processar a mensagem corretamente. Essa mensagem pode fazer com que o aplicativo de recebimento falhe e restaure essa mensagem mal formatada. A mensagem pode então ser entregue repetidamente à fila de entrada e recuperada repetidamente pelo aplicativo. Essas mensagens são conhecidas como mensagens suspeitas. O objeto XMS MessageConsumer detecta mensagens suspeitas e roteia-as para um destino alternativo.

O gerenciador de filas IBM WebSphere MQ mantém um registro do número de vezes que cada mensagem foi restaurada. Quando esse número atinge um valor limite configurável, o consumidor de mensagem recoloca a mensagem em uma fila de restauração denominada. Se esse novo enfileiramento falhar por qualquer razão, a mensagem será removida da fila de entrada e um enfileirada novamente na fila de mensagens não entregues ou descartada.

Os objetos de ConnectionConsumer XMS manipulam mensagens suspeitas da mesma maneira e usando as mesmas propriedades de fila. Se diversos consumidores de conexão estiverem monitorando a mesma

fila, é possível que a mensagem suspeita possa ser entregue a um aplicativo mais vezes do que o valor limite antes que o novo enfileiramento ocorra. Este comportamento ocorre devido à maneira como consumidores de conexões individuais monitoram filas e enfileiram mensagens suspeitas novamente.

O valor do limite e o nome da fila de backup são atributos de uma fila do IBM WebSphere MQ. Os nomes dos atributos são BackoutThreshold e BackoutRequeueQName. A fila à qual eles se aplicam é a seguinte:

- Para o sistema de mensagens ponto a ponto, é a fila local subjacente. Isso é importante quando os consumidores de mensagens e os consumidores de conexão usam aliases de filas.
- Para o sistema de mensagens de publicação/assinatura no modo normal do provedor do sistemas de mensagens do IBM WebSphere MQ, ela é a fila modelo por meio da qual a fila gerenciada do tópico é criada.
- Para o sistema de mensagens de publicar/assinar no modo de migração do provedor do sistema de mensagens do IBM WebSphere MQ, é a fila CCSUB definida no objeto TopicConnectionFactory ou a fila CCDSUB definida no objeto Topic.

Para configurar os atributos BackoutThreshold e BackoutRequeueQName, emita o comando MQSC a seguir:

```
ALTER QLOCAL(your.queue.name) BOTHRESH(threshold value)
BOQNAME(your.backout.queue.name)
```

Para o sistema de mensagens de publicação / assinatura, se o seu sistema criar uma fila dinâmica para cada assinatura, esses valores de atributos serão obtidos das classes IBM WebSphere MQ para a fila modelo JMS, SYSTEM.JMS.MODEL.QUEUE. Para alterar essas configurações, use:

```
ALTER QMODEL(SYSTEM.JMS.MODEL.QUEUE) BOTHRESH(threshold value)
BOQNAME(your.backout.queue.name)
```

Se o valor do limite de restauração for zero, a manipulação de mensagens suspeitas será desativada e as mensagens suspeitas permanecerão na fila de entrada. Caso contrário, quando a contagem de restaurações atingir o valor do limite, a mensagem será enviada para a fila de restauração denominada. Se a contagem de restaurações atingir o valor do limite, mas a mensagem não puder ir para a fila de restauração, a mensagem será enviada para a fila de mensagens não entregues ou será descartada. Essa situação ocorre se a fila de restauração não estiver definida ou se o objeto MessageConsumer não puder enviar a mensagem para a fila de restauração.

Manipulando mensagens suspeitas no ASF
Ao usar o Application Server Facilities (ASF), o ConnectionConsumer, em vez de o MessageConsumer, processa mensagens suspeitas. O ConnectionConsumer recoloca mensagens na fila de acordo com as propriedades BackoutThreshold e BackoutRequeueQName da fila.

Quando um aplicativo usa ConnectionConsumers, as circunstâncias nas quais uma mensagem é restaurada dependem da sessão que o servidor de aplicativos fornece:

- Quando a sessão é não transacionada, com AUTO_ACKNOWLEDGE ou DUPS_OK_ACKNOWLEDGE, uma mensagem é restaurada somente após um erro do sistema ou se o aplicativo for finalizado inesperadamente.
- Quando a sessão não é transacionada com CLIENT_RECONHEÇO, mensagens não reconhecidas podem ser restauradas pelo servidor de aplicativos que chama `Session.recover()`.

  Geralmente, a implementação do cliente de MessageListener ou as chamadas do servidor de aplicativos `Message.acknowledge()` `Message.acknowledge()` reconhece todas as mensagens entregues na sessão até agora.
- Quando a sessão é transacionada, mensagens não reconhecidas podem ser restauradas pelo servidor de aplicativos chamando `Session.rollback()`.

Navegadores de fila

Um aplicativo usa um navegador de filas para pesquisar mensagens em uma fila sem removê-las.

Para criar um navegador de filas, um aplicativo chama o método Criar Navegador de Filas de um objeto de ISession, especificando como um parâmetro um objeto de Destino que identifica a fila a ser procurada. O aplicativo pode criar um navegador de filas com ou sem um seletor de mensagem.

Depois de criar um navegador de filas, o aplicativo pode chamar o método GetEnumerator do objeto IQueueBrowser para obter uma lista de mensagens na fila. O método retorna um enumerador que encapsula uma lista de objetos de Mensagem. A ordem dos objetos de Mensagem na lista é a mesma que a ordem em que as mensagens seriam recuperadas da fila. O aplicativo pode então usar o enumerador para pesquisar cada mensagem por vez.

O enumerador é atualizado dinamicamente conforme as mensagens são colocadas na fila e removidas da fila. Cada vez que o aplicativo chama IEnumerator.MoveNext () para navegar na próxima mensagem na fila, a mensagem reflete os conteúdos atuais da fila.

Um aplicativo pode chamar o método GetEnumerator mais de uma vez para um navegador de filas fornecido. Cada chamada retorna um novo enumerador. O aplicativo pode, portanto, usar mais de um enumerador para procurar as mensagens em uma fila e manter várias posições dentro da fila.

Um aplicativo pode usar um navegador de filas para procurar por uma mensagem adequada para remover de uma fila e, em seguida, usar um consumidor de mensagens com um seletor de mensagem para remover a mensagem. O seletor de mensagem pode selecionar a mensagem de acordo com o valor do campo de cabeçalho JMSMessageID. Para obter informações sobre esse e outros campos de cabeçalho da mensagem JMS, consulte “Campos de cabeçalho na mensagem um XMS” na página 73.

Solicitantes

Um aplicativo usa um solicitante para enviar uma mensagem de solicitação e, em seguida, esperar e receber a resposta.

Muitos aplicativos do sistema de mensagens são baseados em algoritmos que enviam uma mensagem de solicitação e, em seguida, aguardam uma resposta. XMS fornece uma classe chamada Requestor para ajudar com o desenvolvimento desse estilo de aplicativo.

Para criar um solicitante, um aplicativo chama o construtor Create Requestor da classe Requestor, especificando como parâmetros um objeto Session e um objeto de Destino que identifica onde as mensagens de solicitação devem ser enviadas. A sessão não deve ser transacionada e nem ter um modo de confirmação de XMSC_CLIENT_RECONHEÇO O construtor cria automaticamente uma fila ou um tópico provisório para os quais as mensagens de resposta devem ser enviadas.

Depois de criar um solicitante, o aplicativo pode chamar o método Request do objeto Solicitante para enviar uma mensagem de solicitação e, em seguida, aguardar e receber uma resposta do aplicativo que recebe a mensagem de solicitação. A chamada aguarda até que a resposta seja recebida ou até que a sessão termine, o que ocorrer primeiro. Somente uma resposta é requerida pelo solicitante para cada mensagem de solicitação.

Quando o aplicativo fecha o solicitante, a fila temporária ou o tópico é excluído. A sessão associada, no entanto, não fecha.

Exclusão de objeto

Quando um aplicativo exclui um objeto XMS que ele criou, XMS libera os recursos internos que foram alocados para o objeto.

Quando um aplicativo cria o objeto um XMS , o XMS aloca memória e outros recursos internos para o objeto XMS retém esses recursos internos até que o aplicativo exclua explicitamente o objeto chamando o método close ou delete do objeto, no ponto em que o XMS libera os recursos internos. Se um aplicativo tentar excluir um objeto que já está excluído, a chamada será ignorada.

Quando um aplicativo exclui um objeto Connection ou Session, o XMS exclui determinados objetos associados automaticamente e libera seus recursos internos. Esses são objetos que foram criados pelo objeto Connection ou Session e não têm função independente do objeto. Esses objetos são mostrados em Tabela 4 na página 41.

Nota: Se um aplicativo fechar uma conexão com sessões dependentes, todos os objetos dependentes dessas sessões também serão excluídos. Apenas um objeto Connection ou Session pode ter objetos dependentes.

Tabela 4. Objetos que são excluídos automaticamente

Objeto excluído	Método	Objetos Dependentes que São Excluídos Automaticamente
Conexão	Encerrar Conexão	Objetos ConnectionMetaData e Session
Session	Fechar Sessão	Objetos MessageConsumer, MessageProducer, QueueBrowser e Requestor

Transações XA IBM WebSphere MQ Gerenciadas por meio de XMS

Transações XA IBM WebSphere MQ gerenciadas podem ser usadas por meio de XMS.

Para usar as transações XA por meio do XMS, uma sessão transacionada deve ser criada Quando a transação XA está em uso, o controle de transação é por meio de transações globais do Distributed Transaction Coordinator (DTC) e não é por meio de sessões do XMS . Ao usar transações XA, `Session.commit` ou `Session.rollback` não pode ser emitido na sessão XMS .. Em vez disso, use os métodos DTC `Transscope.Commit` ou `Transscope.Rollback` para confirmar ou retroceder as transações Se uma sessão for usada para transação XA, o produtor ou o consumidor que são criados usando a sessão deve ser uma parte da transação XA. Eles não podem ser usados para nenhuma operação fora do escopo de transação XA. Eles não podem ser usados para operações como `Producer.send` ou `Consumer.receive` fora da transação XA.

Um objeto de exceção `IllegalStateException` será lançado se

- A sessão transacionada por XA é usada para `Session.commit` ou `Session.rollback`.
- Os objetos do produtor ou do consumidor que são usados uma vez na sessão transacionada XA são usados fora do escopo de transação XA.

As transações XA não são suportadas em consumidores assíncronos.

Nota:

1. Um fechamento pode ser emitido no objeto `Producer`, `Consumer`, `Session`, ou `Connection` antes da confirmação da transação XA Em quais casos, as mensagens na transação são retrocedidas. Da mesma forma, se a conexão for interrompida antes da confirmação da transação XA, todas as mensagens na transação serão retrocedidas. Para um objeto `Producer` , um retrocesso significa que as mensagens não são colocadas na fila Para um objeto `Consumer` , um retrocesso significa que as mensagens são deixadas na fila
2. Se um objeto `Producer` colocar uma mensagem com `TimeToLive` no `TransactionScope` e `commit` for emitido após o tempo decorrido, a mensagem poderá expirar antes que o `commit` seja emitido. Nesse caso, a mensagem não é disponibilizada para os objetos `Consumer`
3. Objetos `Session` não são suportados em encadeamentos. O uso de transações com objetos `Session` que são compartilhados entre encadeamentos não é suportado.

Tipos primitivos XMS

O XMS fornece equivalentes dos oito tipos primitivos Java (byte, short, int, long, float, double, char e boolean). Isso permite a troca de mensagens entre XMS e JMS sem que os dados sejam perdidos ou corrompidos.

Tabela 5 na página 42 lista o tipo de dados equivalentes Java , o tamanho e o valor mínimo e máximo de cada tipo primitivo XMS .

Tabela 5. XMS tipos de dados e seus equivalentes Java				
Tipo de Dados XMS	**Tipo de DadosJava Compatível**	**Tamanho**	**Valor mínimo**	**Valor máximo**
System.Boolean	booleano	32 bits	false	true
System.SBYTE	byte	8 bits	-2^7 (-128)	2^7-1 (127)
System.BYTE	byte	8 bits	-2^7 (-128)	2^7-1 (127)
System.CHAR	byte	8 bits	-2^7 (-128)	2^7-1 (127)
System.Int16	short	16 bits	-2^{15} (-32768)	2^{15}-1 (32767)
System.Int32	int	32 bits	-2^{31} (-2147483648)	2^{31}-1 (2147483647)
System.Int64	grande	64 bits	-2^{63} (-9223372036854775808)	2^{63}-1 (9223372036854775807)
System.Single	float	32 bits	-3.402823E-38 (para precisão de 7 dígitos)	3.402823E + 38 (para precisão de 7 dígitos)
System.Double	duplo	64 bits	-1.79769313486231E-308 (para 15-dígitos de precisão)	1.79769313486231E + 308 (para precisão de 15 dígitos)

Conceitos relacionados

Atributos e Propriedades de Objetos
Um objeto XMS pode ter atributos e propriedades, que são características do objeto, que são implementados de diferentes maneiras

Conversão implícita de um valor de propriedade de um tipo de dados para outro
Quando um aplicativo obtém o valor de uma propriedade, o valor pode ser convertido por XMS em outro tipo de dados. Muitas regras controlam quais conversões são suportadas e como o XMS executa as conversões.

Referências relacionadas

Tipos de Dados para Elementos de Dados do Aplicativo
Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Conversão implícita de um valor de propriedade de um tipo de dados para outro

Quando um aplicativo obtém o valor de uma propriedade, o valor pode ser convertido por XMS em outro tipo de dados. Muitas regras controlam quais conversões são suportadas e como o XMS executa as conversões.

Uma propriedade de um objeto tem um nome e um valor; o valor tem um tipo de dado associado, em que o valor de uma propriedade também é referido como o tipo de propriedade ..

Um aplicativo usa os métodos da classe PropertyContext para obter e configurar as propriedades de objetos. Para obter o valor de uma propriedade, um aplicativo chama o método que é apropriado para o tipo de propriedade. Por exemplo, para obter o valor de uma propriedade de número inteiro, um aplicativo geralmente chama o método GetIntProperty.

No entanto, quando um aplicativo obtém o valor de uma propriedade, o valor pode ser convertido por XMS em outro tipo de dados. Por exemplo, para obter o valor de uma propriedade de número inteiro, um aplicativo pode chamar o método GetStringProperty, que retorna o valor da propriedade como uma sequência. As conversões suportadas pelo XMS são mostradas em Tabela 6 na página 43.

Tabela 6. Conversões Suportadas de um Tipo de Propriedade para Outros Tipos de Dados	
Tipo de propriedade	**Tipos de dados de destino suportados**
System.String	System.Boolean, System.Double, System.Float, System.Int32, System.Int64, System.SByte, System.Int16
System.Boolean	System.String, System.SByte, System.Int32, System.Int64, System.Int16
System.Char	System.String
System.Double	System.String
System.Float	System.String, System.Double
System.Int32	System.String, System.Int64
System.Int64	System.String
System.SByte	System.String, System.Int32, System.Int64, System.Int16
Matriz System.SByte	System.String
System.Int16	System.String, System.Int32, System.Int64

As regras gerais a seguir controlam as conversões suportadas:

- Os valores de propriedade numérica podem ser convertidos de um tipo de dados para outro, desde que nenhum dado seja perdido durante a conversão. Por exemplo, o valor de uma propriedade com o tipo de dados System.Int32 pode ser convertido em um valor com o tipo de dados System.Int64, mas não pode ser convertido em um valor com tipo de dados System.Int16.
- Um valor de propriedade de qualquer tipo de dados pode ser convertido em uma sequência.
- Um valor de propriedade de sequência pode ser convertido para qualquer outro tipo de dados, desde que a sequência seja formatada corretamente para a conversão. Se um aplicativo tentar converter um valor de propriedade de sequência que não está formatado corretamente, XMS poderá retornar erros.
- Se um aplicativo tentar uma conversão não suportada, o XMS poderá retornar um erro.

As regras a seguir se aplicam quando um valor de propriedade é convertido de um tipo de dados para outro:

- Ao converter um valor de propriedade booleana para uma sequência, o valor true é convertido para a sequência "true"e o valor false é convertido para a sequência "false".
- Ao converter um valor de propriedade booleana para um tipo de dados numérico, incluindo System.SByte, o valor true será convertido para 1 e o valor false será convertido para 0.
- Ao converter um valor de propriedade de sequência para um valor booleano, a sequência "true" (não faz distinção entre maiúsculas e minúsculas) ou "1" é convertida para true e a sequência "false" (sem distinção entre maiúsculas e minúsculas) ou "0" é convertida para false. Todas as outras sequências não podem ser convertidas.
- Ao converter um valor de propriedade de sequência para um valor com tipo de dados System.Int32, System.Int64, System.SByte ou System.Int16, a sequência deve ter o formato a seguir:

  [*espaços em branco*][*sinal*] *dígitos*

  Os componentes de sequência são definidos da seguinte forma:

  espaços em branco
  Caracteres em branco à esquerda opcionais.

  sinal
  Um caractere de sinal de mais (+) ou de sinal de menos (-) opcional.

  dígitos
  Uma sequência contígua de caracteres de dígito (0-9). Pelo menos um caractere de dígito deve estar presente.

Após a sequência de caracteres de dígito, a sequência pode conter outros caracteres que não são caracteres de dígito, mas a conversão para assim que o primeiro desses caracteres for atingido. A sequência é assumida para representar um número inteiro decimal.

XMS pode retornar um erro se a sequência não estiver formatada corretamente.

- Ao converter um valor de propriedade de sequência para um valor com tipo de dados System.Double ou System.Float, a sequência deve ter o formato a seguir:

  [*blanks*] [*sign*] [*digits*] [*point*[*d_digits*]] [*e_char*[*e_sign*]*e_digits*]

  Os componentes de sequência são definidos da seguinte forma:

  espaços em branco
  (Opcional) Leitura de caracteres em branco.

  sinal
  (Opcional) Sinal de mais (+) ou sinal de menos (-).

  dígitos
  Uma sequência contígua de caracteres de dígito (0-9). Pelo menos um caractere de dígito deve estar presente em *digits* ou *d_digits*.

  ponto
  (Opcional) Separador decimal (.).

  d_digits
  Uma sequência contígua de caracteres de dígito (0-9). Pelo menos um caractere de dígito deve estar presente em *digits* ou *d_digits*.

  e_char
  Um caractere expoente, que é um *E* ou *e*.

  e_sign
  (Opcional) Sinal de mais (+) ou sinal de menos (-) para o expoente.

  e_digits
  Uma sequência contígua de caracteres de dígito (0-9) para o expoente. Pelo menos um caractere de dígito deve estar presente se a sequência contiver um caractere exponente.

  Após a sequência de caracteres de dígito ou os caracteres opcionais que representam um expoente, a sequência pode conter outros caracteres que não são caracteres de dígito, mas a conversão para assim que o primeiro desses caracteres for atingido. Supõe-se que a sequência represente um número de vírgula flutuante decimal com um expoente que é uma potência de 10.

  XMS pode retornar um erro se a sequência não estiver formatada corretamente.

- Ao converter um valor de propriedade numérico em uma sequência, incluindo um valor de propriedade com tipo de dados System.SByte, o valor é convertido para a representação de sequência do valor como um número decimal, não a sequência que contém o caractere ASCII para esse valor. Por exemplo, o número inteiro 65 será convertido para a sequência "65", não para a sequência "A".
- Ao converter um valor de propriedade matriz de bytes em uma sequência, cada byte é convertido nos 2 caracteres hexadecimais que representam o byte. Por exemplo, a matriz de bytes {0}xF1, 0x12, 0x00, 0xFF } é convertida para a sequência "F11200FF".

Conversões de um tipo de propriedade para outros tipos de dados são suportadas pelos métodos de ambas as classes Property e PropertyContext.

Conceitos relacionados

Atributos e Propriedades de Objetos
Um objeto XMS pode ter atributos e propriedades, que são características do objeto, que são implementados de diferentes maneiras

Tipos primitivos XMS
O XMS fornece equivalentes dos oito tipos primitivos Java (byte, short, int, long, float, double, char e boolean). Isso permite a troca de mensagens entre XMS e JMS sem que os dados sejam perdidos ou corrompidos.

Referências relacionadas

Mensagens de Mapa
O corpo de uma mensagem de mapa contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado.

Mensagens de Fluxo
O corpo de uma mensagem de fluxo contém um fluxo de valores, em que cada valor possui um tipo de dados associado.

Tipos de Dados para Elementos de Dados do Aplicativo
Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Iteradores

Um agente iterativo encapsula uma lista de objetos e um cursor que mantém a posição atual na lista. O conceito de um Iterador, conforme disponível em Message Service Client for C/C++ , é implementado usando a interface IEnumerator em Message Service Client for .NET.

Quando um agente iterativo é criado, a posição do cursor é anterior ao primeiro objeto. Um aplicativo usa um agente iterativo para recuperar cada objeto por sua vez.

A classe Iterator de Message Service Client for C/C++ é equivalente à classe Enumerator em Java.XMS O .NET é semelhante ao Java e usa uma interface IEnumerator.

Um aplicativo pode usar um IEnumerator para executar as tarefas a seguir:

- Para obter as propriedades de uma mensagem
- Para obter os pares nome-valor no corpo de uma mensagem de mapa
- Para procurar as mensagens em uma fila
- Para obter os nomes das propriedades de mensagem definidas pelo JMS suportadas por uma conexão

Identificadores de conjunto de caracteres codificados

No XMS .NET, todas as sequências são passadas usando a sequência .NET nativa Como isso tem uma codificação fixa, nenhuma informação adicional é necessária para interpretá-la. Portanto, a propriedade XMSC_CLIENT_CCSID não é necessária para os aplicativos XMS .NET.

XMS códigos de erro e de exceção

XMS usa um intervalo de códigos de erro para indicar falhas. Esses códigos de erro não são explicitamente listados nesta documentação porque eles podem variar de liberação para liberação. Apenas códigos de exceção XMS (no formato XMS_X_...) são documentados porque permanecem os mesmos em diferentes versões de XMS.

Construindo seus próprios aplicativos

Você constrói seus próprios aplicativos, como você constrói os aplicativos de amostra.

Construa seu aplicativo .NET , conforme descrito em “Construindo os aplicativos de amostra .NET” na página 21 tópicoseção, que também lista os pré-requisitos necessários para construir seus próprios aplicativos .NET . Para obter orientação adicional sobre como construir seus próprios aplicativos, use os makefiles fornecidos para cada aplicativo de amostra.

Sugestão: Para ajudar no diagnóstico de problemas no caso de uma falha, é possível achar útil compilar aplicativos com símbolos incluídos.

Referências relacionadas

Interfaces do .NET
Este tópicoseção documenta as interface de classe .NET e suas propriedades e métodos.

Propriedades de objetos XMS

Isso seçãocapítulo documenta as propriedades de objeto definidas por XMS

Reconexão automática do cliente do IBM WebSphere MQ por XMS

Configure seu cliente XMS para se reconectar automaticamente após uma falha de rede, gerenciador de filas ou servidor ao usar o Cliente IBM WebSphere MQ V7.1 ou superior como o provedor de mensagens.

Use as propriedades `WMQ_CONNECTION_NAME_LIST` e `WMQ_CLIENT_RECONNECT_OPTIONS` da classe `MQConnectionFactory` para configurar uma conexão do cliente para reconectar automaticamente. A reconexão de cliente automática reconecta um cliente após uma falha de conexão ou como uma opção após parar o gerenciador de filas. O design de alguns aplicativos clientes os torna inadequados para reconexão automática.

As conexões do cliente reconectáveis automaticamente tornam-se reconectáveis depois que a conexão for estabelecida.

Nota: As propriedades `Opções de Reconexão de Cliente`, `Tempo Limite de Reconexão de Clientee` `Lista de Nomes de Conexão` também podem ser configuradas por meio da Tabela de Definições de Canais de Cliente (CCDT) ou ativando a reconexão do cliente por meio do arquivo `mqclient.ini` .

Nota: Se as propriedades de reconexão forem configuradas no objeto `ConnectionFactory` e, assim como na CCDT, a regra de precedência será a seguinte. Se o valor padrão da propriedade da lista de nomes de conexão for configurado no objeto `ConnectionFactory` , então, a CCDT terá precedência. Se a lista de nomes de conexão não estiver configurada para seu valor padrão, os valores da propriedade definidos no objeto `ConnectionFactory` têm precedência. O valor padrão da lista de nomes de conexão é `localhost(1414)`.

Gravando aplicativos XMS .NET

Este seçãocapítulo fornece informações para ajudá-lo ao gravar aplicativos do XMS.NET

Este seçãocapítulo fornece informações que são específicas para gravar aplicativos do XMS .NET . Para obter informações gerais sobre a gravação de aplicativos XMS , consulte “Gravando aplicativos do XMS” na página 22.

O seçãocapítulo contém o seguinte tópicosseções:

Referências relacionadas

Interfaces do .NET
Este tópicoseção documenta as interface de classe .NET e suas propriedades e métodos.

Tipos de dados para .NET

O XMS .NET suporta System.Boolean, System.Byte, System.SByte, System.Char, System.String, System.Single, System.Double, System.Decimal, System.Int32, System.Int64, System.UInt16, System.UInt32, System.UInt64 e System.Object. Os tipos de dados para XMS .NET são diferentes dos tipos de dados para XMS C++. É possível usar esse seçãocapítulo para identificar os tipos de dados correspondentes.

A tabela a seguir mostra os tipos de dados XMS .NET e XMS C++ correspondentes e descreve-os brevemente:

Tabela 7. Tipos de dados para XMS .NET e XMS C++		
Tipo XMS .NET	**Tipo XMS C++**	**Descrição**
System.SByte	xmsSBYTE xmsINT8	Valor de 8 bits sinalizado
System.Byte	xmsBYTE xmsUINT8	Valor de 8 bits não assinado
System.Int16	xmsINT16 xmsSHORT	Valor de 16 bits sinalizado
System.UInt16	xmsUINT16 xmsUSHORT	Valor de 16 bits não assinado
System.Int32	xmsINT32 xmsINT	Valor de 32 bits assinado
System.UInt32	xmsUINT32 xmsUINT	Valor de 32 bits não assinado
System.Int64	xmsLONG xmsINT64	Valor de 64 Bits Assinado
System.UInt64	xmsULONG xmsUINT64	Valor de 64 bits não assinado
System.Char	xmsCHAR16	Caractere de 16 bits não assinado (Unicode para .NET)
System.Single	xmsFLOAT	IEEE 32-bit float
System.Double	xmsDOUBLE	IEEE 64-bit float
System.Boolean	xmsBOOL	Um valor True / False
Não-aplicável	xmsCHAR	Valor de 8-bit assinado ou não assinado (depende da plataforma)
System.Decimal	Não-aplicável	Número inteiro sinalizado de 96 bits 10^0 a 10^{28}
System.Object	Não-aplicável	Base de todos os tipos
System.String	Não-aplicável	Tipo de sequência

Operações gerenciadas e não gerenciada em .NET ...

O código gerenciado é executado exclusivamente dentro do ambiente Common Language Runtime do .NET e depende totalmente dos serviços fornecidos por esse tempo de execução. Um aplicativo é classificado como não gerenciado se qualquer parte do aplicativo executar ou chamar serviços fora do ambiente de tempo de execução de linguagem comum .NET .

Determinadas funcionalidades avançadas não podem ser suportadas atualmente dentro do ambiente gerenciado do .NET.

Se o seu aplicativo requer alguma funcionalidade que não é atualmente suportado no ambiente totalmente gerenciado, é possível alterar seu aplicativo para usar o ambiente não gerenciado sem

requerer uma mudança substancial no seu aplicativo. No entanto, é necessário observar que a pilha XMS faz uso de código não gerenciado quando essa seleção é feita.

Conexões com um IBM WebSphere MQ gerenciador de filas

Conexões gerenciadas para WMQ_CM_CLIENT não suportarão conexões SSL, comunicações não TCP e compactação de canal. No entanto, essas conexões podem ser suportadas pelo uso de uma conexão não gerenciada (WMQ_CM_CLIENT_UNMANAGED). Para obter mais informações, veja Desenvolvendo aplicativos .NET.

Se você criar um connection factory a partir de um objeto administrado em um ambiente não gerenciado, você deverá alterar manualmente o valor para o modo de conexão para XMSC_WMQ_CM_CLIENT_UNMANAGED.

Conexões com um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus

Conexões com um mecanismo de sistema de mensagens do barramento de integração de serviços WebSphere que requerem o uso do protocolo SSL (incluindo HTTPS) não são atualmente suportadas como código gerenciado.

Referências relacionadas
XMSC_WMQ_CONNECTION_MODE

Destinos em .NET

Em .NET, os destinos são criados de acordo com o tipo de protocolo e podem ser usados apenas no tipo de protocolo para o qual eles são criados.

Duas funções são fornecidas para criar destinos, um para tópicos e um para filas:

- `IDestination CreateTopic(String topic);`
- `IDestination CreateQueue(String queue);`

Essas funções estão disponíveis nos dois objetos a seguir na API:

- ISession
- XMSFactoryFactory

Em ambos os casos, esses métodos podem aceitar uma sequência de estilo de URI, que pode incluir parâmetros, no formato a seguir:

```
"topic://some/topic/name?priority=5"
```

Como alternativa, esses métodos podem aceitar um nome de destino apenas, ou seja, um nome sem um tópico: // ou queue: // prefix e sem parâmetros.

Isso significa que a sequência de estilo de URI a seguir:

```
CreateTopic("topic://some/topic/name");
```

produziria o mesmo resultado que o nome de destino a seguir:

```
CreateTopic("some/topic/name");
```

Como para o WebSphere Serviço Integration Bus JMS, os tópicos também podem ser especificados em um formato abreviado, que inclui o *topicname* e o *topicspace* , mas não pode incluir parâmetros:

```
CreateTopic("topicspace:topicname");
```

Propriedades em .NET

Um aplicativo .NET usa os métodos na interface PropertyContext para obter e configurar as propriedades de objetos.

A interface PropertyContext encapsula métodos que obtem e configuram propriedades. Esses métodos são herdados, direta ou indiretamente, pelas classes a seguir:

- BytesMessage
- Conexão
- ConnectionFactory
- ConnectionMetaData
- Destino
- MapMessage
- Mensagem
- MessageConsumer
- MessageProducer
- ObjectMessage
- QueueBrowser
- Sessão
- StreamMessage
- TextMessage

Se um aplicativo configurar o valor de uma propriedade, o novo valor substituirá qualquer valor anterior que a propriedade possuiu.

Para obter mais informações sobre propriedades XMS , consulte “Propriedades de objetos XMS” na página 183.

Para facilidade de uso, os nomes e valores da propriedade XMS em .NET são predefinidos como constantes públicas em uma estrutura chamada XMSC. Os nomes dessas constantes estão no formato XMSC.< *constant*>; por exemplo, XMSC.USERID (uma constante de nome da propriedade) e XMSC.DELIVERY_AS_APP (uma constante de valor)

Além disso, é possível acessar constantes IBM WebSphere MQ usando o IBM.XMS.MQC struct. Se a IBM.XMS já foi importado, é possível acessar os valores para essas propriedades no formato MQC.< *constant*> Por exemplo, MQC.MQRO_COA_WITH_FULL_DATA..

Além disso, se você tiver um aplicativo híbrido que use as classes XMS .NET e IBM WebSphere MQ para .NET e que importe ambos os espaços de nomes IBM.XMS e IBM.WMQ, então você deve qualificar totalmente o espaço de nomes de estrutura MQC para garantir que cada ocorrência seja única.

Algumas funcionalidades avançadas não são atualmente suportadas dentro do ambiente gerenciado .NET . Consulte “Operações gerenciadas e não gerenciada em .NET ...” na página 47 para obter mais detalhes.

Manipulação de propriedades não existentes em .NET

O manuseio de propriedades inexistentes no XMS .NET é amplamente consistente com a especificação JMS, e também com as implementações C e C++ de XMS.

No JMS, o acesso a uma propriedade não existente pode resultar em uma exceção do sistema Java quando um método tenta converter o valor não existente (nulo) para o tipo necessário. Se uma propriedade não existir, ocorrerão as seguintes exceções:

- getStringProperty e getObjectProperty return null
- getBooleanProperty retorna false porque Boolean.valueOf (null) retorna false

- getIntProperty etc.throw java.lang.NumberFormatException porque Integer.valueOf (null) throws a exceção

Se uma propriedade não existir no XMS .NET, ocorrerão as exceções a seguir:

- GetStringProperty e GetObjectProperty (e GetBytesProperty) retornam nulo (que é o mesmo que Java)
- GetBooleanProperty throws System.NullReferenceException
- GetIntProperty etc. throws throws System.NullReferenceException

Essa implementação é diferente de Java, mas ela é amplamente consistente com a especificação JMS, e com as interfaces C e C++ XMS. Assim como a implementação Java, XMS .NET propaga quaisquer exceções da chamada System.Convert para o responsável pela chamada. Ao contrário de Java, no entanto, XMS lança explicitamente NullReferenceExceptions em vez de apenas usar o comportamento nativo da estrutura .NET através de passagem de nulo para rotinas de conversão do sistema. Se seu aplicativo configurar uma propriedade para uma Cadeia como "abc" e chamar GetIntProperty, o System.FormatException lançado pelo Convert.ToInt32 ("abc") será propagado para o responsável pela chamada, o que é consistente com Java. MessageFormatException será lançada apenas se os tipos usados para setProperty e getProperty forem incompatíveis. Esse comportamento também é consistente com Java.

Manipulação de erros em .NET

As exceções XMS.NET são todas derivadas de System.Exception. Chamadas de método XMS podem emitir exceções XMS específicas, como MessageFormatException, General XMSExceptions ou exceções do sistema, como NullReferenceException.

Grave aplicativos para capturar qualquer um desses erros, em blocos de captura específicos ou em blocos de captura gerais `System.Exception` , conforme apropriado para os requisitos de aplicativos.

Listeners de mensagem e de exceção em .NET

Um aplicativo .NET usa um listener de mensagem para receber mensagens de forma assíncrona, e ele usa um listener de exceção para ser notificado de forma assíncrona de um problema com uma conexão.

A funcionalidade de ambos os listeners de mensagem e exceção é a mesma para .NET e para C++. No entanto, existem algumas pequenas diferenças de implementação.

Listeners de mensagens no .NET

Para receber mensagens de forma assíncrona, você deve concluir as seguintes etapas:

1. Defina um método que corresponda à assinatura do delegado de listener de mensagem. O método que você define pode ser um método estático ou de instância e pode ser definido em qualquer classe acessível. A assinatura do delegado é a seguinte:

   ```
   public delegate void MessageListener(IMessage msg);
   ```

   e, portanto, você poderia definir o método como:

   ```
   void SomeMethodName(IMessage msg);
   ```

2. Instancie este método como um delegado usando algo semelhante ao seguinte:

   ```
   MessageListener OnMsgMethod = new MessageListener(SomeMethodName)
   ```

3. Registre o delegado com um ou mais consumidores, configurando-o para a propriedade MessageListener do consumidor:

   ```
   consumer.MessageListener = OnMsgMethod;
   ```

É possível remover o delegado, reconfigurando o MessageListener para nulo:

```
consumer.MessageListener = null;
```

Listeners de exceção no .NET

O listener de exceção funciona da mesma maneira que o listener de mensagem, mas possui uma definição de delegação diferente e é designado para a conexão, em vez disso, o consumidor de mensagens. Isso é o mesmo que para C + +.

1. Defina o método. A assinatura do delegado é a seguinte:

   ```
   public delegate void ExceptionListener(Exception ex);
   ```

   e, portanto, o método definido poderia ser:

   ```
   void SomeMethodName(Exception ex);
   ```

2. Instancie esse método como um delegado usando algo semelhante a:

   ```
   ExceptionListener OnExMethod = new ExceptionListener(SomeMethodName)
   ```

3. Registre o delegado com a conexão configurando sua propriedade ExceptionListener:

   ```
   connection.ExceptionListener = OnExMethod ;
   ```

   É possível remover o delegado, reconfigurando o ExceptionListener para:

   ```
   null:  connection.ExceptionListener = null;
   ```

Quando nenhuma referência a eles permanecer, as exceções ou mensagens serão excluídas automaticamente pelo coletor de lixo do sistema

A seguir está um código de amostra:.

```
using System;
using System.Threading;
using IBM.XMS;

public class Sample
{
    public static void Main()
    {
        XMSFactoryFactory factoryFactory = XMSFactoryFactory.GetInstance(XMSC.CT_RTT);

        IConnectionFactory connectionFactory = factoryFactory.CreateConnectionFactory();
        connectionFactory.SetStringProperty(XMSC.RTT_HOST_NAME, "localhost");
        connectionFactory.SetStringProperty(XMSC.RTT_PORT, "1506");

        //
        // Create the connection and register an exception listener
        //

        IConnection connection = connectionFactory.CreateConnection();
        connection.ExceptionListener = new ExceptionListener(Sample.OnException);

        ISession session = connection.CreateSession(false, AcknowledgeMode.AutoAcknowledge);
        IDestination topic = session.CreateTopic("topic://xms/sample");

        //
        // Create the consumer and register an async message listener
        //

        IMessageConsumer consumer = session.CreateConsumer(topic);
        consumer.MessageListener = new MessageListener(Sample.OnMessage);

        connection.Start();

        while (true)
        {
            Console.WriteLine("Waiting for messages....");
```

```
            Thread.Sleep(1000);
        }
    }

    static void OnMessage(IMessage msg)
    {
        Console.WriteLine(msg);
    }

    static void OnException(Exception ex)
    {
        Console.WriteLine(ex);
    }
}
```

Trabalhando com Objetos Administrados

Este seçãocapítulo fornece informações sobre objetos administrados.. Os aplicativos XMS podem recuperar as definições de objeto de um repositório de objetos administrados centrais e usá-los para criar connection factories e destinos.

Este seçãocapítulo fornece informações para ajudar a criar e gerenciar objetos administrados, descrevendo os tipos de repositório de objetos administrados que o XMS suporta. O seçãocapítulo também explica como um aplicativo XMS faz uma conexão com um repositório de objetos administrados para recuperar os objetos administrado necessários

O seçãocapítulo contém o seguinte tópicosseções:

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Tipos Suportados de Repositório de Objetos Administrados

O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Os diretórios de objetos do Sistema de Arquivos assumem a forma de objetos JNDI (Java and Naming Directory Interface) serializadas.. Os diretórios do objeto LDAP são diretórios que contêm objetos JNDI Os diretórios de objeto do Sistema de Arquivos e LDAP podem ser administrados usando a ferramenta JMSAdmin, que é fornecida com o IBM WebSphere MQ v6.0, ou o WebSphere MQ Explorer, que é fornecido com o WebSphere MQ v7.0 e posterior. Os diretórios do sistema de arquivos e do objeto LDAP podem ser usados para administrar conexões do cliente centralizando connection factories e destinos do IBM WebSphere MQ . O administrador de rede pode implementar vários aplicativos que se referem ao mesmo repositório central e que são automaticamente atualizados para refletir as mudanças nas configurações de conexão feitas no repositório central.

Um CORBA Naming Directory contém fábricas de conexão WebSphere Serviço Integration Bus e destinos e pode ser administrado usando o console administrativo WebSphere Servidor de Aplicação. Para um aplicativo XMS para recuperar objetos do diretório de nomenclatura COS, um serviço da web de consulta JNDI deve ser implementado. Esse serviço da web não está disponível em todos os WebSphere Tecnologias de integração de serviços. Consulte a documentação do produto para obter detalhes.

Nota: Reinicie as conexões do aplicativo para que as mudanças no diretório de objeto tenham efeito.

Conceitos relacionados

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições

devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Objetos Administrados
Usando objetos administrados, é possível administrar as configurações de conexão usadas pelos aplicativos clientes a serem administradas a partir de um repositório central. Um aplicativo recupera definições de objeto do repositório central e as utiliza para criar objetos `ConnectionFactory` e `Destination` . Usando objetos administrados, é possível desacoplar aplicativos dos recursos usados no tempo de execução.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

Mapeamento de Propriedades para Objetos Administrados

Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Para criar, por exemplo, um connection factory XMS com propriedades recuperadas de um connection factory JMS IBM WebSphere MQ , as propriedades devem ser mapeadas entre os dois.

Todos os mapeamentos de propriedade são executados automaticamente.

A tabela a seguir demonstra os mapeamentos entre algumas das propriedades mais comuns de connection factories e destinos. As propriedades mostradas nesta tabela são apenas um pequeno

conjunto de exemplos, e nem todas as propriedades mostradas são relevantes para todos os tipos de conexão e servidores.

Tabela 8. Exemplos de mapeamento de nome para connection factory e propriedades de destino	
IBM WebSphere MQ Nome da propriedade JMS	**Nome da propriedade XMS**
PERSISTÊNCIA (POR)	XMSC_DELIVERY_MODE
EXPIRAÇÃO (EXP)	XMSC_TIME_TO_LIVE
PRIORIDADE (PRI)	XMSC_PRIORITY

Tabela 9. Exemplos de mapeamento de nome para connection factory e propriedades de destino		
IBM WebSphere MQ Nome da propriedade JMS	**Nome da propriedade XMS**	**Nome da propriedade WebSphere Serviço Integration Bus**
PERSISTÊNCIA (POR)	XMSC_DELIVERY_MODE	
EXPIRAÇÃO (EXP)	XMSC_TIME_TO_LIVE	
PRIORIDADE (PRI)	XMSC_PRIORITY	
	XMSC_WPM_HOST_NAME	serverName
	XMSC_WPM_BUS_NAME	busName
	XMSC_WPM_TOPIC_SPACE	topicName

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Tarefas relacionadas

Criando Objetos Administrados

As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

IConnectionFactory (para a interface .NET)
Um aplicativo usa um connection factory para criar uma conexão.

Propriedades de ConnectionFactory
Uma visão geral das propriedades do objeto ConnectionFactory , com links para informações de referência mais detalhadas.

Propriedades necessárias para objetos ConnectionFactory administrados

Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

As propriedades listadas nas tabelas a seguir são o mínimo necessário para que um aplicativo seja configurado para criar uma conexão com um servidor de sistema de mensagens. Se você desejar customizar a maneira que uma conexão é criada, seu aplicativo poderá configurar quaisquer propriedades adicionais do objeto ConnectionFactory, conforme necessário. Para obter mais informações, consulte “Propriedades de ConnectionFactory” na página 185. Uma lista completa de propriedades disponíveis está incluída.

Conexão com um gerenciador do IBM WebSphere MQ

Tabela 10. Configurações de propriedade para objetos ConnectionFactory administrados para conexões com um gerenciador de filas IBM WebSphere MQ

NecessárioXMS	**Propriedade JMS IBM WebSphere MQ equivalente necessária**
XMSC_CONNECTION_TYPE	XMS trabalha isso a partir do nome da classe do connection factory e da propriedade TRANSPORT (TRAN).
XMSC_WMQ_HOST_NAME	HOSTNAME (HOST)
XMSC_WMQ_PORT	PORTA
XMSC_WMQ_QUEUE_MANAGER	Nome do Gerenciador de Filas

Conexão de tempo real com um broker

Tabela 11. Configurações de propriedade para objetos ConnectionFactory administrados para conexões em tempo real com um broker	
NecessárioXMS	**Propriedade JMS IBM WebSphere MQ equivalente necessária**
XMSC_CONNECTION_TYPE	XMS trabalha isso a partir do nome da classe do connection factory e da propriedade TRANSPORT (TRAN).
XMSC_RTT_HOST_NAME	HOSTNAME (HOST)
XMSC_RTT_PORT	PORTA

Conexão com um WebSphere Serviço Integration Bus

Tabela 12. Configurações de propriedade para objetos ConnectionFactory administrados para conexões com um WebSphere Serviço Integration Bus	
Propriedade XMS	**Descrição**
XMSC_CONNECTION_TYPE	O tipo de servidor de mensagens para o qual um aplicativo se conecta.. Isso é determinado a partir do nome da classe do connection factory.
XMSC_WPM_BUS_NAME	Para um connection factory, o nome do barramento de integração de serviços ao qual o aplicativo se conecta ou, para um destino, o nome do barramento de integração de serviços no qual o destino existe.

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Recuperação de Objetos Administrados

XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Conexões seguras para um IBM WebSphere MQ gerenciador de filas
Para permitir que um aplicativo XMS .NET faça conexões seguras para um IBM WebSphere MQ gerenciador de filas, as propriedades relevantes devem ser definidas no objeto ConnectionFactory .

Conexões seguras para um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus
Para ativar um XMS NET para fazer conexões seguras com um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus , as propriedades relevantes devem ser definidas no objeto ConnectionFactory .

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Referências relacionadas

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

IConnectionFactory (para a interface .NET)
Um aplicativo usa um connection factory para criar uma conexão.

Propriedades de ConnectionFactory
Uma visão geral das propriedades do objeto ConnectionFactory , com links para informações de referência mais detalhadas.

Propriedades Necessárias para Objetos de Destino Administrados

Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

Tabela 13. IBM WebSphere MQ Configurações de Propriedade JMS para Objetos de Destino Administrados

Tipo de conexão	Propriedade	Descrição
IBM WebSphere MQ gerenciador de filas	FILA (QU) TÓPICO (SUPERIOR)	A fila à qual você deseja se conectar O tópico que o aplicativo usa como um destino
Conexão de tempo real com um broker	TÓPICO (SUPERIOR)	O tópico que o aplicativo usa como um destino

Tabela 14. Configurações de propriedade para objetos de Destino administrados

Tipo de conexão	Propriedade	Descrição
IBM WebSphere MQ gerenciador de filas	FILA (QU) TÓPICO (SUPERIOR)	A fila à qual você deseja se conectar O tópico que o aplicativo usa como um destino
Conexão de tempo real com um broker	TÓPICO (SUPERIOR)	O tópico que o aplicativo usa como um destino

Tabela 14. Configurações de propriedade para objetos de Destino administrados (continuação)		
Tipo de conexão	**Propriedade**	**Descrição**
WebSphere Serviço Integration Bus	topicName	Se seu aplicativo estiver se conectando a um tópico
	queueName	Se seu aplicativo estiver se conectando a uma fila

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino

Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Criando Objetos Administrados

As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Antes de começar

Para obter detalhes adicionais sobre os diferentes tipos de repositório de objeto administrado que o XMS suporta, consulte "Tipos Suportados de Repositório de Objetos Administrados" na página 52.

Sobre esta tarefa

Para criar os objetos administrados para o IBM WebSphere MQ , use a ferramenta IBM WebSphere MQ Explorer ou IBM WebSphere MQ JMS Administration (JMSAdmin).

Para criar os objetos administrados para IBM WebSphere MQ, WebSphere Event Brokerou WebSphere Message Broker, use a ferramenta IBM WebSphere MQ JMS Administration (JMSAdmin).

Para criar objetos administrados para WebSphere Serviço Integration Bus, use o console administrativo WebSphere Servidor de Aplicação .

As etapas a seguir resumem o que você faz para criar objetos administrados.

Procedimento

1. Crie um connection factory e defina as propriedades necessárias para criar uma conexão de seu aplicativo com o servidor escolhido.

   As propriedades mínimas que o XMS requer para fazer uma conexão são definidas em "Propriedades necessárias para objetos ConnectionFactory administrados" na página 55.
2. Crie o destino necessário no servidor de sistema de mensagens, ao qual seu aplicativo se conecta:
   - Para uma conexão com um IBM WebSphere MQ gerenciador de filas, crie uma fila ou tópico.
   - Para uma conexão em tempo real com um broker, crie um tópico.
   - Para uma conexão com um WebSphere Serviço Integration Bus, crie uma fila ou um tópico.

   As propriedades mínimas que o XMS requer para fazer uma conexão são definidas em "Propriedades Necessárias para Objetos de Destino Administrados" na página 57.

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Objetos Administrados
Usando objetos administrados, é possível administrar as configurações de conexão usadas pelos aplicativos clientes a serem administradas a partir de um repositório central. Um aplicativo recupera definições de objeto do repositório central e as utiliza para criar objetos `ConnectionFactory` e `Destination` . Usando objetos administrados, é possível desacoplar aplicativos dos recursos usados no tempo de execução.

Trabalhando com Objetos Administrados
Este seçãocapítulo fornece informações sobre objetos administrados.. Os aplicativos XMS podem recuperar as definições de objeto de um repositório de objetos administrados centrais e usá-los para criar connection factories e destinos.

ConnectionFactories e objetos de Conexão
Um objeto ConnectionFactory fornece um modelo que um aplicativo usa para criar um objeto Connection. O aplicativo usa o objeto Connection para criar um objeto Session.

Conexão com um Barramento de Integração de Serviços do WebSphere
O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

Tarefas relacionadas

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

IConnectionFactory (para a interface .NET)
Um aplicativo usa um connection factory para criar uma conexão.

Propriedades de ConnectionFactory
Uma visão geral das propriedades do objeto ConnectionFactory , com links para informações de referência mais detalhadas.

IDestination (para a interface .NET)
Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Propriedades de Destino
Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Objetos InitialContext

Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Sobre esta tarefa

Um objeto InitialContext encapsula uma conexão com o repositório. A API XMS fornece métodos para executar as tarefas a seguir:

- Criar um objeto InitialContext
- Consultar um objeto administrado no repositório de objetos administrado.

Para obter detalhes adicionais sobre a criação de um objeto InitialContext , consulte "InitialContext" na página 114 para .NET e "Propriedades de InitialContext" na página 195

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades Necessárias para Objetos de Destino Administrados

Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

InitialContext (para a interface .NET)
Um aplicativo usa um objeto InitialContext para criar objetos de definições de objeto que são recuperados de um repositório de objetos administrados.

Propriedades de InitialContext
Uma visão geral das propriedades do objeto InitialContext com links para informações de referência mais detalhadas.

Propriedades InitialContext

Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

No JNDI e na implementação .NET de XMS, as informações adicionais são fornecidas em um ambiente Hashtable ao construtor.

O local do repositório de objetos administrado é definido na propriedade XMSC_IC_URL . Normalmente, essa propriedade é transmitida na chamada Criar, mas pode ser modificada para se conectar a um diretório de nomenclatura diferente antes da consulta. Para contextos FileSystem ou LDAP, essa propriedade define o endereço do diretório. Para nomenclatura COS, esse é o endereço do serviço da web que usa essas propriedades para se conectar ao diretório JNDI.

As propriedades a seguir são transmitidas sem modificação para o serviço da web que as usará para usar para se conectar ao diretório JNDI.

- XMSC_IC_PROVIDER_URL
- XMSC_IC_SECURITY_CREDENTIALS
- XMSC_IC_SECURITY_AUTHENTICATION
- XMSC_IC_SECURITY_PRINCIPAL
- XMSC_IC_SECURITY_PROTOCOL

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Recuperação de Objetos Administrados

XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

InitialContext (para a interface .NET)
Um aplicativo usa um objeto InitialContext para criar objetos de definições de objeto que são recuperados de um repositório de objetos administrados.

Propriedades de InitialContext
Uma visão geral das propriedades do objeto InitialContext com links para informações de referência mais detalhadas.

Formato de URI para contextos iniciais XMS

O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Contexto de FileSystem

Para o contexto FileSystem, a URL fornece o local do diretório baseado no sistema de arquivos. A estrutura da URL é conforme definido pelo RFC 1738, *Localizadores Uniformes de Recursos (URL)*: a URL tem o prefixo `file://` e a sintaxe após esse prefixo é uma definição válida de um arquivo que pode ser aberto no sistema no qual o XMS está executando.

Esta sintaxe pode ser específica da plataforma e pode usar os separadores '/ separadores ou' \ '. Se você usar '\', cada separador precisará ser escapado usando um '\' adicional. Isso impede que a estrutura .NET tente interpretar o separador como um caractere de escape para o que segue.

Esses exemplos ilustram esta sintaxe:

```
file://myBindings
file:///admin/.bindings
file://\\admin\\.bindings
file://c:/admin/.bindings
file://c:\\admin\\.bindings
file://\\\\madison\\shared\\admin\\.bindings
file:///usr/admin/.bindings
```

Contexto LDAP

Para o contexto LDAP, a estrutura básica da URL é conforme definido pelo RFC 2255, *The LDAP URL Format*, com o prefixo sem distinção entre maiúsculas e minúsculas `ldap://`

A sintaxe exata é ilustrada no exemplo a seguir:

```
LDAP://[Hostname][:Port]["/"[DistinguishedName]]
```

Essa sintaxe é definida no RFC, mas sem suporte para quaisquer atributos, escopo, filtros ou extensões.

Exemplos desta sintaxe incluem:

```
ldap://madison:389/cn=JMSData,dc=IBM,dc=UK
ldap://madison/cn=JMSData,dc=IBM,dc=UK
LDAP:///cn=JMSData,dc=IBM,dc=UK
```

Contexto de WSS

Para o contexto WSS, a URL está no formato de um terminal de serviços da Web, com o prefixo `http://`.

Como alternativa, é possível usar o prefixo `cosnaming://` ou `wsvc://`,

Essas duas prefixos são interpretadas como significando que você está usando um contexto WSS com a URL acessada através de http, o que permite que o tipo de contexto inicial seja derivado facilmente diretamente da URL.

Exemplos dessa sintaxe incluem o seguinte:

```
http://madison.ibm.com:9080/xmsjndi/services/JndiLookup
cosnaming://madison/jndilookup
```

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

InitialContext (para a interface .NET)
Um aplicativo usa um objeto InitialContext para criar objetos de definições de objeto que são recuperados de um repositório de objetos administrados.

Propriedades de InitialContext
Uma visão geral das propriedades do objeto InitialContext com links para informações de referência mais detalhadas.

Serviço da Web de Consulta de JNDI

Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

O serviço da Web é fornecido no archive corporativo SIBXJndiLookupEAR.ear, localizado dentro do diretório de instalação Para a liberação atual do Message Service Client for .NET, SIBXJndiLookupEAR.ear pode ser localizado no diretório `<install_dir>\java\lib` Isso pode ser instalado dentro de um servidor de barramento de integração de serviços do WebSphere usando o console administrativo ou a ferramenta de script wsaadmin Consulte a documentação do produto para obter informações adicionais sobre como implementar aplicativos de serviço da web.

Para definir o serviço da web dentro de aplicativos XMS , você simplesmente precisa configurar a propriedade XMSC_IC_URL do objeto InitialContext para a URL do terminal de serviço da web. Por exemplo, se o serviço da web for implementado em um host do servidor chamado MyServer, um exemplo de uma URL de terminal de serviço da web:

```
wsvc://MyHost:9080/SIBXJndiLookup/services/JndiLookup
```

Configurar a propriedade XMSC_IC_URL permite que chamadas InitialContext Lookup chamem o serviço da web no terminal definido que, por sua vez, consulta o objeto administrado necessário por meio do serviço de nomenclatura COS.

Os aplicativos .NET podem usar o serviço da web. A implementação do lado do servidor é a mesma para XMS C, /C++ e, XMS.NET.XMS.NET chama o serviço da web diretamente por meio da estrutura Microsoft .NET .

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão

com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Instalando Message Service Client for .NET usando o assistente de instalação
A instalação usa um instalador MSI InstallShield X/Windows. Duas opções de configuração estão disponíveis, para que seja possível escolher uma instalação completa ou customizada.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

Recuperação de Objetos Administrados

XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Os objetos a serem recuperados podem ter os seguintes tipos de nomes:

- Um nome simples que descreve o objeto Destination, por exemplo, um destino de fila chamado SalesOrders
- Um nome composto, que pode ser composto de SubContextos, separados por '/' e deve terminar com o nome do objeto. Um exemplo de um nome composto é "Warehouse / PickLists/DispatchQueue2", em que Warehouse e Picklists são SubContextos no diretório de nomenclatura e DispatchQueue2 é o nome de um objeto de Destino.

Conceitos relacionados

Tipos Suportados de Repositório de Objetos Administrados
O XMS suporta três tipos de diretórios de objetos administrados: Sistema de Arquivos, Lightweight Directory Access Protocol (LDAP) e Nomenclatura COS. Os objetos administrados do Sistema de Arquivos e do LDAP podem ser usados para se conectar ao IBM WebSphere MQ e ao WebSphere Servidor de Aplicação, enquanto que o COS Naming pode ser usado para se conectar apenas ao WebSphere Servidor de Aplicação.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Propriedades InitialContext

Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Serviço da Web de Consulta de JNDI
Para acessar um diretório de nomenclatura do COS do XMS, um serviço da web de Consulta JNDI deve ser implementado em um servidor WebSphere Serviço Integration Bus . Este serviço da web converte as informações Java do serviço de nomenclatura do COS em um formulário que os aplicativos XMS podem ler.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

InitialContext (para a interface .NET)
Um aplicativo usa um objeto InitialContext para criar objetos de definições de objeto que são recuperados de um repositório de objetos administrados.

Propriedades de InitialContext
Uma visão geral das propriedades do objeto InitialContext com links para informações de referência mais detalhadas.

Protegendo comunicações para aplicativos XMS

Este seçãocapítulo fornece informações sobre como configurar comunicações seguras para permitir que os aplicativos XMS se conectem via Secure Sockets Layer (SSL) a um WebSphere Serviço Integration Bus mecanismo do sistema de mensagens ou IBM WebSphere MQ gerenciador de filas.

Este seçãocapítulo fornece informações sobre como configurar as propriedades XMS ConnectionFactory para permitir que os aplicativos façam conexões seguras.

O seçãocapítulo contém o seguinte tópicosseções:

Conexões seguras para um IBM WebSphere MQ gerenciador de filas

Para permitir que um aplicativo XMS .NET faça conexões seguras para um IBM WebSphere MQ gerenciador de filas, as propriedades relevantes devem ser definidas no objeto ConnectionFactory .

O protocolo usado na negociação de criptografia pode ser Secure Sockets Layer (SSL) ou Segurança da Camada de Transporte (TLS), dependendo de qual CipherSuite você especificar no objeto ConnectionFactory.

Se você usar a IBM WebSphere MQ Versão 7.0.0.1 e bibliotecas do cliente acima e se conectar a um gerenciador de fila do IBM WebSphere MQ Versão 7, será possível criar várias conexões com o mesmo

gerenciador de filas no aplicativo XMS . No entanto, a conexão com um gerenciador de filas diferente não é permitida. Se você tentar, receberá o erro `MQRC_SSL_ALREADY_INITIALIZED` .

Se você usar o IBM WebSphere MQ Versão 6 e as bibliotecas do cliente acima, será possível criar uma conexão SSL somente se você fechar qualquer conexão SSL anterior primeiro Várias conexões SSL simultâneas a partir do mesmo processo para os mesmos gerenciadores de filas ou gerenciadores de filas diferentes não são permitidas. Se você tentar mais de uma solicitação, receberá o aviso `MQRC_SSL_ALREADY_INITIALIZED`, o que pode significar que alguns parâmetros solicitados para a conexão SSL foram ignorados.

As propriedades ConnectionFactory para conexões via SSL para um gerenciador do IBM WebSphere MQ , com uma breve descrição, são mostradas na tabela a seguir:

Tabela 15. Propriedades de ConnectionFactory para conexões com um IBM WebSphere MQ gerenciador de filas via SSL

Nome da propriedade	**Descrição**
XMSC_WMQ_SSL_CERT_STORES	Os locais dos servidores que retêm as listas de revogação de certificado (CRLs) a serem usadas em uma conexão SSL com um gerenciador de filas.
XMSC_WMQ_SSL_CIPHER_SPEC	O nome da CipherSpec a ser usada em uma conexão segura com um gerenciador de filas.
XMSC_WMQ_SSL_CIPHER_SUITE	O nome do CipherSuite a ser usado em uma conexão SSL ou TLS com um gerenciador de filas O protocolo usado para negociar a conexão segura depende do CipherSuite especificado.
XMSC_WMQ_SSL_CRYPTO_HW	Detalhes de configuração para o hardware criptográfico conectado ao sistema do cliente.
XMSC_WMQ_SSL_FIPS_REQUIRED	O valor dessa propriedade determina se um aplicativo pode ou não usar conjuntos de cifras compatíveis não FIPS. Se essa propriedade for configurada como true, apenas algoritmos do FIPS serão usados para a conexão cliente-servidor.
XMSC_WMQ_SSL_KEY_REPOSITORY	O local do arquivo do banco de dados de chaves no qual chaves e certificados são armazenados.
XMSC_WMQ_SSL_KEY_RESETCOUNT	O KeyResetCount representa o número total de bytes não criptografados enviados e recebidos dentro de uma conversa SSL antes de a chave secreta ser renegociada.
XMSC_WMQ_SSL_PEER_NAME	O nome do peer a ser usado em uma conexão SSL com um gerenciador de filas.

Referências relacionadas

IConnectionFactory (para a interface .NET)
Um aplicativo usa um connection factory para criar uma conexão.

Propriedades de ConnectionFactory
Uma visão geral das propriedades do objeto ConnectionFactory , com links para informações de referência mais detalhadas.

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Mapeamentos de Nome CipherSuite e CipherSpec para Conexões com um IBM WebSphere MQgerenciador de filas

O InitialContext é convertido entre a propriedade SSLCIPHERSUITE da Connection Factory de JMSAdmin e o XMS próximo-equivalente XMSC_WMQ_SSL_CIPHER_SPEC. Uma conversão semelhante é necessária se você especificar um valor para XMSC_WMQ_SSL_CIPHER_SUITE mas omitir valor para XMSC_WMQ_SSL_CIPHER_SPEC.

Tabela 16 na página 69 lista os CipherSpecs disponíveis e seus equivalentes JSSE CipherSuite .

Tabela 16. CipherSpecs Disponíveis e seus Equivalentes JSSE CipherSuite

CipherSpec	**Equivalente CipherSuite JSSE**
DES_SHA_EXPORT	SSL_RSA_WITH_DES_CBC_SHA
DES_SHA_EXPORT1024	SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA
FIPS_WITH_3DES_EDE_CBC_SHA	SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
FIPS_WITH_DES_CBC_SHA	SSL_RSA_FIPS_WITH_DES_CBC_SHA
NULL_MD5	SSL_RSA_WITH_NULL_MD5
NULL_SHA	SSL_RSA_WITH_NULL_SHA
RC2_MD5_EXPORT	SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
RC4_56_SHA_EXPORT1024	SSL_RSA_EXPORT1024_WITH_RC4_56_SHA
RC4_MD5_EXPORT	SSL_RSA_EXPORT_WITH_RC4_40_MD5
RC4_MD5_US	SSL_RSA_WITH_RC4_128_MD5
RC4_SHA_US	SSL_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA	SSL_RSA_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA	SSL_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA	SSL_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_DES_CBC_SHA	SSL_RSA_WITH_DES_CBC_SHA
TRIPLE_DES_SHA_US	SSL_RSA_WITH_3DES_EDE_CBC_SHA

Nota: Um mapeamento um para um para o CipherSuite name SSL_RSA_WITH_3DES_EDE_CBC_SHA ou SSL_RSA_WITH_DES_CBC_SHA deve considerar a configuração da propriedade XMSC_WMQ_SSL_FIPSREQUIRED e aplicar uma heurística.

Se especificar SSL_RSA_WITH_3DES_EDE_CBC_SHA ou SSL_RSA_WITH_DES_CBC_SHA para a propriedade XMSC_WMQ_SSL_CIPHER_SUITE, e não houver valor para XMSC_WMQ_SSL_CIPHER_SPEC, um valor para XMSC_WMQ_SSL_CIPHER_SPEC será escolhido de acordo com as tabelasa seguir.

Os valores usados para XMSC_WMQ_SSL_CIPHER_SPEC ao especificar SSL_RSA_WITH_3DES_EDE_CBC_SHA para a propriedade XMSC_WMQ_SSL_CIPHER_SUITE são mostrados na tabela a seguir:

Tabela 17. Valores usados para XMSC_WMQ_SSL_CIPHER_SPEC ao especificar SSL_RSA_WITH_3DES_EDE_CBC_SHA para a propriedade XMSC_WMQ_SSL_CIPHER_SUITE

Entrada: Valor XMSC_WMQ_SSL_FIPSREQUIRED	**Saída: XMSC_WMQ_SSL_CIPHER_SPEC escolhido**
false (ou seja, MQSSL_FIPS_NO)	TRIPLE_DES_SHA_US
true (ou seja, MQSSL_FIPS_YES)	TLS_RSA_WITH_3DES_EDE_CBC_SHA

Nota:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA foi descontinuado. No entanto, ele ainda pode ser usado para transferir até 32 GB de dados antes de a conexão ser finalizada com erro AMQ9288. Para evitar esse erro, você precisará evitar o uso de DES triplo ou ativar a reconfiguração de chave secreta ao usar esse CipherSpec.

Os valores usados para XMSC_WMQ_SSL_CIPHER_SPEC ao especificar SSL_RSA_WITH_DES_CBC_SHA para a propriedade XMSC_WMQ_SSL_CIPHER_SUITE são mostrados na tabela a seguir:

Tabela 18. Valores usados para XMSC_WMQ_SSL_CIPHER_SPEC ao especificar SSL_RSA_WITH_DES_CBC_SHA para a propriedade XMSC_WMQ_SSL_CIPHER_SUITE

Entrada: Valor XMSC_WMQ_SSL_FIPSREQUIRED	**Saída: XMSC_WMQ_SSL_CIPHER_SPEC escolhido**
false (ou seja, MQSSL_FIPS_NO)	DES_SHA_EXPORT
true (ou seja, MQSSL_FIPS_YES)	TLS_RSA_WITH_DES_CBC_SHA

Conexões seguras para um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus

Para ativar um XMS NET para fazer conexões seguras com um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus , as propriedades relevantes devem ser definidas no objeto ConnectionFactory .

XMS fornece suporte SSL e HTTPS para conexões com um WebSphere Serviço Integration Bus. SSL e HTTPS fornecem conexões seguras para autenticação e confidencialidade.

Assim como a segurança WebSphere , a segurança XMS é configurada com relação aos padrões de segurança JSSE e convenções de nomenclatura, que incluem o uso de CipherSuites para especificar os algoritmos que são usados ao negociar uma conexão segura. O protocolo usado na negociação de criptografia pode ser SSL ou TLS, dependendo de qual CipherSuite você especifica no objeto ConnectionFactory.

Nota: Para um aplicativo .NET, os recursos de segurança são fornecidos pelo Microsoft Secure Channel (SChannel).

Tabela 19 na página 70 lista as propriedades que devem ser definidas no objeto ConnectionFactory.

Tabela 19. Propriedades de ConnectionFactory para conexões seguras com um mecanismo do sistema de mensagens do barramento de integração de serviços do WebSphere

Nome da propriedade	**Descrição**
XMSC_WPM_SSL_CIPHER_SUITE	O nome do CipherSuite a ser usado em uma conexão SSL ou TLS para um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus O protocolo usado para negociar a conexão segura depende do CipherSuite especificado. **Nota:** Essa propriedade é suportada em um aplicativo .NET
XMSC_WPM_SSL_KEYRING_LABEL	O certificado a ser usado ao autenticar-se com o servidor. **Nota:** Essa propriedade é suportada em um aplicativo .NET

A seguir há um exemplo de propriedades ConnectionFactory para conexões seguras com um mecanismo do sistema de mensagens de integração doWebSphere :

```
cf.setStringProperty(XMSC_WPM_PROVIDER_ENDPOINTS, host_name:port_number:chain_name);
cf.setStringProperty(XMSC_WPM_SSL_KEY_REPOSITORY, key_repository_pathname);
cf.setStringProperty(XMSC_WPM_TARGET_TRANSPORT_CHAIN, transport_chain);
cf.setStringProperty(XMSC_WPM_SSL_CIPHER_SUITE, cipher_suite);
cf.setStringProperty(XMSC_WPM_SSL_KEYRING_STASH_FILE, stash_file_pathname);
```

Em que chain_name deve ser configurado como BootstrapTunneledeledSecureMessaging ou BootstrapSecureMessaging e port_number é o número da porta na qual o servidor de autoinicialização atende a solicitações recebidas.

A seguir está um exemplo de propriedades ConnectionFactory para conexões seguras para um mecanismo do sistema de mensagens de integração do WebSphere com valores de amostra inseridos:

```
  /* CF properties needed for an SSL connection */
    cf.setStringProperty(XMSC_WPM_PROVIDER_ENDPOINTS,"localhost:7286:BootstrapSecureMessaging");
    cf.setStringProperty(XMSC_WPM_TARGET_TRANSPORT_CHAIN,"InboundSecureMessaging");
    cf.setStringProperty(XMSC_WPM_SSL_KEY_REPOSITORY,"C:\\Program Files\\IBM\\gsk7\\bin\
\XMSkey.kdb");
    cf.setStringProperty(XMSC_WPM_SSL_KEYRING_STASH_FILE,"C:\\Program Files\\IBM\\gsk7\\bin\
\XMSkey.sth");
    cf.setStringProperty(XMSC_WPM_SSL_CIPHER_SUITE,"SSL_RSA_EXPORT_WITH_RC4_40_MD5");
```

Referências relacionadas

IConnectionFactory (para a interface .NET)
Um aplicativo usa um connection factory para criar uma conexão.

Propriedades de ConnectionFactory
Uma visão geral das propriedades do objeto ConnectionFactory , com links para informações de referência mais detalhadas.

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Mapeamentos de nomes CipherSuite e CipherSpec para conexões com um WebSphere Serviço Integration Bus

Como o GSKit usa CipherSpecs em vez de CipherSuites, os nomes de CipherSuite de estilo JSSE especificados na propriedade XMSC_WPM_SSL_CIPHER_SUITE devem ser mapeados para os nomes de CipherSpec de estilo GSKit.

Tabela 20 na página 71 lista o CipherSpec equivalente para cada CipherSuite reconhecido.

Tabela 20. CipherSuites Disponíveis e seus CipherSpecs equivalentes

CipherSuite	**CipherSpec equivalente**
SSL_RSA_WITH_NULL_MD5	NULL_MD5
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	RC2_MD5_EXPORT
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RC4_MD5_EXPORT
SSL_RSA_WITH_RC4_128_MD5	RC4_MD5_US
SSL_RSA_WITH_NULL_SHA	NULL_SHA
SSL_RSA_EXPORT1024_WITH_RC4_56_SHA	RC4_56_SHA_EXPORT1024
SSL_RSA_WITH_RC4_128_SHA	RC4_SHA_US
SSL_RSA_WITH_DES_CBC_SHA	DES_SHA_EXPORT
SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA	DES_SHA_EXPORT1024
SSL_RSA_FIPS_WITH_DES_CBC_SHA	FIPS_WITH_DES_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	TRIPLE_DES_SHA_US
SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA	FIPS_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_DES_CBC_SHA	TLS_RSA_WITH_DES_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA

Tabela 20. CipherSuites Disponíveis e seus CipherSpecs equivalentes (continuação)	
CipherSuite	**CipherSpec equivalente**
TLS_RSA_WITH_AES_128_CBC_SHA	TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA	TLS_RSA_WITH_AES_256_CBC_SHA

Nota:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA foi descontinuado. No entanto, ele ainda pode ser usado para transferir até 32 GB de dados antes de a conexão ser finalizada com erro AMQ9288. Para evitar esse erro, você precisará evitar o uso de DES triplo ou ativar a reconfiguração de chave secreta ao usar esse CipherSpec.

XMS

Este seçãocapítulo descreve a estrutura e o conteúdo de mensagens XMS e explica como os aplicativos processam mensagens XMS .

Esse seçãocapítulo contém o seguinte tópicosseções:

Referências relacionadas

IMessage (para a interface .NET)
Um objeto de mensagem representa uma mensagem que um aplicativo envia ou recebe. IMessage é uma superclasse para as classes de mensagem, como IMapMessage.

Partes da mensagem um XMS

Um XMS mensagem consiste em um cabeçalho, um conjunto de propriedades e um corpo.

Cabeçalho
O cabeçalho de uma mensagem contém campos, e todas as mensagens contêm o mesmo conjunto de campos de cabeçalho. XMS e aplicativos usam os valores dos campos de cabeçalho para identificar e rotear mensagens. Para obter mais informações sobre campos de cabeçalho, consulte "Campos de cabeçalho na mensagem um XMS" na página 73.

Conjunto de propriedades
As propriedades de uma mensagem especificam informações adicionais sobre a mensagem. Embora todas as mensagens tenham o mesmo conjunto de campos de cabeçalho, cada mensagem pode ter um conjunto diferente de propriedades. Para obter mais informações, consulte "Propriedades da mensagem um XMS" na página 74.

Conteúdo
O corpo de uma mensagem contém dados do aplicativo. Para obter mais informações, consulte "O corpo da mensagem um XMS" na página 77.

Um aplicativo pode selecionar quais mensagens ele deseja receber. Usando seletores de mensagens, que especificam os critérios de seleção. Os critérios podem ser baseados nos valores de determinados campos de cabeçalho e os valores de qualquer uma das propriedades de uma mensagem. Para obter mais informações sobre os seletores de mensagem, veja "Seletores de mensagens" na página 83.

Referências relacionadas

Campos de cabeçalho na mensagem um XMS

Para permitir que o aplicativo um XMS troque mensagens com um aplicativo WebSphere JMS , o cabeçalho da mensagem um XMS contém os campos do cabeçalho da mensagem JMS..

Propriedades da mensagem um XMS
O XMS suporta três tipos de propriedades de mensagem: propriedades definidas JMS, propriedades definidas IBM e propriedades definidas pelo aplicativo.

O corpo da mensagem um XMS
O corpo de uma mensagem contém dados do aplicativo. No entanto, uma mensagem não pode ter nenhum corpo e engloba apenas os campos de cabeçalho e propriedades.

Seletores de mensagens
O aplicativo Um XMS usa seletores de mensagens para selecionar as mensagens que deseja receber.

Mapeando XMS mensagens para IBM WebSphere MQ mensagens
Os campos e propriedades do cabeçalho JMS da mensagem um XMS são mapeados para campos nas estruturas do cabeçalho de uma mensagem IBM WebSphere MQ .

Campos de cabeçalho na mensagem um XMS

Para permitir que o aplicativo um XMS troque mensagens com um aplicativo WebSphere JMS , o cabeçalho da mensagem um XMS contém os campos do cabeçalho da mensagem JMS..

Os nomes desses campos de cabeçalho começam com o prefixo JMS.. Para obter uma descrição dos campos do cabeçalho da mensagem JMS, consulte *Java Message Service Specification, Versão 1.1*.

XMS implementa os campos de cabeçalho da mensagem JMS como atributos de um objeto de Mensagem. Cada campo de cabeçalho possui seus próprios métodos para configurar e obter seu valor. Para uma descrição desses métodos, consulte “IMessage” na página 127. Um campo de cabeçalho é sempre legível e gravável.

Tabela 21 na página 73 lista os campos de cabeçalho da mensagem JMS e indica como o valor de cada campo é configurado para uma mensagem transmitida. Alguns dos campos são configurados automaticamente por XMS quando um aplicativo envia uma mensagem ou, no caso de JMSRedelivered, quando um aplicativo recebe uma mensagem.

Tabela 21. Campos do cabeçalho da mensagem JMS

Nome do campo de cabeçalho da mensagem JMS	**Como o valor é configurado para uma mensagem transmitida (no formato *method* [*class*])**
JMSCorrelationID	Set JMSCorrelationID [Message]
JMSDeliveryMode	Send [MessageProducer]
JMSDestination	Send [MessageProducer]
JMSExpiration	Send [MessageProducer]
JMSMessageID	Send [MessageProducer]
JMSPriority	Send [MessageProducer]
JMSRedelivered	Receive [MessageConsumer]
JMSReplyTo	Set JMSReplyTo [Message]
JMSTimestamp	Send [MessageProducer]
JMSType	Set JMSType [Message]

Referências relacionadas

Partes da mensagem um XMS
Um XMS mensagem consiste em um cabeçalho, um conjunto de propriedades e um corpo.

Propriedades da mensagem um XMS

O XMS suporta três tipos de propriedades de mensagem: propriedades definidas JMS, propriedades definidas IBM e propriedades definidas pelo aplicativo.

O corpo da mensagem um XMS
O corpo de uma mensagem contém dados do aplicativo. No entanto, uma mensagem não pode ter nenhum corpo e engloba apenas os campos de cabeçalho e propriedades.

Seletores de mensagens
O aplicativo Um XMS usa seletores de mensagens para selecionar as mensagens que deseja receber.

Mapeando XMS mensagens para IBM WebSphere MQ mensagens
Os campos e propriedades do cabeçalho JMS da mensagem um XMS são mapeados para campos nas estruturas do cabeçalho de uma mensagem IBM WebSphere MQ .

Propriedades da mensagem um XMS

O XMS suporta três tipos de propriedades de mensagem: propriedades definidas JMS, propriedades definidas IBM e propriedades definidas pelo aplicativo.

Um aplicativo XMS pode trocar mensagens com um aplicativo WebSphere JMS porque XMS suporta as seguintes propriedades predefinidas de um objeto de Mensagem:

- As mesmas propriedades definidas pelo JMS que o WebSphere JMS suporta Os nomes dessas propriedades começam com o prefixo JMSX.
- As mesmas propriedades definidas pela IBMque o WebSphere JMS suporta Os nomes dessas propriedades começam com o prefixo JMS_IBM_.

Cada propriedade predefinida possui dois nomes:

- Um nome JMS para uma propriedade definida pelo JMS ou um nome WebSphere JMS para uma propriedade definida pelo IBM.

  Esse é o nome pelo qual a propriedade é conhecida no JMS ou no WebSphere JMSe também é o nome que é transmitido com uma mensagem que possui essa propriedade O aplicativo Um XMS usa esse nome para identificar a propriedade em uma expressão do seletor da mensagem
- Um XMS nome para identificar a propriedade em todas as situações, exceto em uma expressão de seletor de mensagens Cada nome XMS é definido como uma constante nomeada na classe `IBM.XMS.XMSC` . O valor da constante nomeada é o nome JMS ou WebSphere JMS correspondente.

Além das propriedades predefinidas, o aplicativo um XMS pode criar e usar seu próprio conjunto de propriedades de mensagem Essas propriedades são chamadas de *propriedades definidas pelo aplicativo.*

Depois que um aplicativo cria uma mensagem, as propriedades da mensagem são legíveis e graváveis. As propriedades permanecem legíveis e graváveis depois que o aplicativo envia a mensagem. Quando um aplicativo recebe uma mensagem, as propriedades da mensagem são somente leitura. Se um aplicativo chamar o método `Clear Properties` da classe de Mensagem quando as propriedades de uma mensagem forem somente leitura, as propriedades se tornarão legíveis e graváveis O método também limpa as propriedades.

A mensagem recebida, quando encaminhada após a limpeza das propriedades da mensagem, se comportará de uma maneira consistente com o comportamento de encaminhamento de um BytesMessage XMS for .NET do WMQ padrão com propriedades de mensagem limpas.

Isso, no entanto, não é recomendado desde que as seguintes propriedades sejam perdidas:

- Valor da propriedade JMS_IBM_Encoding, implicando que os dados da mensagem não podem ser decodificados de forma significativa.
- Valor da propriedade JMS_IBM_Format, implicando que o encadeamento de cabeçalho entre o cabeçalho da mensagem (MQMD ou o novo MQRFH2) e os cabeçalhos existentes seriam quebrados.

Referências relacionadas

Partes da mensagem um XMS
Um XMS mensagem consiste em um cabeçalho, um conjunto de propriedades e um corpo.

Campos de cabeçalho na mensagem um XMS
Para permitir que o aplicativo um XMS troque mensagens com um aplicativo WebSphere JMS , o cabeçalho da mensagem um XMS contém os campos do cabeçalho da mensagem JMS..

O corpo da mensagem um XMS
O corpo de uma mensagem contém dados do aplicativo. No entanto, uma mensagem não pode ter nenhum corpo e engloba apenas os campos de cabeçalho e propriedades.

Seletores de mensagens
O aplicativo Um XMS usa seletores de mensagens para selecionar as mensagens que deseja receber.

Mapeando XMS mensagens para IBM WebSphere MQ mensagens
Os campos e propriedades do cabeçalho JMS da mensagem um XMS são mapeados para campos nas estruturas do cabeçalho de uma mensagem IBM WebSphere MQ .

JMS-propriedades definidas de uma mensagem
Várias propriedades definidas pelo JMS de uma mensagem são suportadas por ambos XMS e WebSphere JMS

Tabela 22 na página 75 lista as propriedades definidas pelo JMS de uma mensagem suportada por XMS e WebSphere JMS. Para obter uma descrição das propriedades definidas pelo JMS, consulte *Java Message Service Specification, Versão 1.1* As propriedades definidas pelo JMS não são válidas para uma conexão em tempo real com um broker

A tabela especifica o tipo de dados de cada propriedade e indica como o valor da propriedade é configurado para uma mensagem transmitida. Algumas das propriedades são configuradas automaticamente por XMS quando um aplicativo envia uma mensagem ou, no caso de JMSXDeliveryCount, quando um aplicativo recebe uma mensagem.

Tabela 22. JMS-propriedades definidas de uma mensagem

Nome XMS da propriedade JMS definida	**JMS nome**	**Tipo de Dados**	**Como o valor é configurado para uma mensagem transmitida (no formato *method* [*class*])**
JMSX_APPID	JMSXAppID	System.String	Send [MessageProducer]
JMSX_DELIVERY_COUNT	JMSXDeliveryCount	System.Int32	Receive [MessageConsumer]
JMSX_GROUPID	JMSXGroupID	System.String	Configurar propriedade de sequência [PropertyContext]
JMSX_GROUPSEQ	JMSXGroupSeq	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMSX_USERID	JMSXUserID	System.String	Send [MessageProducer]

Propriedades Definidas pela IBM de uma Mensagem
Várias propriedades definidas pela IBMde uma mensagem são suportadas pelo XMS e pelo WebSphere JMS

Tabela 23 na página 76 lista as propriedades definidas IBM de uma mensagem que são suportadas por XMS e WebSphere JMS. Para obter mais informações sobre as propriedades definidas pela IBM, consulte *IBM WebSphere MQ Usando Java* ou a documentação do produto WebSphere Servidor de Aplicação

A tabela especifica o tipo de dados de cada propriedade e indica como o valor da propriedade é configurado para uma mensagem transmitida. Algumas das propriedades são configuradas automaticamente por XMS quando um aplicativo envia uma mensagem.

Tabela 23. Propriedades Definidas pela IBM de uma Mensagem

Nome XMS da propriedade IBM definida	**WebSphere JMS nome**	**Tipo de Dados**	**Como o valor é configurado para uma mensagem transmitida (no formato *method* [*class*])**
JMS_IBM_CHARACTER_SET	JMS_IBM_Character_Set	System.Int32	Configurar Propriedade Integer [PropertyContext]
CODIFICAÇÃO DE JMS_IBM_ENCODING	JMS_IBM_Encoding	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_EXCEPTIONMESSAGE	JMS_IBM_ExceptionMessage	System.String	Receive [MessageConsumer]
JMS_IBM_EXCEPTIONREASON	JMS_IBM_ExceptionReason	System.Int32	Receive [MessageConsumer]
JMS_IBM_EXCEPTIONTIMESTAMP	JMS_IBM_ExceptionTimestamp	System.Int64	Receive [MessageConsumer]
JMS_IBM_EXCEPTIONPROBLEMA DESTINO:	JMS_IBM_ExceptionProblemDestination	System.String	Receive [MessageConsumer]
JMS_IBM_FEEDBACK	JMS_IBM_Feedback	System.Int32	Configurar Propriedade Integer [PropertyContext]
FORMATO JMS_IBM_FORMAT	JMS_IBM_Format	System.String	Configurar propriedade de sequência [PropertyContext]
JMS_IBM_LAST_MSG_IN_GROUP	JMS_IBM_Last_Msg_In_Group	System.Boolean	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_MSGTYPE	JMS_IBM_MsgType	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_PUTAPPLTYPE	JMS_IBM_PutApplType	System.Int32	Send [MessageProducer]
JMS_IBM_PUTDATE	JMS_IBM_PutDate	System.String	Send [MessageProducer]
JMS_IBM_PUTTIME	JMS_IBM_PutTime	System.String	Send [MessageProducer]
JMS_IBM_REPORT_COA	JMS_IBM_Report_COA	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_REPORT_COD	JMS_IBM_Report_COD	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_REPORT_DISCARD_MSG	JMS_IBM_Report_Discard_Msg	System.Int32	Configurar Propriedade Integer [PropertyContext]

Tabela 23. Propriedades Definidas pela IBM de uma Mensagem (continuação)			
Nome XMS da propriedade IBM definida	**WebSphere JMS nome**	**Tipo de Dados**	**Como o valor é configurado para uma mensagem transmitida (no formato *method* [*class*])**
JMS_IBM_REPORT_EXCEPTION	JMS_IBM_Report_Exception	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_REPORT_EXPIRATION	JMS_IBM_Report_Expiration	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_REPORT_NAN	JMS_IBM_Report_NAN	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_REPORT_PAN	JMS_IBM_Report_PAN	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_REPORT_PASS_CORREL_ ID	JMS_IBM_Report_Pass_Correl_ID	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_REPORT_PASS_MSG_ID	JMS_IBM_Report_Pass_Msg_ID	System.Int32	Configurar Propriedade Integer [PropertyContext]
JMS_IBM_SYSTEM_MESSAGEID	JMS_IBM_System_MessageID	System.String	Send [MessageProducer]

Propriedades Definidas pelo Aplicativo de uma Mensagem

Um aplicativo XMS pode criar e usar seu próprio conjunto de propriedades de mensagem. Quando um aplicativo envia uma mensagem, essas propriedades também são transmitidas com a mensagem. Um aplicativo de recebimento, usando seletores de mensagens, pode, então, selecionar quais mensagens ele deseja receber com base nos valores dessas propriedades.

Para permitir que um aplicativo WebSphere JMS selecione e processe mensagens enviadas por um aplicativo XMS , o nome de uma propriedade definida pelo aplicativo deve estar em conformidade com as regras para formar identificadores em expressões do seletor de mensagens, conforme documentado em *IBM WebSphere MQ Usando Java.*. O valor de uma propriedade de aplicativo definido deve ter um dos seguintes tipos de dados: System.Boolean, System.SByte, System.Int16, System.Int32, System.Int64, System.Float, System.Double ou System.String.

O corpo da mensagem um XMS

O corpo de uma mensagem contém dados do aplicativo. No entanto, uma mensagem não pode ter nenhum corpo e engloba apenas os campos de cabeçalho e propriedades.

XMS suporta cinco tipos de corpo da mensagem:

bytes

O corpo contém um fluxo de bytes. Uma mensagem com esse tipo de corpo é chamada de mensagem de *bytes.* A interface IBytesMessage contém os métodos para processar o corpo de uma mensagem de bytes. Para obter mais informações, consulte “Mensagens de Bytes” na página 79.

Mapear
O corpo contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado. Uma mensagem com esse tipo de corpo é chamada de *mensagem de mapa.* A interface IMapMessage contém os métodos para processar o corpo de uma mensagem de mapa. Para obter mais informações, consulte “Mensagens de Mapa” na página 80.

Object
O corpo contém um objeto serializado Java ou .NET . Uma mensagem com esse tipo de corpo é chamada de mensagem de objeto . A interface IObjectMessage contém os métodos para processar o corpo de uma mensagem de objeto. Para obter mais informações, consulte “Mensagens de objeto” na página 81.

Fluxo
O corpo contém um fluxo de valores, em que cada valor possui um tipo de dados associado. Uma mensagem com esse tipo de corpo é chamada de *mensagem de fluxo.* A interface IStreamMessage contém os métodos para processar o corpo de uma mensagem de fluxo. Para obter mais informações, consulte “Mensagens de Fluxo” na página 82.

text
O corpo contém uma sequência. Uma mensagem com esse tipo de corpo é chamada de *mensagem de texto* . A interface ITextMessage contém os métodos para processar o corpo de uma mensagem de texto. Para obter mais informações, consulte “Mensagens de texto” na página 83.

A interface IMessage é o pai de todos os objetos de mensagem e pode ser usada em funções do sistema de mensagens para representar qualquer um dos tipos de mensagens do XMS.

Para obter informações sobre o tamanho e o máximo e o mínimo de valores de cada um desses tipos de dados, consulte Tabela 5 na página 42.

Para obter mais informações sobre os tipos de dados necessários para os elementos de dados do aplicativo gravados no corpo de uma mensagem e sobre os cinco tipos de mensagens do corpo, consulte os subtópicos

Referências relacionadas

Partes da mensagem um XMS
Um XMS mensagem consiste em um cabeçalho, um conjunto de propriedades e um corpo.

Campos de cabeçalho na mensagem um XMS
Para permitir que o aplicativo um XMS troque mensagens com um aplicativo WebSphere JMS , o cabeçalho da mensagem um XMS contém os campos do cabeçalho da mensagem JMS..

Propriedades da mensagem um XMS
O XMS suporta três tipos de propriedades de mensagem: propriedades definidas JMS, propriedades definidas IBM e propriedades definidas pelo aplicativo.

Seletores de mensagens
O aplicativo Um XMS usa seletores de mensagens para selecionar as mensagens que deseja receber.

Mapeando XMS mensagens para IBM WebSphere MQ mensagens
Os campos e propriedades do cabeçalho JMS da mensagem um XMS são mapeados para campos nas estruturas do cabeçalho de uma mensagem IBM WebSphere MQ .

Tipos de Dados para Elementos de Dados do Aplicativo

Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Por essa razão, cada elemento de dados do aplicativo gravado no corpo de uma mensagem pelo aplicativo um XMS deve ter um dos tipos de dados listados em Tabela 24 na página 79 Para cada tipo de dados, a tabela mostra o tipo de dados Java compatível. XMS fornece os métodos para gravar elementos de dados do aplicativo apenas com esses tipos de dados.

Tabela 24. Tipos de dados XMS que são compatíveis com tipos de dados Java		
XMS Tipo de dados	**Representa**	**Tipo de DadosJava Compatível**
System.Boolean	O valor booleano true ou false	booleano
System.Char16	Caractere de byte duplo	caractere
System.SByte	Inteiro de 8 bits sinalizado	byte
System.Int16	Número inteiro de 16 bits assinado	short
System.Int32	Número inteiro de 32 bits assinado	int
System.Int64	Número inteiro de 64 bits sinalizado	grande
System.Float	Número de vírgula flutuante assinado	float
System.Double	Número de vírgula flutuante de precisão dupla assinado	duplo
System.String	Sequência de caracteres	Sequência

Para obter informações sobre o tamanho, o valor máximo e o valor mínimo de cada um desses tipos de dados, consulte "Tipos primitivos XMS" na página 41.

Conceitos relacionados

Atributos e Propriedades de Objetos
Um objeto XMS pode ter atributos e propriedades, que são características do objeto, que são implementados de diferentes maneiras

Tipos primitivos XMS
O XMS fornece equivalentes dos oito tipos primitivos Java (byte, short, int, long, float, double, char e boolean). Isso permite a troca de mensagens entre XMS e JMS sem que os dados sejam perdidos ou corrompidos.

Conversão implícita de um valor de propriedade de um tipo de dados para outro
Quando um aplicativo obtém o valor de uma propriedade, o valor pode ser convertido por XMS em outro tipo de dados. Muitas regras controlam quais conversões são suportadas e como o XMS executa as conversões.

Referências relacionadas

Mensagens de Bytes
O corpo de uma mensagem de bytes contém um fluxo de bytes. O corpo contém apenas os dados reais, e é responsabilidade dos aplicativos de envio e recebimento interpretarem esses dados.

Mensagens de Mapa
O corpo de uma mensagem de mapa contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado.

Mensagens de objeto
O corpo de uma mensagem de objeto contém um objeto serializadoJava ou .NET..

Mensagens de Fluxo
O corpo de uma mensagem de fluxo contém um fluxo de valores, em que cada valor possui um tipo de dados associado.

Mensagens de texto
O corpo de uma mensagem de texto contém uma sequência.

Mensagens de Bytes

O corpo de uma mensagem de bytes contém um fluxo de bytes. O corpo contém apenas os dados reais, e é responsabilidade dos aplicativos de envio e recebimento interpretarem esses dados.

Mensagens de bytes são úteis se o aplicativo um XMS precisar trocar mensagens com aplicativos que não estão usando a interface de programação de aplicativos XMS ou JMS.

Depois que um aplicativo cria uma mensagem de byte, o corpo da mensagem é somente gravação. O aplicativo monta os dados do aplicativo para o corpo, chamando os métodos de gravação apropriados da interface IBytesMessage para .NET. Toda vez que o aplicativo grava um valor para o fluxo de mensagens de bytes, o valor é montado imediatamente após o valor anterior gravado pelo aplicativo. XMS mantém um cursor interno para lembrar a posição do último byte que foi montado.

Quando o aplicativo envia a mensagem, o corpo da mensagem se torna somente leitura. Nesse modo, o aplicativo pode enviar a mensagem repetidamente.

Quando um aplicativo recebe uma mensagem de bytes, o corpo da mensagem é de leitura. O aplicativo pode usar os métodos de leitura apropriados da interface IBytesMessage para ler o conteúdo do fluxo de mensagens de bytes. O aplicativo lê os bytes na sequência e XMS mantém um cursor interno para lembrar a posição do último byte que foi lido.

Se um aplicativo chamar o método Reconfigurar da interface IBytesMessage quando o corpo de uma mensagem de bytes for gravável, o corpo se tornará somente leitura. O método também reposiciona o cursor no início do fluxo de mensagens de bytes.

Se um aplicativo chamar o método `Clear Body` da interface IMessage para o .NET quando o corpo de uma mensagem de bytes for somente leitura, o corpo se tornará gravável. O método também limpa o corpo.

Referências relacionadas

Tipos de Dados para Elementos de Dados do Aplicativo
Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Mensagens de Mapa
O corpo de uma mensagem de mapa contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado.

Mensagens de objeto
O corpo de uma mensagem de objeto contém um objeto serializadoJava ou .NET..

Mensagens de Fluxo
O corpo de uma mensagem de fluxo contém um fluxo de valores, em que cada valor possui um tipo de dados associado.

Mensagens de texto
O corpo de uma mensagem de texto contém uma sequência.

IBytesMessage (para a interface .NET)
Uma mensagem de bytes é uma mensagem cujo corpo compreende um fluxo de bytes.

Mensagens de Mapa

O corpo de uma mensagem de mapa contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado.

Em cada par de nome-valor, o nome é uma sequência que identifica o valor e o valor é um elemento de dados do aplicativo que possui um dos tipos de dados XMS listados em Tabela 24 na página 79. A ordem dos pares nome-valor não está definida. A classe MapMessage contém os métodos para configurar e obter pares nome-valor.

Um aplicativo pode acessar um par nome-valor aleatoriamente, especificando seu nome.

Um aplicativo .NET pode usar a propriedade MapNames para obter uma enumeração dos nomes no corpo da mensagem de mapa.

Quando um aplicativo obtém o valor de um par nome-valor, o valor pode ser convertido por XMS em outro tipo de dados. Por exemplo, para obter um número inteiro a partir do corpo de uma mensagem de mapa, um aplicativo pode chamar o método GetString da classe MapMessage, que retorna o número inteiro

como uma sequência. As conversões suportadas são as mesmas que aquelas suportadas quando XMS converte um valor de propriedade de um tipo de dados para outro. Para obter mais informações sobre as conversões suportadas, consulte “Conversão implícita de um valor de propriedade de um tipo de dados para outro” na página 42.

Depois que um aplicativo cria uma mensagem de mapa, o corpo da mensagem é legível e gravável. O corpo permanece legível e gravável depois que o aplicativo envia a mensagem. Quando um aplicativo recebe uma mensagem de mapa, o corpo da mensagem é de leitura. Se um aplicativo chamar o método Clear Body da classe de mensagem quando o corpo de uma mensagem de mapa for de leitura, o corpo se tornará legível e gravável. O método também limpa o corpo.

Conceitos relacionados

Conversão implícita de um valor de propriedade de um tipo de dados para outro
Quando um aplicativo obtém o valor de uma propriedade, o valor pode ser convertido por XMS em outro tipo de dados. Muitas regras controlam quais conversões são suportadas e como o XMS executa as conversões.

Referências relacionadas

Tipos de Dados para Elementos de Dados do Aplicativo
Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Mensagens de Bytes
O corpo de uma mensagem de bytes contém um fluxo de bytes. O corpo contém apenas os dados reais, e é responsabilidade dos aplicativos de envio e recebimento interpretarem esses dados.

Mensagens de objeto
O corpo de uma mensagem de objeto contém um objeto serializadoJava ou .NET..

Mensagens de Fluxo
O corpo de uma mensagem de fluxo contém um fluxo de valores, em que cada valor possui um tipo de dados associado.

Mensagens de texto
O corpo de uma mensagem de texto contém uma sequência.

IMapMessage (para a interface .NET)
Uma mensagem de mapa é uma mensagem cujo corpo consiste em um conjunto de pares nome-valor, em que cada valor possui um tipo de dados associado.

Mensagens de objeto

O corpo de uma mensagem de objeto contém um objeto serializadoJava ou .NET..

O aplicativo Um XMS pode receber uma mensagem de objeto, mudar seus campos de cabeçalho e propriedades e, em seguida, enviá-la para outro destino Um aplicativo também pode copiar o corpo de uma mensagem de objeto e usá-lo para formar outra mensagem de objeto. XMS trata o corpo de uma mensagem de objeto como uma matriz de bytes.

Depois que um aplicativo cria uma mensagem de objeto, o corpo da mensagem é legível e gravável. O corpo permanece legível e gravável depois que o aplicativo envia a mensagem. Quando um aplicativo recebe uma mensagem de objeto, o corpo da mensagem é de leitura. Se um aplicativo chamar o método `Clear Body` da interface do IMessage para .NET quando o corpo de uma mensagem de objeto for somente leitura, o corpo se tornará legível e gravável O método também limpa o corpo.

Referências relacionadas

Tipos de Dados para Elementos de Dados do Aplicativo
Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Mensagens de Bytes
O corpo de uma mensagem de bytes contém um fluxo de bytes. O corpo contém apenas os dados reais, e é responsabilidade dos aplicativos de envio e recebimento interpretarem esses dados.

Mensagens de Mapa
O corpo de uma mensagem de mapa contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado.

Mensagens de Fluxo
O corpo de uma mensagem de fluxo contém um fluxo de valores, em que cada valor possui um tipo de dados associado.

Mensagens de texto
O corpo de uma mensagem de texto contém uma sequência.

IObjectMessage (para a interface .NET)
Uma mensagem de objeto é uma mensagem cujo corpo compreende um objeto Java ou .NET serializado.

Mensagens de Fluxo

O corpo de uma mensagem de fluxo contém um fluxo de valores, em que cada valor possui um tipo de dados associado.

O tipo de dados de um valor é um dos tipos de dados XMS listados em Tabela 24 na página 79.

Depois que um aplicativo cria uma mensagem de fluxo, o corpo da mensagem é gravável. O aplicativo monta os dados do aplicativo para o corpo, chamando os métodos de gravação apropriados da interface IStreamMessage para .NET. Toda vez que o aplicativo grava um valor para o fluxo de mensagens, o valor e seu tipo de dados são montados imediatamente após o valor anterior gravado pelo aplicativo. XMS mantém um cursor interno para lembrar a posição do último valor que foi montado.

Quando o aplicativo envia a mensagem, o corpo da mensagem se torna somente leitura. Nesse modo, o aplicativo pode enviar a mensagem várias vezes.

Quando um aplicativo recebe uma mensagem de fluxo, o corpo da mensagem é de leitura. O aplicativo pode usar os métodos de leitura apropriados da interface IStreamMessage para .NET para ler o conteúdo do fluxo de mensagens. O aplicativo lê os valores na sequência, e XMS mantém um cursor interno para lembrar a posição do último valor que foi lido.

Quando um aplicativo lê um valor do fluxo de mensagens, o valor pode ser convertido pelo XMS em outro tipo de dados. Por exemplo, para ler um número inteiro a partir do fluxo de mensagens, um aplicativo pode chamar o método ReadString, que retorna o número inteiro como uma sequência. As conversões suportadas são as mesmas que aquelas suportadas quando XMS converte um valor de propriedade de um tipo de dados para outro. Para obter mais informações sobre as conversões suportadas, consulte “Conversão implícita de um valor de propriedade de um tipo de dados para outro” na página 42.

Se um erro ocorrer enquanto um aplicativo está tentando ler um valor do fluxo de mensagens, o cursor não está avançado. O aplicativo pode se recuperar do erro ao tentar ler o valor como outro tipo de dados.

Se um aplicativo chamar o método Reconfigurar da interface IStreamMessage para .NET quando o corpo de uma mensagem de fluxo for de gravação somente, o corpo se tornará somente leitura. O método também reposiciona o cursor no início do fluxo de mensagens.

Se um aplicativo chamar o método `Clear Body` da interface do IMessage para .NET quando o corpo de uma mensagem de fluxo for somente leitura, o corpo se tornará somente gravação O método também limpa o corpo.

Conceitos relacionados

Conversão implícita de um valor de propriedade de um tipo de dados para outro
Quando um aplicativo obtém o valor de uma propriedade, o valor pode ser convertido por XMS em outro tipo de dados. Muitas regras controlam quais conversões são suportadas e como o XMS executa as conversões.

Referências relacionadas

Tipos de Dados para Elementos de Dados do Aplicativo
Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Mensagens de Bytes

O corpo de uma mensagem de bytes contém um fluxo de bytes. O corpo contém apenas os dados reais, e é responsabilidade dos aplicativos de envio e recebimento interpretarem esses dados.

Mensagens de Mapa
O corpo de uma mensagem de mapa contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado.

Mensagens de objeto
O corpo de uma mensagem de objeto contém um objeto serializadoJava ou .NET..

Mensagens de texto
O corpo de uma mensagem de texto contém uma sequência.

IStreamMessage (para a interface .NET)
Uma mensagem de fluxo é uma mensagem cujo corpo compreende um fluxo de valores, em que cada valor possui um tipo de dados associado. Os conteúdos do corpo são gravados e lidos sequencialmente.

Mensagens de texto

O corpo de uma mensagem de texto contém uma sequência.

Depois que um aplicativo cria uma mensagem de texto, o corpo da mensagem é legível e gravável. O corpo permanece legível e gravável depois que o aplicativo envia a mensagem. Quando um aplicativo recebe uma mensagem de texto, o corpo da mensagem é somente leitura. Se um aplicativo chamar o método Clear Body da interface IMessage para .NET quando o corpo de uma mensagem de texto for de leitura, o corpo se tornará legível e gravável. O método também limpa o corpo.

Referências relacionadas

Tipos de Dados para Elementos de Dados do Aplicativo
Para assegurar que um aplicativo XMS possa trocar mensagens com um aplicativo IBM WebSphere MQ classes para JMS , ambos os aplicativos devem ser capazes de interpretar os dados do aplicativo no corpo de uma mensagem da mesma maneira.

Mensagens de Bytes
O corpo de uma mensagem de bytes contém um fluxo de bytes. O corpo contém apenas os dados reais, e é responsabilidade dos aplicativos de envio e recebimento interpretarem esses dados.

Mensagens de Mapa
O corpo de uma mensagem de mapa contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado.

Mensagens de objeto
O corpo de uma mensagem de objeto contém um objeto serializadoJava ou .NET..

Mensagens de Fluxo
O corpo de uma mensagem de fluxo contém um fluxo de valores, em que cada valor possui um tipo de dados associado.

ITextMessage (para a interface .NET)
Uma mensagem de texto é uma mensagem cujo corpo compreende uma sequência.

Seletores de mensagens

O aplicativo Um XMS usa seletores de mensagens para selecionar as mensagens que deseja receber.

Quando um aplicativo cria um consumidor de mensagens, ele pode associar uma expressão de seletor de mensagem ao consumidor. A expressão do seletor de mensagem especifica os critérios de seleção.

Quando um aplicativo está se conectando ao gerenciador de fila do IBM WebSphere MQ V7.0 , a seleção de mensagem é feita no lado do gerenciador de filas XMS não faz nenhuma seleção e simplesmente entrega a mensagem que recebeu do gerenciador de filas, fornecendo, assim, melhor desempenho.

No entanto, quando um aplicativo está se conectando ao IBM WebSphere MQ V6.0 e abaixo, WebSphere Event Brokerou WebSphere Message Broker, WebSphere Barramento de Integração do Integration Bus XMS determina se cada mensagem recebida satisfaz os critérios de seleção. Se uma mensagem atender aos critérios de seleção, XMS entregará a mensagem para o consumidor de mensagens. Se uma

mensagem não atender aos critérios de seleção, XMS não entregará a mensagem e, no domínio ponto a ponto, a mensagem permanecerá na fila.

Um aplicativo pode criar mais de um consumidor de mensagens, cada um com sua própria expressão de seletor de mensagem. Se uma mensagem recebida atender aos critérios de seleção de mais de um consumidor de mensagens, o XMS entregará a mensagem para cada um desses consumidores.

Uma expressão de seletor de mensagem pode referenciar as seguintes propriedades de uma mensagem:

- Propriedades Definidas pelo JMS
- Propriedades Definidas pela IBM
- Propriedades Definidas pelo Aplicativo

Ele também pode referenciar os campos de cabeçalho da mensagem a seguir:

- JMSCorrelationID
- JMSDeliveryMode
- JMSMessageID
- JMSPriority
- JMSTimestamp
- JMSType

Uma expressão de seletor de mensagem, no entanto, não pode referenciar dados no corpo de uma mensagem.

Aqui está um exemplo de uma expressão de seletor de mensagem:

```
JMSPriority > 3 AND manufacturer = 'Jaguar' AND model in ('xj6','xj12')
```

O XMS entrega uma mensagem para um consumidor de mensagens com essa expressão de seletor de mensagem apenas se a mensagem tiver uma prioridade maior que 3; uma propriedade definida pelo aplicativo, fabricante, com um valor de `Jaguar;` e outra propriedade definida pelo aplicativo, modelo, com um valor de `xj6` ou `xj12.`

As regras de sintaxe para formar uma expressão de seletor de mensagem em XMS são as mesmas que aquelas em IBM WebSphere MQ classes para JMS. Para obter informações sobre como construir uma expressão do seletor de mensagens, consulte *WebSphere MQ Usando Java.* . Observe que, em uma expressão de seletor de mensagem, os nomes das propriedades definidas pelo JMS devem ser os nomes JMS e os nomes das propriedades definidas pela IBMdevem ser os nomes IBM WebSphere MQ classes para JMS . Não é possível usar os nomes XMS em uma expressão de seletor de mensagem.

Referências relacionadas

Partes da mensagem um XMS
Um XMS mensagem consiste em um cabeçalho, um conjunto de propriedades e um corpo.

Campos de cabeçalho na mensagem um XMS
Para permitir que o aplicativo um XMS troque mensagens com um aplicativo WebSphere JMS , o cabeçalho da mensagem um XMS contém os campos do cabeçalho da mensagem JMS..

Propriedades da mensagem um XMS
O XMS suporta três tipos de propriedades de mensagem: propriedades definidas JMS, propriedades definidas IBM e propriedades definidas pelo aplicativo.

O corpo da mensagem um XMS
O corpo de uma mensagem contém dados do aplicativo. No entanto, uma mensagem não pode ter nenhum corpo e engloba apenas os campos de cabeçalho e propriedades.

Mapeando XMS mensagens para IBM WebSphere MQ mensagens
Os campos e propriedades do cabeçalho JMS da mensagem um XMS são mapeados para campos nas estruturas do cabeçalho de uma mensagem IBM WebSphere MQ .

Mapeando XMS mensagens para IBM WebSphere MQ mensagens

Os campos e propriedades do cabeçalho JMS da mensagem um XMS são mapeados para campos nas estruturas do cabeçalho de uma mensagem IBM WebSphere MQ .

Quando o aplicativo um XMS é conectado a um gerenciador de fila do IBM WebSphere MQ , as mensagens enviadas ao gerenciador de filas são mapeados para mensagens do IBM WebSphere MQ da mesma maneira que as mensagens do IBM WebSphere MQ classes para JMS são mapeadas para mensagens do IBM WebSphere MQ em circunstâncias similares.

Se a propriedade XMSC_WMQ_TARGET_CLIENT de um objeto Destination estiver configurada como XMSC_WMQ_TARGET_DEST_JMS, os campos de cabeçalho JMS e as propriedades de uma mensagem enviada para o destino serão mapeados para os campos nas estruturas de cabeçalho MQMD e MQRFH2 da mensagem IBM WebSphere MQ . Configurar a propriedade XMSC_WMQ_TARGET_CLIENT dessa maneira assume que o aplicativo que recebe a mensagem pode manipular um cabeçalho MQRFH2. O aplicativo de recebimento pode, portanto, ser outro aplicativo XMS , um aplicativo IBM WebSphere MQ classes para JMS ou um aplicativo IBM WebSphere MQ nativo que foi projetado para manipular um cabeçalho MQRFH2 .

Se a propriedade XMSC_WMQ_TARGET_CLIENT de um objeto de Destino for configurada como XMSC_WMQ_TARGET_DEST_MQ em vez disso, os campos de cabeçalho JMS e as propriedades de uma mensagem enviada para o destino serão mapeados para os campos na estrutura de cabeçalho MQMD da mensagem IBM WebSphere MQ . A mensagem não contém um cabeçalho MQRFH2 e quaisquer campos de cabeçalho JMS e propriedades que não podem ser mapeados para campos na estrutura de cabeçalho MQMD são ignorados. O aplicativo que recebe a mensagem pode, portanto, ser um IBM WebSphere MQ nativo que não foi projetado para manipular um cabeçalho MQRFH2.

As mensagens do IBM WebSphere MQ recebidas de um gerenciador de filas são mapeadas para mensagens do XMS da mesma maneira que mensagens do IBM WebSphere MQ são mapeadas para mensagens do IBM WebSphere MQ classes para JMS em circunstâncias semelhantes.

Se uma mensagem IBM WebSphere MQ recebida tiver um cabeçalho MQRFH2, a mensagem XMS resultante terá um corpo cujo tipo seja determinado pelo valor da propriedade **Msd** contida na pasta `mcd` do cabeçalho MQRFH2. Se a propriedade **Msd** não estiver presente no cabeçalho MQRFH2, ou se a mensagem IBM WebSphere MQ não tiver cabeçalho MQRFH2, a mensagem XMS resultante terá um corpo cujo tipo seja determinado pelo valor do campo *`Format`* no cabeçalho MQMD. Se o campo *`Format`* estiver configurado como MQFMT_STRING, a mensagem XMS será uma mensagem de texto. Caso contrário, a mensagem XMS é uma mensagem de bytes. Se a mensagem IBM WebSphere MQ não tiver cabeçalho MQRFH2, apenas os campos de cabeçalho JMS e propriedades que podem ser derivados de campos no cabeçalho MQMD são configurados.

Para obter mais informações sobre o mapeamento de IBM WebSphere MQ classes para JMS mensagens para IBM WebSphere MQ mensagens, consulte *IBM WebSphere MQ Usando Java*.

Referências relacionadas

Partes da mensagem um XMS
Um XMS mensagem consiste em um cabeçalho, um conjunto de propriedades e um corpo.

Campos de cabeçalho na mensagem um XMS
Para permitir que o aplicativo um XMS troque mensagens com um aplicativo WebSphere JMS , o cabeçalho da mensagem um XMS contém os campos do cabeçalho da mensagem JMS..

Propriedades da mensagem um XMS
O XMS suporta três tipos de propriedades de mensagem: propriedades definidas JMS, propriedades definidas IBM e propriedades definidas pelo aplicativo.

O corpo da mensagem um XMS
O corpo de uma mensagem contém dados do aplicativo. No entanto, uma mensagem não pode ter nenhum corpo e engloba apenas os campos de cabeçalho e propriedades.

Seletores de mensagens
O aplicativo Um XMS usa seletores de mensagens para selecionar as mensagens que deseja receber.

Lendo e Gravando o Descritor de Mensagens a partir de um Aplicativo Message Service Client for .NET

É possível acessar todos os campos do descritor de mensagens (MQMD) de uma mensagem IBM WebSphere MQ , exceto StrucId e Versão; BackoutCount pode ser lido, mas não gravado. Esse recurso está disponível apenas ao conectar-se a um IBM WebSphere MQ gerenciador de filas Versão 6 e acima e é controlado pelas propriedades de destino descritas posteriormente

Os atributos da mensagem fornecidos pelo Message Service Client for .NET facilitam os aplicativos XMS para configurar campos MQMD e também para a unidade de aplicativos IBM WebSphere MQ .

Algumas restrições se aplicam ao usar o sistema de mensagensPublicação/Assinatura . Por exemplo, os campos MQMD como MsgID e CorrelId, se configurados, são ignorados.

A função descrita neste tópico está indisponível para o sistema de mensagens do Publicação/Assinatura quando você está se conectando a um gerenciador de filas do IBM WebSphere MQ V6 Ele também estará indisponível quando a propriedade **PROVIDERVERSION** for configurada como 6.

AcessandoIBM WebSphere MQ Dados da mensagem a partir de um aplicativo Message Service Client for .NET

Você pode acessar os dados da mensagem completos IBM WebSphere MQ, incluindo o cabeçalho MQRFH2 (se presente) e quaisquer outros cabeçalhos IBM WebSphere MQ (se presentes) dentro de um aplicativo Message Service Client for .NET como o corpo de uma JMSBytesMessage.

A função descrita neste tópico está disponível apenas ao se conectar a um gerenciador de filas do IBM WebSphere MQ na Versão 7 ou posterior e o provedor de sistemas de mensagens do IBM WebSphere MQ está no modo normal.

As propriedades do objeto de destino determinam como o aplicativo XMS acessa toda uma mensagem do IBM WebSphere MQ(incluindo o cabeçalho MQRFH2 , se presente) como o corpo de um JMSBytesMessage.

Resolução de Problemas

Este seçãocapítulo fornece informações para ajudar a detectar e lidar com problemas ao usar o Message Service Client for .NET.

Este seçãocapítulo fornece informações para ajudá-lo com a determinação de problema para aplicativos XMS e descreve como configurar o First Failure Data Capture (FFDC) e o rastreamento para aplicativos .NET

Esse seçãocapítulo contém o seguinte tópicosseções:

- "Configuração de rastreio para aplicativos .NET" na página 86
- "Configuração do FFDC para aplicativos .NET" na página 91
- "Dicas para Resolução de Problemas" na página 91

Configuração de rastreio para aplicativos .NET

Para aplicativos XMS .NET, é possível configurar o rastreio por meio de um arquivo de configuração do aplicativo, bem como por meio das variáveis de ambiente do XMS. É possível selecionar os componentes que você deseja rastrear. O rastreio é normalmente usado sob a orientação do Suporte IBM

O rastreio para o XMS .NET baseia-se na infraestrutura de rastreio padrão do .NET.

Todo o rastreio, exceto o rastreio de erro, fica desativado por padrão. É possível ativar o rastreio e definir as configurações de rastreio de uma das maneiras a seguir:

- Usando um arquivo de configuração de aplicativo com um nome que consiste no nome do programa executável ao qual o arquivo se relaciona, com o sufixo `.config`. Por exemplo, o arquivo de configuração de aplicativo para text.exe teria o nome text.exe.config. O uso de um arquivo de configuração de aplicativo é a maneira preferida de ativar o rastreio para aplicativos XMS.NET. Para

obter detalhes adicionais, consulte “Configuração de rastreio usando um arquivo de configuração de aplicativo” na página 87.

- Usando as variáveis de ambiente XMS como para aplicativos XMS C ou C++. Para obter detalhes adicionais, consulte “Configuração de rastreio usando variáveis de ambiente XMS” na página 89.

O arquivo de rastreio ativo tem um nome do formato xms_trace <*PID*> .log, em que <*PID*> representa o ID do processo do aplicativo. O tamanho do arquivo de rastreio ativo está, por padrão, limitado a 20 MB. Quando esse limite é atingido, o arquivo é renomeado e arquivado. Os arquivos arquivados têm nomes do formato xms_trace <*PID*> *_YY.MM.DD_HH.MM.SS*.log

Por padrão, o número de arquivos de rastreio retidos é quatro, isto é, um arquivo ativo e três arquivos arquivados. Esses quatro arquivos são usados como um buffer de rolagem até que o aplicativo pare, com o arquivo mais antigo sendo removido e substituído pelo arquivo mais recente. É possível mudar o número de arquivos de rastreio especificando um número diferente no arquivo de configuração de aplicativo. No entanto, deve haver pelo menos dois arquivos (um arquivo ativo e um arquivo arquivado).

Há dois formatos de arquivo de rastreio disponíveis:

- Os arquivos de rastreio de formato básico são legíveis por humanos em um formato WebSphere Servidor de Aplicação. Esse é o formato de arquivo de rastreio padrão. O formato básico não é compatível com as ferramentas do analisador de rastreio.
- Os arquivos de rastreio no formato avançado são compatíveis com as ferramentas do analisador de rastreio. Deve-se especificar o desejo de produzir arquivos de rastreio no formato avançado no arquivo de configuração de aplicativo.

As entradas de rastreio contêm as informações a seguir:

- A data e hora em que o rastreio foi registrado
- O nome de classe
- O tipo de rastreio
- A mensagem de rastreio

O exemplo a seguir mostra uma extração de algum rastreio:

```
[09/11/2005 14:33:46:914276]    00000004    IBM.XMS.Comms.IoRequest    >    Allocate    Entry
[09/11/2005 14:33:46:914276]    00000004    IBM.XMS.Comms.IoRequest    >    Initialize    Entry
[09/11/2005 14:33:46:914276]    00000004    IBM.XMS.Comms.IoRequest    <    Initialize    Exit
[09/11/2005 14:33:46:914276]    00000004    IBM.XMS.Comms.IoRequest    <    Allocate    Exit
```

No exemplo anterior, o formato é:

```
[Date Time:Microsecs]         Thread-id    Classname        Trace-type    Methodname    Entry
or Exit
```

em que `Trace-type` é:

> para entrada
< para saída
d para informações sobre depuração

Configuração de rastreio usando um arquivo de configuração de aplicativo

A maneira preferida de configurar o rastreio para aplicativos XMS .NET é com um arquivo de configuração de aplicativo. A seção de rastreio desse arquivo inclui parâmetros que definem o que deve ser rastreado, o local do arquivo de rastreio e o tamanho máximo permitido, o número de arquivos de rastreio usados e o formato do arquivo de rastreio.

Para ativar o rastreio usando o arquivo de configuração de aplicativo, simplesmente coloque o arquivo no mesmo diretório que o arquivo executável para seu aplicativo.

O rastreio pode ser ativado por componente e tipo de rastreio. Também é possível ativar o rastreio para um grupo inteiro de rastreio. É possível ativar o rastreio para componentes em uma hierarquia, individual ou coletivamente. Os tipos de rastreio disponíveis incluem:

- Rastreamento de Depuração
- Rastreio da Exceção
- Avisos, mensagens informativas e mensagens de erro
- Rastreio de entrada e saída do método

O exemplo a seguir mostra as configurações de rastreio definidas na seção Rastreio de um arquivo de configuração de aplicativo:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <configSections>
        <sectionGroup name="IBM.XMS">
            <section name="Trace"
                     type="System.Configuration.SingleTagSectionHandler" />
        </sectionGroup>
    </configSections>

    <IBM.XMS>
        <Trace traceSpecification="*=all=enabled" traceFilePath=""
               traceFileSize="20000000" traceFileNumber="3"
               traceFormat="advanced" />
    </IBM.XMS>
 </configuration>
```

Tabela 25 na página 88 descreve as configurações de parâmetros em mais detalhes.

Tabela 25. Configurações de parâmetros de rastreio do arquivo de configuração de aplicativo

Parâmetro	**Descrição**
`traceSpecification=<ComponentName>=<type>=<state>`	`<ComponentName>` é o nome da classe que deseja rastrear. É possível usar um caractere curinga * neste nome. Por exemplo, `*=all=enabled` especifica que você deseja rastrear todas as classes e `IBM.XMS.impl.*=all=enabled` especifica que é necessário apenas o rastreio de API. `<type>` pode ser qualquer um dos seguintes tipos de rastreio: • all • debug • evento • EntryExit O `<state>` pode ser ativado ou desativado É possível sequenciar vários elementos de rastreio juntos usando um delimitador ':' (dois pontos).
`traceFilePath="<filename>"`	Se você não especificar um traceFilePath ou se o traceFilePath estiver presente, mas contiver uma sequência vazia, o arquivo de rastreio será colocado no diretório atual. Para armazenar o arquivo de rastreio em um diretório nomeado, especifique o nome de diretório no traceFilePath, por exemplo: `traceFilePath="c:\somepath"`

Tabela 25. Configurações de parâmetros de rastreio do arquivo de configuração de aplicativo (continuação)	
Parâmetro	**Descrição**
`traceFileSize="<size>"`	O tamanho máximo permitido do arquivo de rastreio. Quando um arquivo atinge este tamanho, ele é arquivado e renomeado. O máximo padrão é 20 KB, especificado como `traceFileSize="20000000"`.
`traceFileNumber="<number>"`	O número de arquivos de rastreio que devem ser retidos. O padrão é 4 (um arquivo ativo e 3 archives). O número mínimo permitido é 2.
`traceFormat="<format>"`	O formato de rastreio padrão é básico. Os arquivos de rastreio serão produzidos nesse formato se você especificar `traceFormat="basic"`ou se não especificar um traceFormatou se o traceFormat estiver presente, mas contiver uma sequência vazia. Se você precisar de rastreio que seja compatível com as ferramentas do analisador de rastreio, deverá especificar `traceFormat="advanced"`.

As configurações de rastreio no arquivo de configuração de aplicativo são dinâmicas e lidas novamente sempre que o arquivo é salvo ou substituído. Se forem localizados erros no arquivo após sua edição, as configurações do arquivo de rastreio serão revertidas para seus valores padrão.

Conceitos relacionados

Configuração de rastreio usando variáveis de ambiente XMS
Como alternativa ao uso de um arquivo de configuração do aplicativo, é possível ativar o rastreio usando variáveis de ambiente do XMS. Essas variáveis de ambiente serão usadas apenas se não houver especificação de rastreio no arquivo de configuração do aplicativo.

Configuração de rastreio usando variáveis de ambiente XMS

Como alternativa ao uso de um arquivo de configuração do aplicativo, é possível ativar o rastreio usando variáveis de ambiente do XMS. Essas variáveis de ambiente serão usadas apenas se não houver especificação de rastreio no arquivo de configuração do aplicativo.

Para configurar o rastreio de um aplicativo XMS .NET, configure as variáveis de ambiente a seguir, antes de executar o aplicativo:

Tabela 26. Configurações da variável de ambiente para o rastreio .NET			
Variáveis de ambiente	**Padrão**	**Configurações**	**Significado**
XMS_TRACE_ON	Não-aplicável	Não aplicável: o valor dessa variável é ignorado	Se XMS_TRACE_ON estiver configurado, todo o rastreio será ativado por padrão.

Tabela 26. Configurações da variável de ambiente para o rastreio .NET (continuação)			
Variáveis de ambiente	**Padrão**	**Configurações**	**Significado**
XMS_TRACE_FILE_PATH	Diretório de trabalho atual	/dirpath/	O caminho do diretório no qual os registros de rastreio e de FFDC são gravados. O XMS cria arquivos FFDC e de rastreio no diretório ativo atual, a menos que você especifique um local alternativo. É possível especificar um local alternativo configurando a variável de ambiente XMS_TRACE_FILE_PATH para o nome do caminho completo do diretório em que você deseja que o XMS crie os arquivos FFDC e de rastreio. Deve-se configurar essa variável de ambiente antes de iniciar o aplicativo que você deseja rastrear. Deve-se certificar de que o identificador de usuário sob o qual o aplicativo é executado tenha a autoridade para gravar no diretório em que o XMS cria os arquivos FFDC e de rastreio.
XMS_TRACE_FORMAT	BÁSICA	BASIC, ADVANCED	Especifica o formato de rastreio necessário, que pode ser BASIC ou ADVANCED. O formato padrão é BASIC. O formato ADVANCED é compatível com as ferramentas do analisador de rastreio.
XMS_TRACE_SPECIFICATION	Não-aplicável	Consulte “Configuração de rastreio usando um arquivo de configuração de aplicativo” na página 87	Substitui a especificação de rastreio, que segue o formato especificado em “Configuração de rastreio usando um arquivo de configuração de aplicativo” na página 87.

Conceitos relacionados

Configuração de rastreio usando um arquivo de configuração de aplicativo

A maneira preferida de configurar o rastreio para aplicativos XMS .NET é com um arquivo de configuração de aplicativo. A seção de rastreio desse arquivo inclui parâmetros que definem o que deve ser rastreado,

o local do arquivo de rastreio e o tamanho máximo permitido, o número de arquivos de rastreio usados e o formato do arquivo de rastreio.

Configuração do FFDC para aplicativos .NET

Para a implementação .NET do XMS, um arquivo FFDC é produzido para cada FFDC.

Os arquivos de Primeira captura de dados com falha (FFDC) são armazenados em arquivos de texto legíveis por humanos. Esses arquivos possuem nomes no formato `xmsffdc<`*`processID`*`>_<`*`Date`*`>T<`*`Timestamp`*`>.txt`. Um exemplo de um nome de arquivo é `xmsffdc264_2006.01.06T13.18.52.990955.txt` O registro de data e hora contém a resolução de microssegundos.

Os arquivos começam com a data e hora em que a exceção ocorreu, seguido pelo tipo de exceção. Os arquivos incluem um probeId curto exclusivo, que pode ser usado para localizar onde esse FFDC ocorreu.

Não é necessário realizar nenhuma configuração para ativar o FFDC. Por padrão, todos os arquivos FFDC são gravados no diretório atual. No entanto, se necessário, é possível especificar um diretório diferente mudando ffdcDirectory na seção Rastreio do arquivo de configuração de aplicativo. No exemplo a seguir, todos os arquivos de rastreio são registrados no diretório c:client\ffdc:

```
<IBM.XMS>
       <Trace ffdc=true ffdcDirectory="c:\client\ffdc"/>
</IBM.XMS>
```

É possível desativar o rastreio configurando o FFDC como false na seção Rastreio do arquivo de configuração do aplicativo.

Se você não estiver usando um arquivo de configuração do aplicativo, o FFDC estará ativo e o rastreio desativado.

Dicas para Resolução de Problemas

Use estas dicas para ajudar a solucionar problemas com o uso do XMS.

O aplicativo Um XMS não pode se conectar a um gerenciador de filas (MQRC_NOT_AUTHORIZED)

O cliente XMS .NET pode ter um comportamento diferente do comportamento do cliente JMS do IBM WebSphere MQ . Portanto, você pode achar que seu aplicativo XMS não pode se conectar ao gerenciador de filas, embora seu aplicativo JMS possa.

- Uma solução simples para esse problema é tentar usar um ID de usuário que não tenha mais de 12 caracteres de comprimento e esteja autorizado completamente na lista de autoridade do gerenciador de filas. Se essa solução não for ideal, uma abordagem diferente, mas mais complexa, será a utilização de saídas de segurança. Se você precisar de ajuda adicional sobre esse problema, entre em contato com o Suporte IBM para obter assistência
- Se você configurar a propriedade XMSC_USERID do connection factory, ela deverá corresponder ao ID de usuário e à senha do usuário conectado. Se você não configurar essa propriedade, o gerenciador de filas usará o ID de usuário do usuário conectado, por padrão.
- A autenticação do usuário para o IBM WebSphere MQ é executada usando os detalhes do usuário atualmente conectado e não as informações fornecidas nos campos XMSC.USERID e XMSC.PASSWORD. Isso foi projetado para manter a consistência com o IBM WebSphere MQ. Para obter mais informações sobre autenticação, consulte *Informações sobre Autenticação* na documentação on-line do produto IBM IBM WebSphere MQ .

Conexão redirecionada para o mecanismo do sistema de mensagens

Quando você se conecta a um barramento de integração de serviços do WebSphere Servidor de Aplicação versão 6.0.2, todas as conexões podem ser redirecionadas do terminal do provedor original para o

mecanismo do sistema de mensagens escolhido pelo barramento para essa conexão do cliente. Ao fazer isso, ele sempre redirecionará a conexão para um servidor host especificado pelo nome do host, em vez de por um endereço IP Portanto, você pode ter problemas de conexão se o nome do host não puder ser resolvido

Para se conectar com sucesso ao barramento de integração de serviços da WebSphere Servidor de Aplicação Versão 6.0.2 , pode ser necessário fornecer um mapeamento entre os nomes do host e os endereços IP em sua máquina host do cliente Por exemplo, é possível especificar o mapeamento em uma tabela de hosts locais em sua máquina host do cliente.

Um aplicativo XMS usando um heap JVM maior

O aplicativo .NET XMS que envia mensagens por meio dos mecanismos do sistema de mensagens do WebSphere Servidor de Aplicação geralmente precisa usar um heap da JVM maior do que o padrão especificado Para alterar as definições de configuração de heap, consulte Ajustando o desempenho do sistema de mensagens com tecnologias de integração de serviços na documentação do produto WebSphere Application Server Versão 7

Suporte para autenticação de senha semelhante a telnet

O protocolo XMS .NET Real Time Transport suporta apenas autenticação de senha semelhante a telnet simples. O protocolo XMS .NET Real Time Transport não suporta Qualidade de proteção.

Configurando valores para o tipo de propriedade double

Em uma plataforma Windows de 64 bits, os métodos SetDoubleProperty () ou GetDoubleProperty () poderão não funcionar corretamente ao configurar ou obter valores para o tipo de propriedade double, se os valores forem menores que Double.Epsilon.

Por exemplo, se você tentar configurar um valor de 4.9E-324 para uma propriedade como tipo double, as plataformas Windows de 64 bits o tratam como 0.0. Portanto, em um ambiente de sistema de mensagens distribuído, se um JMS ou outro aplicativo configurar o valor para uma propriedade dupla como 4.9E-324 em qualquer máquina Unix ou Windows de 32 bits e XMS .NET executar em uma máquina de 64 bits, o valor retornado por GetDoubleProperty() será 0.0. Este é um problema conhecido no Microsoft .NET 2.0 Framework.

Clientes de serviço de mensagens para referência .NET

Esta seção de referência fornece informações para ajudá-lo a usar Message Service Client for .NET. Essas informações ajudam você a executar as tarefas envolvidas na programação com o XMS

Interfaces do .NET

Este tópicoseção documenta as interface de classe .NET e suas propriedades e métodos.

A tabela a seguir resume todas as interfaces, que são definidas no namespace `IBM.XMS` .

Tabela 27. Resumo das interfaces de classe .NET

Interface	Descrição
“IBytesMessage” na página 95	Uma mensagem de bytes é uma mensagem cujo corpo compreende um fluxo de bytes.
“IConnection” na página 106	Um objeto Connection representa a conexão ativa do aplicativo para um servidor de sistema de mensagens.
“IConnectionFactory” na página 109	Um aplicativo usa um connection factory para criar uma conexão.

Tabela 27. Resumo das interfaces de classe .NET (continuação)	
Interface	**Descrição**
“IConnectionMetaDados” na página 111	Um objeto de dados ConnectionMetafornece informações sobre uma conexão.
“IDestination” na página 111	Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.
“ExceptionListener” na página 113	Um aplicativo usa um listener de exceção para ser notificado assincronamente de um problema com uma conexão.
“Exceção IllegalState” na página 114	XMS lança essa exceção se um aplicativo chamar um método em um momento incorreto ou inadequado ou se XMS não estiver em um estado apropriado para a operação solicitada.
“InitialContext” na página 114	Um aplicativo usa um objeto InitialContext para criar objetos de definições de objeto que são recuperados de um repositório de objetos administrados.
“InvalidClientIDException” na página 117	XMS lança essa exceção se um aplicativo tentar configurar um identificador de cliente para uma conexão, mas o identificador de cliente não for válido ou já estiver em uso.
“Exceção de InvalidDestination” na página 117	XMS lançará essa exceção se um aplicativo especificar um destino que não seja válido
“InvalidSelectorExceção” na página 117	XMS emitirá essa exceção se um aplicativo fornecer uma expressão de seletor de mensagem cuja sintaxe não é válida
“IMapMessage” na página 118	Uma mensagem de mapa é uma mensagem cujo corpo consiste em um conjunto de pares nome-valor, em que cada valor possui um tipo de dados associado.
“IMessage” na página 127	Um objeto de mensagem representa uma mensagem que um aplicativo envia ou recebe. IMessage é uma superclasse para as classes de mensagem, como IMapMessage.
“IMessageConsumer” na página 134	Um aplicativo usa um consumidor de mensagem para receber mensagens enviadas a um destino.
“MessageEOFException” na página 136	XMS lança essa exceção se XMS encontrar o final de um fluxo de mensagens de bytes quando um aplicativo estiver lendo o corpo de uma mensagem de bytes.
“MessageFormatExceção” na página 137	XMS lançará essa exceção se o XMS encontrar uma mensagem com um formato que não seja válido
“IMessageListener (delegado)” na página 137	Um aplicativo usa um listener de mensagens para receber mensagens assincronamente.
“MessageNotReadableException” na página 138	XMS lança essa exceção se um aplicativo tentar ler o corpo de uma mensagem que é somente gravação.

Tabela 27. Resumo das interfaces de classe .NET (continuação)	
Interface	**Descrição**
“MessageNotWritableException” na página 138	XMS lança essa exceção se um aplicativo tentar gravar no corpo de uma mensagem que é somente leitura.
“IMessageProducer” na página 138	Um aplicativo usa um produtor de mensagem para enviar mensagens para um destino.
“IObjectMessage” na página 144	Uma mensagem de objeto é uma mensagem cujo corpo compreende um objeto Java ou .NET serializado.
“IPropertyContext” na página 145	IPropertyContext é uma superclasse abstrata que contém métodos que obtêm e configuram propriedades. Estes métodos são herdados por outras classes.
“IQueueBrowser” na página 155	Um aplicativo usa um navegador de filas para pesquisar mensagens em uma fila sem removê-las.
“Solicitante” na página 156	Um aplicativo usa um solicitante para enviar uma mensagem de solicitação e, em seguida, aguardar e receber a resposta..
“Exceção de ResourceAllocation” na página 158	XMS lança essa exceção se XMS não puder alocar os recursos necessários por um método.
“SecurityException” na página 158	XMS lança essa exceção se o identificador de usuário e a senha fornecidos para autenticar um aplicativo forem rejeitados XMS também lança essa exceção se uma verificação de autoridade falhar e impedir que um método seja concluído.
“ISession” na página 159	Uma sessão é um único contexto encadeado para enviar e receber mensagens.
“IStreamMessage” na página 169	Uma mensagem de fluxo é uma mensagem cujo corpo compreende um fluxo de valores, em que cada valor possui um tipo de dados associado.
“ITextMessage” na página 179	Uma mensagem de texto é uma mensagem cujo corpo compreende uma sequência.
“TransactionInProgressException” na página 180	XMS lançará essa exceção se um aplicativo solicitar uma operação que não seja válida porque uma transação está em andamento
“TransactionRolledBackException” na página 180	XMS lança esta exceção se um aplicativo chamar Session.commit() para confirmar a transação atual, mas a transação será, então, revertida.
XMSC	Para .NET, XMS nomes de propriedades e valores são definidos nessa classe como constantes públicas. Para obter detalhes adicionais, consulte “Propriedades de objetos XMS” na página 183.

Tabela 27. Resumo das interfaces de classe .NET (continuação)	
Interface	**Descrição**
"XMSException" na página 180	Se XMS detectar um erro ao processar uma chamada para um método .NET , XMS lançará uma exceção. Uma exceção é um objeto que contém informações sobre o erro. Há diferentes tipos de exceção XMS , e um objeto XMSException é apenas um tipo de excepção. Entretanto, a classe XMSException é uma superclasse das outras classes de exceção XMS . XMS lança um objeto XMSException em situações em que nenhum dos outros tipos de exceção é apropriado.
"XMSFactoryFactory" na página 181	Se um aplicativo não estiver usando objetos administrados, utilize essa classe para criar connection factories, filas e tópicos..

A definição de cada método lista os códigos de exceção que o XMS pode retornar se detectar um erro ao processar uma chamada para o método. Cada código de exceções é representado por sua constante nomeada, que possui uma exceção correspondente.

Conceitos relacionados

Construindo seus próprios aplicativos
Você constrói seus próprios aplicativos, como você constrói os aplicativos de amostra.

Gravando aplicativos do XMS
Este seçãocapítulo fornece informações para ajudá-lo na gravação de aplicativos XMS

Gravando aplicativos XMS .NET
Este seçãocapítulo fornece informações para ajudá-lo ao gravar aplicativos do XMS.NET

Referências relacionadas

Propriedades de objetos XMS
Isso seçãocapítulo documenta as propriedades de objeto definidas por XMS

IBytesMessage

Uma mensagem de bytes é uma mensagem cujo corpo compreende um fluxo de bytes.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.IBytesMessage
```

Referências relacionadas

Mensagens de Bytes
O corpo de uma mensagem de bytes contém um fluxo de bytes. O corpo contém apenas os dados reais, e é responsabilidade dos aplicativos de envio e recebimento interpretarem esses dados.

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
BodyLength	Obtenha o comprimento do corpo da mensagem em bytes quando o corpo da mensagem for somente leitura

BodyLength -Obter Comprimento do Corpo

Interface:

```
Int64 BodyLength
    {
      get;
    }
```

Obtenha o comprimento do corpo da mensagem em bytes quando o corpo da mensagem for somente leitura

O valor retornado é o comprimento do corpo inteiro, independentemente de onde o cursor para ler a mensagem está atualmente posicionado.

Exceções:

- XMSException
- MessageNotReadableException

Methods

Resumo dos métodos:

Método	Descrição
ReadBoolean	Leia um valor booleano do fluxo de mensagens de bytes.
ReadSignedReadSigned	Leia o próximo byte do fluxo de mensagens de bytes como um número inteiro de 8 bits assinado.
ReadBytes	Ler uma matriz de bytes a partir do fluxo de mensagens de bytes a partir da posição atual do cursor.
ReadChar	Leia os próximos 2 bytes do fluxo de mensagens de bytes como um caractere.
ReadDouble	Leia os próximos 8 bytes do fluxo de mensagens de bytes como um número de ponto flutuante de precisão dupla.
ReadFloat	Leia os próximos 4 bytes do fluxo de mensagens de bytes como um número de ponto flutuante.
ReadInt	Leia os próximos 4 bytes do fluxo de mensagens de bytes como um número inteiro de 32 bits assinado.
ReadLong	Leia os próximos 8 bytes do fluxo de mensagens de bytes como um número inteiro de 64 bits assinado.
ReadShort	Leia os próximos 2 bytes do fluxo de mensagens de bytes como um número inteiro de 16 bits assinado.
ReadByte	Leia o próximo byte do fluxo de mensagens de bytes como um número inteiro de 8 bits não assinado.
ReadUnsignedShort	Leia os próximos 2 bytes do fluxo de mensagens de bytes como um número inteiro de 16 bits não assinado.

Método	Descrição
ReadUTF	Leia uma cadeia, codificada em UTF-8, a partir do fluxo de mensagens de bytes.
Reconfigurar	Coloque o corpo da mensagem no modo somente leitura e reposicione o cursor no início do fluxo de mensagem de bytes..
WriteBoolean	Grave um valor booleano no fluxo de mensagens de bytes.
WriteByte	Gravar um byte no fluxo de mensagens de bytes.
WriteBytes	Grave uma matriz de bytes no fluxo de mensagens de bytes.
WriteBytes	Grave uma matriz parcial de bytes no fluxo de mensagens de bytes, conforme definido pelo comprimento especificado.
WriteChar	Grave um caractere no fluxo de mensagens de bytes como 2 bytes, primeiro byte de alta ordem.
WriteDouble	Converta um número de vírgula flutuante de precisão dupla em um número inteiro longo e grave o número inteiro longo no fluxo de mensagens de bytes como 8 bytes, primeiro byte de alta ordem..
WriteFloat	Converta um número de vírgula flutuante em um número inteiro e grave o número inteiro no fluxo de mensagens de bytes como 4 bytes, primeiro byte de alta ordem.
WriteInt	Grave um número inteiro no fluxo de mensagens de bytes como 4 bytes, primeiro byte de alta ordem.
WriteLong	Grave um número inteiro longo no fluxo de mensagens de bytes como 8 bytes, primeiro byte de alta ordem.
WriteObject	Grave o objeto especificado no fluxo da mensagem de byte
WriteShort	Grave um número inteiro curto no fluxo de mensagens de bytes como 2 bytes, primeiro byte de alta ordem.
WriteUTF	Grave uma sequência, codificada em UTF-8, para o fluxo de mensagens de bytes

ReadBoolean -Valor Booleano de leitura

Interface:

```
Boolean ReadBoolean();
```

Leia um valor booleano do fluxo de mensagens de bytes.

Parâmetros:
Nenhum

Retorna:
O valor booleano que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadSignedByte-Byte de Leitura

Interface:

```
Int16   ReadSignedByte();
```

Leia o próximo byte do fluxo de mensagens de bytes como um número inteiro de 8 bits assinado.

Parâmetros:
Nenhum

Retorna:
O byte lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadBytes -Bytes de leitura

Interface:

```
Int32   ReadBytes(Byte[] array);
Int32   ReadBytes(Byte[] array, Int32 length);
```

Ler uma matriz de bytes a partir do fluxo de mensagens de bytes a partir da posição atual do cursor.

Parâmetros:

matriz (saída)
O buffer para conter a matriz de bytes que é lida. Se o número de bytes restantes a serem lidos do fluxo antes da chamada for maior ou igual ao comprimento do buffer, o buffer será preenchido. Caso contrário, o buffer será parcialmente preenchido, com todos os bytes restantes.

Se você especificar um ponteiro nulo na entrada, o método ignorará os bytes sem lê-los. Se o número de bytes restantes a serem lidos a partir do fluxo antes da chamada for maior ou igual ao comprimento do buffer, o número de bytes ignorados será igual ao comprimento do buffer. Caso contrário, todos os bytes restantes serão ignorados.. O cursor é deixado na próxima posição para ser lido no fluxo de mensagens de byte

comprimento (entrada)
O comprimento do buffer em bytes

Retorna:
O número de bytes lidos no buffer. Se o buffer estiver parcialmente preenchido, o valor será menor que o comprimento do buffer, indicando que não há mais bytes restantes para serem lidos. Se não houver bytes restantes a serem lidos do fluxo antes da chamada, o valor será `XMSC_END_OF_STREAM`.

Se você especificar um ponteiro nulo na entrada, o método não retornará valor.

Exceções:

- XMSException
- MessageNotReadableException

ReadChar -Caractere de Leitura

Interface:

```
Char    ReadChar();
```

Leia os próximos 2 bytes do fluxo de mensagens de bytes como um caractere.

Parâmetros:
Nenhum

Retorna:
O caractere lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadDouble -Ler Número de Ponto Flutuante de Precisão Dupla

Interface:

```
Double  ReadDouble();
```

Leia os próximos 8 bytes do fluxo de mensagens de bytes como um número de ponto flutuante de precisão dupla.

Parâmetros:
Nenhum

Retorna:
O número do ponto flutuante de precisão dupla que é lido

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadFloat -Ler número de ponto flutuante

Interface:

```
Single  ReadFloat();
```

Leia os próximos 4 bytes do fluxo de mensagens de bytes como um número de ponto flutuante.

Parâmetros:
Nenhum

Retorna:
O número de ponto flutuante que é lido

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadInt -Ler Número Inteiro

Interface:

```
Int32   ReadInt();
```

Leia os próximos 4 bytes do fluxo de mensagens de bytes como um número inteiro de 32 bits assinado.

Parâmetros:
Nenhum

Retorna:
O número inteiro que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadLong -Número inteiro longo de leitura

Interface:

```
Int64   ReadLong();
```

Leia os próximos 8 bytes do fluxo de mensagens de bytes como um número inteiro de 64 bits assinado.

Parâmetros:
Nenhum

Retorna:
O número inteiro longo que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadShort -Número Inteiro Curto de Leitura

Interface:

```
Int16   ReadShort();
```

Leia os próximos 2 bytes do fluxo de mensagens de bytes como um número inteiro de 16 bits assinado.

Parâmetros:
Nenhum

Retorna:
O número inteiro curto que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadByte -Byte não assinado de leitura

Interface:

```
Byte    ReadByte();
```

Leia o próximo byte do fluxo de mensagens de bytes como um número inteiro de 8 bits não assinado.

Parâmetros:
Nenhum

Retorna:
O byte lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadUnsignedShort-Read Unsigned Short Integer

Interface:

```
Int32   ReadUnsignedShort();
```

Leia os próximos 2 bytes do fluxo de mensagens de bytes como um número inteiro de 16 bits não assinado.

Parâmetros:
Nenhum

Retorna:
O número inteiro curto não assinado que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadUTF -Ler Sequência UTF

Interface:

```
String  ReadUTF();
```

Leia uma cadeia, codificada em UTF-8, a partir do fluxo de mensagens de bytes.

Nota: Antes de chamar ReadUTF(), assegure que o cursor do buffer esteja apontando para o início do fluxo de mensagens de byte.

Parâmetros:
Nenhum

Retorna:
Um objeto String encapsulando a sequência que é lida.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

Reconfigurar-Reconfigurar

Interface:

```
void Reset();
```

Coloque o corpo da mensagem no modo somente leitura e reposicione o cursor no início do fluxo de mensagem de bytes..

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotReadableException

WriteBoolean -Gravar Valor Booleano

Interface:

```
void  WriteBoolean(Boolean value);
```

Grave um valor booleano no fluxo de mensagens de bytes.

Parâmetros:

valor (entrada)
O valor booleano a ser gravado.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteByte -Byte de gravação

Interface:

```
void  WriteByte(Byte value);
void  WriteSignedByte(Int16 value);
```

Gravar um byte no fluxo de mensagens de bytes.

Parâmetros:

valor (entrada)
O byte a ser gravado..

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteBytes -Bytes de Gravação

Interface:

```
void  WriteBytes(Byte[] value);
```

Grave uma matriz de bytes no fluxo de mensagens de bytes.

Parâmetros:

valor (entrada)
A matriz de bytes a ser gravada

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteBytes -Matriz de bytes parciais de gravação

Interface:

```
void  WriteBytes(Byte[] value, int offset, int length);
```

Grave uma matriz parcial de bytes no fluxo de mensagens de bytes, conforme definido pelo comprimento especificado.

Parâmetros:

valor (entrada)
A matriz de bytes a ser gravada

deslocamento (entrada)
O ponto de início para a matriz de bytes a ser gravada

comprimento (entrada)
O número de bytes a serem gravados

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteChar -Caractere de gravação

Interface:

```
void  WriteChar(Char value);
```

Grave um caractere no fluxo de mensagens de bytes como 2 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O caractere a ser gravado

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteDouble -Número do ponto flutuante de precisão dupla de gravação

Interface:

```
void  WriteDouble(Double value);
```

Converta um número de vírgula flutuante de precisão dupla em um número inteiro longo e grave o número inteiro longo no fluxo de mensagens de bytes como 8 bytes, primeiro byte de alta ordem..

Parâmetros:

valor (entrada)
O número de ponto flutuante de precisão dupla a ser gravado

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteFloat -Número do ponto flutuante de gravação

Interface:

```
void  WriteFloat(Single value);
```

Converta um número de vírgula flutuante em um número inteiro e grave o número inteiro no fluxo de mensagens de bytes como 4 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O número de ponto flutuante a ser gravado

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteInt -Número inteiro de gravação

Interface:

```
void  WriteInt(Int32 value);
```

Grave um número inteiro no fluxo de mensagens de bytes como 4 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O número inteiro a ser gravado.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteLong -Número inteiro longo de gravação

Interface:

```
void  WriteLong(Int64 value);
```

Grave um número inteiro longo no fluxo de mensagens de bytes como 8 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O número inteiro longo a ser gravado.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteObject -Gravar Objeto

Interface:

```
void  WriteObject(Object value);
```

Grave o objeto especificado no fluxo da mensagem de byte

Parâmetros:

valor (entrada)
O objeto a ser gravado, que deve ser uma referência a um tipo primitivo..

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteShort -Escrever Número Inteiro Curto

Interface:

```
void  WriteShort(Int16 value);
```

Grave um número inteiro curto no fluxo de mensagens de bytes como 2 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O número inteiro curto a ser gravado.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteUTF -Gravar sequência UTF

Interface:

```
void  WriteUTF(String value);
```

Grave uma sequência, codificada em UTF-8, para o fluxo de mensagens de bytes

Parâmetros:

valor (entrada)
Um objeto String encapsulando a sequência a ser gravada.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

Propriedades e métodos herdados

As propriedades a seguir são herdadas da interface IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Os métodos a seguir são herdados da interface IMessage:

clearBody, clearProperties, PropertyExists

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty,

SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IConnection

Um objeto Connection representa a conexão ativa do aplicativo para um servidor de sistema de mensagens.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IConnection
```

Para obter uma lista das propriedades definidas XMS de um objeto Connection, consulte "Propriedades da Conexão" na página 184.

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
ClientID	Obter e configurar o identificador de cliente para a conexão.
ExceptionListener	Obtenha o listener de exceção que está registrado com a conexão e registre um listener de exceção com a conexão.
MetaData	Obter os metadados da conexão.

ClientID -Obter e Configurar ID do Cliente

Interface:

```
String ClientID
      {
        get;
        set;
      }
```

Obter e configurar o identificador de cliente para a conexão.

O identificador de cliente pode ser pré-configurado pelo administrador em um ConnectionFactoryou designado configurando ClientID.

Um identificador de cliente é usado apenas para suportar assinaturas duráveis no domínio de publicação / assinatura e é ignorado no domínio ponto a ponto.

Se um aplicativo configurar um identificador de cliente para uma conexão, o aplicativo deverá fazer isso imediatamente após criar a conexão e antes de executar qualquer outra operação na conexão. Se o aplicativo tentar configurar um identificador de cliente após esse ponto, a chamada emitirá exceção IllegalStateException.

Essa propriedade não é válida para uma conexão em tempo real com um broker

Exceções:

- XMSException
- Exceção IllegalState
- InvalidClientIDException

ExceptionListener -Obter e Configurar Listener de Exceção

Interface:

```
    ExceptionListener ExceptionListener
        {
          get;
```

```
        set;
      }
```

Obtenha o listener de exceção que está registrado com a conexão e registre um listener de exceção com a conexão.

Se nenhum listener de exceção for registrado com a conexão, o método retorna nulo. Se um listener de exceção já estiver registrado com a conexão, será possível cancelar o registro especificando um nulo em vez do listener de exceção.

Para obter mais informações sobre como usar listeners de exceções, consulte "Listeners de mensagem e de exceção em .NET" na página 50

Exceções:

- XMSException

Metadados-Obter metadados

Interface:

```
IConnectionMetaData MetaData
    {
      get;
    }
```

Obter os metadados da conexão.

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
Fechar	Feche a conexão.
CreateSession	Criar uma sessão
Início	Iniciar ou reiniciar a entrega de mensagens recebidas para a conexão.
Parar	Pare a entrega de mensagens recebidas para a conexão.

Fechar-Fechar Conexão

Interface:

```
void  Close();
```

Feche a conexão.

Se um aplicativo tentar fechar uma conexão que já esteja fechada, a chamada será ignorada..

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException

CreateSession -Criar Sessão.

Interface:

```
    ISession CreateSession(Boolean transacted,
                        AcknowledgeMode acknowledgeMode);
```

Criar uma sessão

Parâmetros:

transacionado (entrada)
O valor `True` significa que a sessão é transacionada O valor `False` significa que a sessão não foi transacionada.

Para uma conexão em tempo real com um broker, o valor deve ser `False`.

acknowledgeMode (entrada).
Indica como mensagens recebidas por um aplicativo são reconhecidas. O valor deve ser um dos seguintes do enumerador AcknowledgeMode :

`AcknowledgeMode.AutoAcknowledge`
`AcknowledgeMode.ClientAcknowledge`
`AcknowledgeMode.DupsOkAcknowledge`

Para uma conexão em tempo real com um broker, o valor deve ser `AcknowledgeMode.AutoAcknowledge` ou `AcknowledgeMode.DupsOkAcknowledge`

Esse parâmetro será ignorado se a sessão for transacionada.. Para obter mais informações sobre os modos de confirmação, consulte “Confirmação da mensagem..” na página 28

Retorna:
O objeto Session

Exceções:

- XMSException

Iniciar-Iniciar Conexão

Interface:

```
void  Start();
```

Iniciar ou reiniciar a entrega de mensagens recebidas para a conexão. A chamada será ignorada se a conexão já tiver sido iniciada

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException

Parar-Parar Conexão

Interface:

```
void  Stop();
```

Pare a entrega de mensagens recebidas para a conexão. A chamada será ignorada se a conexão já tiver sido interrompida

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IConnectionFactory

Um aplicativo usa um connection factory para criar uma conexão.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IConnectionFactory
```

Para obter uma lista das propriedades definidas XMS de um objeto ConnectionFactory , consulte “Propriedades de ConnectionFactory” na página 185.

Conceitos relacionados

ConnectionFactories e objetos de Conexão
Um objeto ConnectionFactory fornece um modelo que um aplicativo usa para criar um objeto Connection. O aplicativo usa o objeto Connection para criar um objeto Session.

Conexão com um Barramento de Integração de Serviços do WebSphere
O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

Conexões seguras para um IBM WebSphere MQ gerenciador de filas
Para permitir que um aplicativo XMS .NET faça conexões seguras para um IBM WebSphere MQ gerenciador de filas, as propriedades relevantes devem ser definidas no objeto ConnectionFactory .

Conexões seguras para um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus
Para ativar um XMS NET para fazer conexões seguras com um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus , as propriedades relevantes devem ser definidas no objeto ConnectionFactory .

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Methods

Resumo dos métodos:

Método	Descrição
CreateConnection	Crie um connection factory com as propriedades padrão..
CreateConnection	Crie uma conexão usando uma identidade do usuário especificada

CreateConnection -Criar Connection Factory (utilizando a identidade do usuário padrão).

Interface:

```
IConnection CreateConnection();
```

Crie um connection factory com as propriedades padrão..

Se você estiver se conectando ao IBM WebSphere MQe configurar a propriedade XMSC_USERID do connection factory, ele deverá corresponder ao **userid** do usuário conectado. Se você não configurar essas propriedades, o gerenciador de filas usará o **userid** do usuário conectado por padrão. If you require further connection-level authentication of individual users you can write a client authentication exit which is configured in IBM WebSphere MQ.

Parâmetros:
Nenhum

Exceções:

- XMSException

CreateConnection -Criar Conexão (usando uma identidade do usuário especificada).

Interface:

```
IConnection CreateConnection(String userId, String password);
```

Crie uma conexão usando uma identidade do usuário especificada

Se você estiver se conectando ao IBM WebSphere MQe configurar a propriedade XMSC_USERID do connection factory, ele deverá corresponder ao **userid** do usuário conectado. Se você não configurar essas propriedades, o gerenciador de filas usará o **userid** do usuário conectado por padrão. If you require further connection-level authentication of individual users you can write a client authentication exit which is configured in IBM WebSphere MQ.

A conexão é criada em modo interrompido Nenhuma mensagem é entregue até que o aplicativo chame **Connection.start()**

Parâmetros:

userID (entrada)
Um objeto String que encapsula o identificador de usuário a ser usado para autenticar o aplicativo.. Se você fornecer um valor nulo, será feita uma tentativa de criar a conexão sem autenticação.

senha (entrada)
Um objeto String encapsulando a senha a ser usada para autenticar o aplicativo. Se você fornecer um valor nulo, será feita uma tentativa de criar a conexão sem autenticação.

Retorna:
O objeto Conexão.

Exceções:

- XMSException
- XMS_X_SECURITY_EXCEPTION

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IConnectionMetaDados

Um objeto de dados ConnectionMetafornece informações sobre uma conexão.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IConnectionMetaData
```

Para obter uma lista das propriedades definidas pelo XMS de um objeto de dados ConnectionMeta, consulte "Propriedades de Dados ConnectionMeta" na página 192

Propriedades .NET

Resumo das propriedades:

Método	Descrição
JMSXPropertyNames	Retornar uma enumeração dos nomes das propriedades de mensagens definidas JMS suportadas pela conexão.

JMSXPropertyNames -Obter Propriedades de Mensagem Definidas JMS

Interface:

```
System.Collections.IEnumerator JMSXPropertyNames
    {
      get;
    }
```

Retornar uma enumeração dos nomes das propriedades de mensagens definidas JMS suportadas pela conexão.

JMS propriedades de mensagem definidas não são suportadas por uma conexão em tempo real com um broker.

Exceções:

- XMSException

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IDestination

Um destino é para onde um aplicativo envia mensagens ou é uma origem da qual um aplicativo recebe mensagens, ou ambos.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IDestination
```

Para obter uma lista das propriedades definidas do XMS de um objeto de Destino, consulte “Propriedades de Destino” na página 192

Conceitos relacionados

ConnectionFactories e objetos de Conexão
Um objeto ConnectionFactory fornece um modelo que um aplicativo usa para criar um objeto Connection. O aplicativo usa o objeto Connection para criar um objeto Session.

Conexão com um Barramento de Integração de Serviços do WebSphere
O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

Destinos
Um aplicativo XMS usa um objeto de Destino para especificar o destino das mensagens que estão sendo enviadas e a origem de mensagens que estão sendo recebidas.

Curinga de destino
O XMS fornece suporte para curingas de destino, assegurando que os curingas possam ser transmitidos para o local no qual eles são necessários para correspondência Há um esquema curinga diferente para cada tipo de servidor com o qual XMS pode trabalhar.

Identificadores de recursos uniformes do tópico
O URI (Identificador Uniforme de Recursos (URI) do tópico especifica o nome do tópico; ele também pode especificar uma ou mais propriedades para ele.

Identificadores uniformes de recursos da fila
O URI para uma fila especifica o nome da fila; ele também pode especificar uma ou mais propriedades da fila.

Destinos Temporários
Os aplicativos XMS podem criar e usar destinos temporários.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Referências relacionadas

Propriedades Necessárias para Objetos de Destino Administrados
Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

Propriedades .NET

Resumo dos métodos:

Método	Descrição
Nome	Obtenha o nome do destino
TypeId	Obtenha o tipo do destino

Nome-Obter Nome do Destino

Interface:

```
String Name
{
  get;
}
```

Obtenha o nome do destino O nome é uma sequência encapsulando o nome de uma fila ou o nome de um tópico..

Exceções:

- XMSException

TypeId -Obter Tipo de Destino

Interface:

```
DestinationType TypeId
{
  get;
}
```

Obtenha o tipo do destino O tipo do destino é um dos seguintes valores:

```
DestinationType.Queue
DestinationType.Topic
```

Exceções:

- XMSException

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

ExceptionListener

Hierarquia de herança:

Nenhum

Um aplicativo usa um listener de exceção para ser notificado assincronamente de um problema com uma conexão.

Se um aplicativo usar uma conexão apenas para consumir mensagens de forma assíncrona e sem outro propósito, a única maneira de o aplicativo aprender sobre um problema com a conexão será usando um listener de exceção. Em outras situações, um listener de exceção pode fornecer uma maneira mais imediata de aprender sobre um problema com uma conexão do que esperar até a próxima chamada síncrona para XMS..

Delegar

Resumo do delegado:

Delegar	Descrição
ExceptionListener	Notificar o aplicativo de um problema com uma conexão.

ExceptionListener -Listener de Exceção

Interface:

```
public delegate void ExceptionListener(Exception ex)
```

Notificar o aplicativo de um problema com uma conexão.

Os métodos que implementam esse delegado podem ser registrados com a conexão

Para obter mais informações sobre como usar listeners de exceções, consulte "Listeners de mensagem e de exceção em .NET" na página 50

Parâmetros:

exceção (entrada)
Um ponteiro para uma exceção criada por XMS

Retorna:
Cancelado

Exceção IllegalState

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.Exception
           |
           +----IBM.XMS.IllegalStateException
```

XMS lança essa exceção se um aplicativo chamar um método em um momento incorreto ou inadequado ou se XMS não estiver em um estado apropriado para a operação solicitada.

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

InitialContext

Um aplicativo usa um objeto InitialContext para criar objetos de definições de objeto que são recuperados de um repositório de objetos administrados.

Hierarquia de herança:
Nenhum

Conceitos relacionados

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Tarefas relacionadas

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
ambiente	Obter o ambiente

Ambiente-Obter o ambiente

Interface:

```
Hashtable Environment
{
     get;
}
```

Obter o ambiente

Exceções:

- As exceções são específicas para o serviço de diretório que está sendo usado

Construtores

Resumo de construtores:

Construtor	Descrição
InitialContext	Crie um objeto InitialContext ..

InitialContext -Criar Contexto Inicial

Interface:

```
InitialContext(Hashtable env);
```

Crie um objeto InitialContext ..

Parâmetros:

As informações necessárias para estabelecer uma conexão com o repositório de objetos administrados são fornecidas ao construtor em um ambiente Hashtable.

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
AddToAmbiente	Inclua uma nova propriedade no ambiente
Fechar	Feche este contexto.
Consulta	Crie um objeto a partir de uma definição de objeto que é recuperado do repositório de objetos administrados.
RemoveFromAmbiente	Remova uma propriedade do ambiente.

Ambiente AddTo-Incluir uma nova propriedade no ambiente

Interface:

```
Object AddToEnvironment(String propName, Object propVal);
```

Inclua uma nova propriedade no ambiente

Parâmetros:

propName (entrada)
Um objeto String encapsulando o nome da propriedade a ser incluída.

propVal (entrada)
O valor da propriedade a ser incluída.

Retorna:
O valor antigo da propriedade.

Exceções:

- As exceções são específicas para o serviço de diretório que está sendo usado

Fechar-Fechar este contexto

Interface:

```
void Close()
```

Feche este contexto.

Parâmetros:
Nenhum

Retorna:
Nenhum

Exceções:

- As exceções são específicas para o serviço de diretório que está sendo usado

Consulta-objeto de consulta no contexto inicial

Interface:

```
Object Lookup(String name);
```

Crie um objeto a partir de uma definição de objeto que é recuperado do repositório de objetos administrados.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome do objeto administrado a ser recuperado. O nome pode ser um nome simples ou um nome complexo. Para obter detalhes adicionais, consulte “Recuperação de Objetos Administrados” na página 66.

Retorna:
Um IConnectionFactory ou um IDestination, dependendo do tipo de objeto sendo recuperado. Se a função puder acessar o diretório, mas não puder localizar o objeto necessário, um nulo será retornado..

Exceções:

- As exceções são específicas para o serviço de diretório que está sendo usado

Ambiente RemoveFrom-Remover uma Propriedade do Ambiente.

Interface:

```
Object RemoveFromEnvironment(String propName);
```

Remova uma propriedade do ambiente.

Parâmetros:

propName (entrada)
Um objeto String encapsulando o nome da propriedade a ser removida.

Retorna:
O objeto que foi removido..

Exceções:

- As exceções são específicas para o serviço de diretório que está sendo usado

InvalidClientIDException

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.InvalidClientIDException
```

XMS lança essa exceção se um aplicativo tentar configurar um identificador de cliente para uma conexão, mas o identificador de cliente não for válido ou já estiver em uso.

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

Exceção de InvalidDestination

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.InvalidDestinationException
```

XMS lançará essa exceção se um aplicativo especificar um destino que não seja válido

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

InvalidSelectorExceção

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.InvalidSelectorException
```

XMS emitirá essa exceção se um aplicativo fornecer uma expressão de seletor de mensagem cuja sintaxe não é válida

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

IMapMessage

Uma mensagem de mapa é uma mensagem cujo corpo consiste em um conjunto de pares nome-valor, em que cada valor possui um tipo de dados associado.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.IMapMessage
```

Quando um aplicativo obtém o valor do par nome-valor, o valor pode ser convertido por XMS em outro tipo de dado.. Para obter mais informações sobre essa forma de conversão implícita, consulte “Mensagens de Mapa” na página 80

Referências relacionadas

Mensagens de Mapa
O corpo de uma mensagem de mapa contém um conjunto de pares nome-valor, em que cada valor tem um tipo de dados associado.

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
MapNames	Obter uma enumeração dos nomes no corpo da mensagem de mapa.

MapNames -Obter nomes de mapa

Interface:

```
System.Collections.IEnumerator MapNames
{
  get;
}
```

Obter uma enumeração dos nomes no corpo da mensagem de mapa.

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
GetBoolean	Obtenha o valor booleano identificado pelo nome do corpo da mensagem do mapa.
GetByte	Obtenha o byte identificado por nome a partir do corpo da mensagem do mapa
GetBytes	Obtenha a matriz de bytes identificados por nome a partir do corpo da mensagem do mapa
GetChar	Obter o caractere identificado pelo nome a partir do corpo da mensagem do mapa
GetDouble	Obter o número de vírgula flutuante de precisão dupla identificado por nome a partir do corpo da mensagem do mapa
GetFloat	Obtenha o número de vírgula flutuante identificado por nome a partir do corpo da mensagem do mapa

Método	Descrição
GetInt	Obter o número inteiro identificado por nome do corpo da mensagem do mapa.
GetLong	Obter o número inteiro longo identificado pelo nome do corpo da mensagem do mapa.
GetObject	Obter uma referência para o valor de um par nome-valor a partir do corpo da mensagem do mapa.
GetShort	Obter o número inteiro curto identificado por nome a partir do corpo da mensagem do mapa.
GetString	Obtenha a sequência identificada pelo nome a partir do corpo da mensagem do mapa
ItemExists	Verifique se o corpo da mensagem de mapa contém um par nome-valor com o nome especificado.
SetBoolean	Configure um valor booleano no corpo da mensagem de mapa.
SetByte	Configure um byte no corpo da mensagem do mapa
SetBytes	Configure uma matriz de bytes no corpo da mensagem de mapa
SetChar	Configure um caractere de 2 bytes no corpo da mensagem de mapeamento.
SetDouble	Configure um número de ponto flutuante de precisão dupla no corpo da mensagem do mapa.
SetFloat	Configure um número de vírgula flutuante no corpo da mensagem de mapa
SetInt	Configure um número inteiro no corpo da mensagem do mapa.
SetLong	Configure um número inteiro longo no corpo da mensagem de mapa
SetObject	Configure um valor, que deve ser um tipo primitivo XMS , no corpo da mensagem do mapa.
SetShort	Configure um número inteiro curto no corpo da mensagem do mapa.
SetString	Configure uma sequência no corpo da mensagem do mapa.

GetBoolean -Obter Valor Booleano

Interface:

```
Boolean GetBoolean(String name);
```

Obtenha o valor booleano identificado pelo nome do corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica o valor booleano.

Retorna:
O valor booleano recuperado do corpo da mensagem de mapa.

Exceções:

- XMSException

GetByte -Obter Byte.

Interface:

```
Byte    GetByte(String name);
    Int16   GetSignedByte(String name);
```

Obtenha o byte identificado por nome a partir do corpo da mensagem do mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica o byte.

Retorna:
O byte recuperado do corpo da mensagem de mapa. Nenhuma conversão de dados é executada no byte..

Exceções:

- XMSException

GetBytes -Obter Bytes

Interface:

```
Byte[]  GetBytes(String name);
```

Obtenha a matriz de bytes identificados por nome a partir do corpo da mensagem do mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica a matriz de bytes..

Retorna:
O número de bytes na matriz

Exceções:

- XMSException

GetChar -Obter Caractere

Interface:

```
Char    GetChar(String name);
```

Obter o caractere identificado pelo nome a partir do corpo da mensagem do mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica o caractere.

Retorna:
O caractere recuperado do corpo da mensagem do mapa.

Exceções:

- XMSException

GetDouble -Obter Número de Ponto Flutuante de Precisão Dupla

Interface:

```
Double  GetDouble(String name);
```

Obter o número de vírgula flutuante de precisão dupla identificado por nome a partir do corpo da mensagem do mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica o número de vírgula flutuante de precisão dupla.

Retorna:
O número de vírgula flutuante de dupla precisão recuperado do corpo de mensagem do mapa

Exceções:

- XMSException

GetFloat -Obter número de ponto flutuante

Interface:

```
Single  GetFloat(String name);
```

Obtenha o número de vírgula flutuante identificado por nome a partir do corpo da mensagem do mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica o número do ponto flutuante.

Retorna:
O número de ponto flutuante recuperado do corpo da mensagem do mapa

Exceções:

- XMSException

GetInt -Obter Número Inteiro

Interface:

```
Int32   GetInt(String name);
```

Obter o número inteiro identificado por nome do corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica o número inteiro.

Retorna:
O número inteiro recuperado do corpo da mensagem de mapa.

Exceções:

- XMSException

GetLong -Obter Número Inteiro Longo

Interface:

```
Int64   GetLong(String name);
```

Obter o número inteiro longo identificado pelo nome do corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica o número inteiro longo.

Retorna:
O número inteiro longo recuperado do corpo da mensagem do mapa.

Exceções:

- XMSException

GetObject -Obter Objeto

Interface:

```
Object  GetObject(String name);
```

Obter uma referência para o valor de um par nome-valor a partir do corpo da mensagem do mapa. O par de nome-valor é identificado por nome.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome do par nome-valor.

Retorna:
O valor, que é um dos seguintes tipos de objeto:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

Exceções:
XMSException

GetShort -Obter Número Inteiro Curto

Interface:

```
Int16   GetShort(String name);
```

Obter o número inteiro curto identificado por nome a partir do corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica o número inteiro curto.

Retorna:
O número inteiro curto recuperado do corpo da mensagem do mapa.

Exceções:

- XMSException

GetString -Obter sequência

Interface:

```
String  GetString(String name);
```

Obtenha a sequência identificada pelo nome a partir do corpo da mensagem do mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome que identifica a sequência no corpo da mensagem do mapa.

Retorna:
Um objeto String encapsulando a sequência recuperada do corpo de uma mensagem de mapa Se a conversão de dados for necessária, esse valor será a cadeia após a conversão..

Exceções:

- XMSException

ItemExists -Verifique se o par nome-valor existe

Interface:

```
Boolean ItemExists(String name);
```

Verifique se o corpo da mensagem de mapa contém um par nome-valor com o nome especificado.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome do par nome-valor.

Retorna:

- `True`, se o corpo da mensagem de mapa contiver um par nome-valor com o nome especificado.
- `False`, se o corpo da mensagem de mapa não contiver um par nome-valor com o nome especificado.

Exceções:

- XMSException

SetBoolean -Configurar Valor Booleano

Interface:

```
void  SetBoolean(String name, Boolean value);
```

Configure um valor booleano no corpo da mensagem de mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o valor booleano no corpo da mensagem do mapa..

valor (entrada)
O valor booleano a ser configurado.

Retorna:
Cancelado

Exceções:

- XMSException

SetByte -Configurar Byte

Interface:

```
void  SetByte(String name, Byte value);
void  SetSignedByte(String name, Int16 value);
```

Configure um byte no corpo da mensagem do mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o byte no corpo de uma mensagem de mapa

valor (entrada)
O byte a ser configurado..

Retorna:
Cancelado

Exceções:

- XMSException

SetBytes -Configurar Bytes

Interface:

```
void  SetBytes(String name, Byte[] value);
```

Configure uma matriz de bytes no corpo da mensagem de mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar a matriz de bytes no corpo da mensagem do mapa

valor (entrada)
A matriz de bytes a ser configurada

Retorna:
Cancelado

Exceções:

- XMSException

SetChar -Configurar Caractere

Interface:

```
void  SetChar(String name, Char value);
```

Configure um caractere de 2 bytes no corpo da mensagem de mapeamento.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o caractere no corpo da mensagem do mapa

valor (entrada)
O caractere a ser configurado.

Retorna:
Cancelado

Exceções:

- XMSException

SetDouble -Configurar Número de Ponto Flutuante de Precisão Dupla

Interface:

```
void  SetDouble(String name, Double value);
```

Configure um número de ponto flutuante de precisão dupla no corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o número do ponto flutuante de precisão dupla no corpo da mensagem do mapa

valor (entrada)
O número do ponto flutuante de precisão dupla a ser configurado

Retorna:
Cancelado

Exceções:

- XMSException

SetFloat -Configurar número de ponto flutuante

Interface:

```
void  SetFloat(String name, Single value);
```

Configure um número de vírgula flutuante no corpo da mensagem de mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o número de ponto flutuante no corpo da mensagem do mapa

valor (entrada)
O número de vírgula flutuante a ser configurado

Retorna:
Cancelado

Exceções:

- XMSException

SetInt -Configurar Número Inteiro

Interface:

```
void  SetInt(String name, Int32 value);
```

Configure um número inteiro no corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o número inteiro no corpo da mensagem do mapa.

valor (entrada)
O número inteiro a ser configurado.

Retorna:
Cancelado

Exceções:

- XMSException

SetLong -Configurar número inteiro longo

Interface:

```
void  SetLong(String name, Int64 value);
```

Configure um número inteiro longo no corpo da mensagem de mapa

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o número inteiro longo no corpo da mensagem do mapa..

valor (entrada)
O número inteiro longo a ser configurado.

Retorna:
Cancelado

Exceções:

- XMSException

SetObject -Configurar Objeto

Interface:

```
void  SetObject(String name, Object value);
```

Configure um valor, que deve ser um tipo primitivo XMS , no corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o valor no corpo da mensagem do mapa.

valor (entrada)
Uma matriz de bytes contendo o valor a ser configurado.

Retorna:
Cancelado

Exceções:

- XMSException

SetShort -Configurar Número Inteiro Curto

Interface:

```
void  SetShort(String name, Int16 value);
```

Configure um número inteiro curto no corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar o número inteiro curto no corpo da mensagem do mapa.

valor (entrada)
O número inteiro curto a ser configurado.

Retorna:
Cancelado

Exceções:

- XMSException

SetString -Configurar Sequência

Interface:

```
void  SetString(String name, String value);
```

Configure uma sequência no corpo da mensagem do mapa.

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome para identificar a sequência no corpo da mensagem do mapa.

valor (entrada)
Um objeto String encapsulando a sequência a ser configurada.

Retorna:
Cancelado

Exceções:

- XMSException

Propriedades e métodos herdados

As propriedades a seguir são herdadas da interface IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Os métodos a seguir são herdados da interface IMessage:

clearBody, clearProperties, PropertyExists

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IMessage

Um objeto de mensagem representa uma mensagem que um aplicativo envia ou recebe. IMessage é uma superclasse para as classes de mensagem, como IMapMessage.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
```

Para obter uma lista dos campos do cabeçalho da mensagem JMS em um objeto de Mensagem, consulte “Campos de cabeçalho na mensagem um XMS” na página 73 Para obter uma lista das propriedades definidas pelo JMS de um objeto de Mensagem, consulte “JMS-propriedades definidas de uma mensagem” na página 75 Para obter uma lista das propriedades definidas pelo IBM de um objeto de Mensagem, consulte “Propriedades Definidas pela IBM de uma Mensagem” na página 75 Para obter uma lista de propriedades JMS_IBM_MQMD* para o objeto Message, consulte “Propriedades JMS_IBM_MQMD*” na página 197

As mensagens são excluídas pelo coletor de lixo Quando uma mensagem é excluída, isso libera os recursos que ela estava usando.

Referências relacionadas
XMS
Este seçãocapítulo descreve a estrutura e o conteúdo de mensagens XMS e explica como os aplicativos processam mensagens XMS .

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
JMSCorrelationID	Obtenha e configure o identificador de correlação da mensagem como um objeto de Sequência.
JMSDeliveryMode	Obter e configurar o modo de entrega da mensagem.

propriedade .NET	Descrição
JMSDestination	Obtenha e configure o destino da mensagem
JMSExpiration	Obter e configurar o prazo de expiração da mensagem
JMSMessageID	Obtenha e configure o identificador de mensagem da mensagem como um objeto de sequência encapsulando o identificador de mensagens.
JMSPriority	Obtenha e configure a prioridade da mensagem
JMSRedelivered	Obter uma indicação se a mensagem está sendo entregue novamente e indicar se a mensagem está sendo entregue novamente.
JMSReplyTo	Obter e configurar o destino para o qual uma resposta à mensagem deve ser enviada.
JMSTimestamp	Obter e configurar o horário em que a mensagem foi enviada
JMSType	Get e configure o tipo da mensagem.
PropertyNames	Obter uma enumeração das propriedades de nomes da mensagem

GetJMSCorrelationID-Obter e Configurar JMSCorrelationID

Interface:

```
String JMSCorrelationID
{
  get;
  set;
}
```

Obtenha e configure o identificador de correlação da mensagem como um objeto de Sequência.

Exceções:

- XMSException

JMSDeliveryMode -Obter e configurar JMSDeliveryMode

Interface:

```
DeliveryMode JMSDeliveryMode
{
  get;
  set;
}
```

Obter e configurar o modo de entrega da mensagem.

O modo de entrega da mensagem é um dos seguintes valores:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

Para uma mensagem recém-criada que não foi enviada, o modo de entrega é `DeliveryMode.Persistente`, exceto para uma conexão em tempo real com um broker para o qual o modo de entrega é `DeliveryMode.NonPersistent` Para uma mensagem recebida, o método retorna o modo de entrega que foi configurado pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o modo de entrega configurando JMSDeliveryMode.

Exceções:

- XMSException

JMSDestination-Obter e Configurar JMSDestination.

Interface:

```
IDestination JMSDestination
{
  get;
  set;
}
```

Obtenha e configure o destino da mensagem

O destino é configurado pela chamada de IMessageProducer.send () quando a mensagem é enviada. O valor de JMSDestination é ignorado. No entanto, é possível usar JMSDestination para alterar o destino de uma mensagem recebida.

Para uma mensagem recém-criada que não foi enviada, o método retorna um objeto Destino nulo, a menos que o aplicativo de envio configure um destino configurando JMSDestination. Para uma mensagem recebida, o método retorna um objeto Destination para o destino que foi configurado pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o destino configurando JMSDestination.

Exceções:

- XMSException

JMSExpiration-Obter e configurar JMSExpiration

Interface:

```
Int64 JMSExpiration
{
  get;
  set;
}
```

Obter e configurar o prazo de expiração da mensagem

O prazo de expiração é configurado pela chamada de IMessageProducer.send () quando a mensagem é enviada. Seu valor é calculado adicionando o tempo de vida, conforme especificado pelo aplicativo de envio, ao tempo em que a mensagem é enviada. O prazo de expiração é expresso em milissegundos desde 00:00:00 GMT de 1 de janeiro de 1970.

Para uma mensagem recém-criada que não foi enviada, o prazo de expiração é 0, a menos que o aplicativo de envio configure um prazo de expiração diferente configurando JMSExpiration. Para uma mensagem recebida, o método retorna o prazo de expiração que foi configurado pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o prazo de expiração configurando JMSExpiration.

Se o tempo de vida for 0, a chamada IMessageProducer.send () configurará o tempo de vencimento como 0 para indicar que a mensagem não expira...

O XMS descarta mensagens expiradas e não as entrega para aplicativos.

Exceções:

- XMSException

JMSMessageID -Obter e configurar JMSMessageID

Interface:

```
String JMSMessageID
{
  get;
  set;
}
```

Obtenha e configure o identificador de mensagem da mensagem como um objeto de sequência encapsulando o identificador de mensagens.

O identificador de mensagem é configurado pela chamada .send () de IMessageProducerquando a mensagem é enviada. Para uma mensagem que foi recebida, o método retorna o identificador de mensagem que foi configurado pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o identificador de mensagem configurando JMSMessageID...

Se a mensagem não tiver identificador de mensagem, o método retornará um nulo.

Exceções:

- XMSException

JMSPriority-Obter e configurar JMSPriority.

Interface:

```
Int32 JMSPriority
{
  get;
  set;
}
```

Obtenha e configure a prioridade da mensagem

A prioridade é configurada pela chamada IMessageProducer.send () quando a mensagem é enviada O valor é um número inteiro no intervalo 0, a prioridade mais baixa, para 9, a prioridade mais alta.

Para uma mensagem recém-criada que não foi enviada, a prioridade será 4 a menos que o aplicativo de envio configure uma prioridade diferente configurando JMSPriority. Para uma mensagem recebida, o método retorna a prioridade que foi configurada pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere a prioridade configurando JMSPriority.

Exceções:

- XMSException

JMSRedelivered-Obter e Configurar JMSRedelivered

Interface:

```
Boolean JMSRedelivered
{
  get;
  set;
}
```

Obter uma indicação se a mensagem está sendo entregue novamente e indicar se a mensagem está sendo entregue novamente. A indicação é configurada pela chamada IMessageConsumer.receive () quando a mensagem é recebida.

Essa propriedade possui os seguintes valores:

- `True`, se a mensagem estiver sendo entregue novamente
- `False`, se a mensagem não estiver sendo entregue novamente

Para uma conexão em tempo real com um broker, o valor é sempre `False`

Uma indicação de nova entrega configurada por JMSRedelivered antes de a mensagem ser enviada é ignorada pela chamada IMessageProducer.send () quando a mensagem é enviada e é ignorada e substituída pela chamada IMessageConsumer.receive () quando a mensagem é recebida. No entanto, é possível usar JMSRedelivered para alterar a indicação de uma mensagem recebida.

Exceções:

- XMSException

JMSReplyTo -Get e Configurar JMSReplyTo

Interface:

```
IDestination JMSReplyTo
{
  get;
  set;
}
```

Obter e configurar o destino para o qual uma resposta à mensagem deve ser enviada.

O valor dessa propriedade é um objeto de Destino para o destino no qual uma resposta à mensagem deve ser enviada. Um objeto Destino nulo significa que nenhuma resposta é esperada.

Exceções:

- XMSException

JMSTimestamp-Obter e configurar JMSTimestamp.

Interface:

```
Int64 JMSTimestamp
{
  get;
  set;
}
```

Obter e configurar o horário em que a mensagem foi enviada

O registro de data e hora é configurado pela chamada IMessageProducer.send () quando a mensagem é enviada e é expressa em milissegundos desde 00:00:00 GMT de 1 de janeiro de 1970.

Para uma mensagem recém-criada que não foi enviada, o registro de data e hora será 0, a menos que o aplicativo de envio configure um registro de data e hora diferente configurando JMSTimestamp. Para uma mensagem que foi recebida, o método retorna o registro de data e hora que foi configurado pela chamada IMessageProducer.send () quando a mensagem foi enviada, a menos que o aplicativo de recebimento altere o registro de data e hora configurando JMSTimestamp.

Exceções:

- XMSException

Notes:

1. Se o registro de data e hora for indefinido, o método retornará 0, mas não lançará nenhuma exceção

JMSType-Obter e Configurar JMSType

Interface:

```
String JMSType
{
  get;
  set;
}
```

Get e configure o tipo da mensagem.

O valor de JMSType é uma sequência encapsulando o tipo da mensagem... Se a conversão de dados for necessária, esse valor será o tipo após a conversão..

Exceções:

- XMSException

PropertyNames -Obter Propriedades

Interface:

```
System.Collections.IEnumerator PropertyNames
{
  get;
}
```

Obter uma enumeração das propriedades de nomes da mensagem

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
Confirmação	Reconheça essa mensagem e todas as mensagens não reconhecidas anteriormente recebidas pela sessão.
ClearBody	Limpe o corpo da mensagem.
ClearProperties	limpa as propriedades da mensagem.
PropertyExists	Verifique se a mensagem possui uma propriedade com o nome especificado.

Reconhecimento-Reconhecimento

Interface:

```
void  Acknowledge();
```

Reconheça essa mensagem e todas as mensagens não reconhecidas anteriormente recebidas pela sessão.

Um aplicativo pode chamar esse método se o modo de confirmação da sessão for AcknowledgeMode.ClientAcknowledge As chamadas para o método serão ignoradas se a sessão tiver qualquer outro modo de confirmação ou for transacionada

As mensagens que foram recebidas, mas não reconhecidas, podem ser entregues novamente.

Para obter mais informações sobre o reconhecimento de mensagens, consulte “Confirmação da mensagem..” na página 28

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException
- Exceção IllegalState

ClearBody -Limpar corpo

Interface:

```
void  ClearBody();
```

Limpe o corpo da mensagem. Os campos de cabeçalho e as propriedades de mensagem não são limpos

Se um aplicativo limpar um corpo da mensagem, o corpo será deixado no mesmo estado que um corpo vazio em uma mensagem recém-criada O estado de um corpo vazio em uma mensagem recém-criada depende do tipo de corpo da mensagem.. Para obter mais informações, consulte “O corpo da mensagem um XMS” na página 77.

Um aplicativo pode limpar um corpo de mensagem a qualquer momento, não importa em qual estado o corpo está. Se um corpo da mensagem for somente leitura, a única maneira de um aplicativo poder gravar no corpo será para o aplicativo limpar o corpo primeiro.

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException

ClearProperties -Limpar Propriedades

Interface:

```
void  ClearProperties();
```

limpa as propriedades da mensagem. Os campos de cabeçalho e o corpo da mensagem não são limpos

Se um aplicativo limpar as propriedades de uma mensagem, as propriedades se tornarão legíveis e graváveis

Um aplicativo pode limpar as propriedades de uma mensagem a qualquer momento, independentemente do estado em que as propriedades estão. Se as propriedades de uma mensagem forem somente leitura, a única maneira de as propriedades se tornarem graváveis será para o aplicativo limpar as propriedades primeiro.

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException

PropertyExists -Verificar propriedade existe

Interface:

```
Boolean PropertyExists(String propertyName);
```

Verifique se a mensagem possui uma propriedade com o nome especificado.

Parâmetros:

propertyName (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:

- `True`, se a mensagem tiver uma propriedade com o nome especificado
- `False`, se a mensagem não tiver uma propriedade com o nome especificado

Exceções:

- XMSException

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IMessageConsumer

Um aplicativo usa um consumidor de mensagem para receber mensagens enviadas a um destino.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessageConsumer
```

Para obter uma lista das propriedades definidas do XMS de um objeto MessageConsumer , consulte “Propriedades de MessageConsumer” na página 201

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
MessageListener	Obter o listener de mensagens que está registrado com o consumidor de mensagem e registrar um listener de mensagens com o consumidor de mensagem.
MessageSelector	Obter o seletor de mensagem para o consumidor de mensagens

MessageListener -Obter e Configurar Listener de Mensagens

Interface:

```
MessageListener MessageListener
{
  get;
  set;
}
```

Obter o listener de mensagens que está registrado com o consumidor de mensagem e registrar um listener de mensagens com o consumidor de mensagem.

Se nenhum listener de mensagens estiver registrado com o consumidor de mensagens, MessageListener será nulo. Se um listener de mensagem já estiver registrado com o consumidor de mensagens, será possível cancelar o registro especificando um nulo.

Para obter mais informações sobre o uso de listeners de mensagens, consulte “Listeners de mensagem e de exceção em .NET” na página 50

Exceções:

- XMSException

MessageSelector -Obter Seletor de Mensagem

Interface:

```
String MessageSelector
{
  get;
}
```

Obter o seletor de mensagem para o consumidor de mensagens O valor de retorno é um objeto String encapsulando a expressão do seletor de mensagem. Se a conversão de dados for necessária, esse valor será a expressão do seletor de mensagem após a conversão.. Se o consumidor de mensagens não tiver um seletor de mensagem, o valor de MessageSelector será um objeto String nulo.

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
Fechar	Feche o consumidor de mensagens.
Receber	Receber a próxima mensagem para o consumidor de mensagens A chamada espera indefinidamente por uma mensagem ou até o consumidor de mensagem ser fechado.
Receber	Receber a próxima mensagem para o consumidor de mensagens A chamada aguarda apenas um período especificado para uma mensagem ou até que o consumidor de mensagens seja fechado.
ReceiveNoEspera	Receba a próxima mensagem para o consumidor de mensagens se uma estiver disponível imediatamente.

Fechar-Fechar Consumidor da Mensagem.

Interface:

```
void  Close();
```

Feche o consumidor de mensagens.

Se um aplicativo tentar fechar um consumidor de mensagens que já esteja encerrado, a chamada será ignorada

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException

Receber-Receber

Interface:

```
IMessage Receive();
```

Receber a próxima mensagem para o consumidor de mensagens A chamada espera indefinidamente por uma mensagem ou até o consumidor de mensagem ser fechado.

Parâmetros:
Nenhum

Retorna:
Um ponteiro para o objeto de Mensagem Se o consumidor de mensagens for fechado enquanto a chamada estiver aguardando uma mensagem, o método retornará um ponteiro para um objeto Mensagem nulo.

Exceções:

- XMSException

Recebimento-Recebimento (com um intervalo de espera)

Interface:

```
IMessage Receive(Int64 delay);
```

Receber a próxima mensagem para o consumidor de mensagens A chamada aguarda apenas um período especificado para uma mensagem ou até que o consumidor de mensagens seja fechado.

Parâmetros:

atraso (entrada)
O tempo, em milissegundos, que a chamada espera por uma mensagem Se você especificar um intervalo de espera de 0, a chamada aguardará indefinidamente uma mensagem.

Retorna:
Um ponteiro para o objeto de Mensagem Se nenhuma mensagem chegar durante o intervalo de espera ou se o consumidor de mensagens for fechado enquanto a chamada estiver aguardando uma mensagem, o método retornará um ponteiro para um objeto de Mensagem nulo, mas não emitirá nenhuma exceção.

Exceções:

- XMSException

ReceiveNoEspera-Receber com Nenhuma Espera

Interface:

```
IMessage ReceiveNoWait();
```

Receba a próxima mensagem para o consumidor de mensagens se uma estiver disponível imediatamente.

Parâmetros:
Nenhum

Retorna:
Um ponteiro para um objeto de Mensagem Se nenhuma mensagem estiver disponível imediatamente, o método retornará um ponteiro para um objeto Mensagem nulo.

Exceções:

- XMSException

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

MessageEOFException

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
            |
            +----IBM.XMS.MessageEOFException
```

XMS lança essa exceção se XMS encontrar o final de um fluxo de mensagens de bytes quando um aplicativo estiver lendo o corpo de uma mensagem de bytes.

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

MessageFormatExceção

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.MessageFormatException
```

XMS lançará essa exceção se o XMS encontrar uma mensagem com um formato que não seja válido

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

IMessageListener (delegado)

Hierarquia de herança:
Nenhum

Um aplicativo usa um listener de mensagens para receber mensagens assincronamente.

Delegar

Resumo dos métodos:

Método	Descrição
MessageListener	Entregar uma mensagem de forma assíncrona ao consumidor de mensagem.

MessageListener -Listener de Mensagens

Interface:

```
public delegate void MessageListener(IMessage msg);
```

Entregar uma mensagem de forma assíncrona ao consumidor de mensagem.

Os métodos que implementam esse delegado podem ser registrados com a conexão

Para obter mais informações sobre o uso de listeners de mensagens, consulte “Listeners de mensagem e de exceção em .NET” na página 50

Parâmetros:

mesg (entrada)
O objeto de Mensagem

Retorna:
Cancelado

MessageNotReadableException

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.MessageNotReadableException
```

XMS lança essa exceção se um aplicativo tentar ler o corpo de uma mensagem que é somente gravação.

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

MessageNotWritableException

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.MessageNotWritableException
```

XMS lança essa exceção se um aplicativo tentar gravar no corpo de uma mensagem que é somente leitura.

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

IMessageProducer

Um aplicativo usa um produtor de mensagem para enviar mensagens para um destino.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessageProducer
```

Para obter uma lista das propriedades definidas XMS de um objeto MessageProducer , consulte “Propriedades do MessageProducer” na página 201.

Propriedades .NET

Resumo de propriedades .NET :

Propriedades .NET	Descrição
DeliveryMode	Obter e configurar o modo de entrega padrão para mensagens enviadas pelo produtor de mensagens
Destino	Obter o destino para o produtor de mensagem
DisableMsgID	Obter uma indicação de se um aplicativo de recebimento requer que identificadores de mensagens sejam incluídos em mensagens enviadas pelo produtor de mensagens e indicar se um aplicativo de recebimento requer que identificadores de mensagens sejam incluídos em mensagens enviadas pelo produtor de mensagem.

Propriedades .NET	Descrição
DisableMsgTS	Obter uma indicação se um aplicativo de recebimento requer que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagens e indicar se um aplicativo de recebimento requer que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagem.
Prioridade	Obter e configurar a prioridade padrão para mensagens enviadas pelo produtor de mensagem
TimeToLive	Obter e configurar o período de tempo padrão que uma mensagem existe antes de expirar.

DeliveryMode -Obter e Configurar Modo de Entrega Padrão

Interface:

```
DeliveryMode DeliveryMode
{
  get;
  set;
}
```

Obter e configurar o modo de entrega padrão para mensagens enviadas pelo produtor de mensagens

O modo de entrega padrão possui um dos seguintes valores:

`DeliveryMode.Persistent`
`DeliveryMode.NonPersistent`

Para uma conexão em tempo real com um broker, o valor deve ser `DeliveryMode.NonPersistent`.

O valor padrão é `DeliveryMode.Persistent`, exceto para uma conexão em tempo real com um broker para o qual o valor padrão é `DeliveryMode.NonPersistent`.

Exceções:

- XMSException

Destino-Obter Destino

Interface:

```
IDestination Destination
{
  get;
}
```

Obter o destino para o produtor de mensagem

Parâmetros:
Nenhum

Retorna:
O objeto de Destino Se o produtor da mensagem não tiver um destino, o método retornará um objeto de Destino nulo...

Exceções:

- XMSException

ID de DisableMsg-Obter e Configurar Sinalizador de ID de Mensagem de Desabilitação

Interface:

```
Boolean DisableMessageID
{
  get;
```

```
  set;
}
```

Obter uma indicação de se um aplicativo de recebimento requer que identificadores de mensagens sejam incluídos em mensagens enviadas pelo produtor de mensagens e indicar se um aplicativo de recebimento requer que identificadores de mensagens sejam incluídos em mensagens enviadas pelo produtor de mensagem.

Em uma conexão com um gerenciador de fila ou em uma conexão em tempo real com um broker, esse sinalizador é ignorado. Em uma conexão com um barramento de integração de serviços, o sinalizador é honrado

O ID DisabledMsgpossui os valores a seguir:

- `True`, se um aplicativo de recebimento não requerer que identificadores de mensagem sejam incluídos em mensagens enviadas pelo produtor de mensagem
- `False`, se um aplicativo de recebimento exigir que identificadores de mensagem sejam incluídos em mensagens enviadas pelo produtor de mensagens

Exceções:

- XMSException

DisableMsgTS-Obter e Configurar Sinalizador de Registro de Data e Hora de Desativação

Interface:

```
Boolean DisableMessageTimestamp
{
  get;
  set;
}
```

Obter uma indicação se um aplicativo de recebimento requer que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagens e indicar se um aplicativo de recebimento requer que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagem.

Em uma conexão em tempo real com um broker, essa sinalização é ignorada Em uma conexão com um gerenciador de filas ou em uma conexão com um barramento de integração de serviços, a sinalização é honrada.

DisableMsgTS possui os seguintes valores:

- `True`, se um aplicativo de recebimento não precisar que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagem
- `False`, se um aplicativo de recebimento não precisar que registros de data e hora sejam incluídos em mensagens enviadas pelo produtor de mensagem

Retorna:

Exceções:

- XMSException

Prioridade-Obter e Configurar Prioridade Padrão

Interface:

```
Int32 Priority
{
  get;
  set;
}
```

Obter e configurar a prioridade padrão para mensagens enviadas pelo produtor de mensagem

O valor da prioridade da mensagem padrão é um número inteiro no intervalo `0`, a prioridade mais baixa, para 9, a prioridade mais alta.

Em uma conexão em tempo real com um broker, a prioridade de uma mensagem é ignorada

Exceções:

- XMSException

TimeToLive-Obter e configurar o tempo de vida padrão

Interface:

```
Int64 TimeToLive
{
  get;
  set;
}
```

Obter e configurar o período de tempo padrão que uma mensagem existe antes de expirar.

O tempo é medido a partir de quando o produtor da mensagem envia a mensagem e é o tempo padrão de vida em milissegundos Um valor 0 significa que uma mensagem nunca expira.

Para uma conexão em tempo real com um broker, este valor é sempre 0.

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
Fechar	Feche o produtor da mensagem.
Enviar	Enviar uma mensagem para o destino especificado quando o produtor da mensagem foi criado. Envie a mensagem usando o modo de entrega padrão, prioridade e tempo de vida do produtor da mensagem.
Enviar	Enviar uma mensagem para o destino especificado quando o produtor da mensagem foi criado. Envie a mensagem usando o modo de entrega, prioridade e tempo de vida especificados.
Enviar	Enviar uma mensagem para um destino especificado se estiver usando um produtor de mensagem para o qual nenhum destino foi especificado quando o produtor de mensagem foi criado. Envie a mensagem usando o modo de entrega padrão, prioridade e tempo de vida do produtor da mensagem.
Enviar	Enviar uma mensagem para um destino especificado se estiver usando um produtor de mensagem para o qual nenhum destino foi especificado quando o produtor de mensagem foi criado. Envie a mensagem usando o modo de entrega, prioridade e tempo de vida especificados.

Fechar-Fechar Produtor de Mensagem

Interface:

```
void  Close();
```

Feche o produtor da mensagem.

Se um aplicativo tentar fechar um produtor de mensagem que já está fechado, a chamada será ignorada

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException

Enviar-Enviar

Interface:

```
void Send(IMessage msg) ;
```

Enviar uma mensagem para o destino especificado quando o produtor da mensagem foi criado. Envie a mensagem usando o modo de entrega padrão, prioridade e tempo de vida do produtor da mensagem.

Parâmetros:

msg (entrada)
O objeto de Mensagem

Retorna:
Cancelado

Exceções:

- XMSException
- MessageFormatExceção
- Exceção de InvalidDestination

Enviar-Enviar (especificando um modo de entrega, prioridade e tempo de vida)

Interface:

```
void Send(IMessage msg,
          DeliveryMode deliveryMode,
          Int32 priority,
          Int64 timeToLive);
```

Enviar uma mensagem para o destino especificado quando o produtor da mensagem foi criado. Envie a mensagem usando o modo de entrega, prioridade e tempo de vida especificados.

Parâmetros:

msg (entrada)
O objeto de Mensagem

deliveryMode (entrada)
O modo de entrega para a mensagem, que deve ser um dos seguintes valores:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

Para uma conexão em tempo real com um broker, o valor deve ser `DeliveryMode.NonPersistent`.

prioridade (entrada)
A prioridade da mensagem. O valor pode ser um número inteiro no intervalo 0, para a prioridade mais baixo, para 9, para a prioridade mais alta. Em uma conexão em tempo real com um broker, o valor é ignorado

timeToAtivo (entrada)
O tempo de vida da mensagem em milissegundos. Um valor 0 significa que a mensagem nunca expira. Para uma conexão em tempo real com um broker, o valor deve ser 0

Retorna:
Cancelado

Exceções:

- XMSException

- MessageFormatExceção
- Exceção de InvalidDestination
- Exceção IllegalState

Enviar-Enviar (para um destino especificado)

Interface:

```
void Send(IDestination dest, IMessage msg) ;
```

Enviar uma mensagem para um destino especificado se estiver usando um produtor de mensagem para o qual nenhum destino foi especificado quando o produtor de mensagem foi criado. Envie a mensagem usando o modo de entrega padrão, prioridade e tempo de vida do produtor da mensagem.

Geralmente, você especifica um destino ao criar um produtor de mensagens, mas, se não o fizer, deverá especificar um destino toda vez que enviar uma mensagem.

Parâmetros:

dest (entrada)
O objeto de Destino

msg (entrada)
O objeto de Mensagem

Retorna:
Cancelado

Exceções:

- XMSException
- MessageFormatExceção
- Exceção de InvalidDestination

Enviar-Enviar (para um destino especificado, especificando um modo de entrega, prioridade e tempo de vida)

Interface:

```
void Send(IDestination dest,
          IMessage msg,
          DeliveryMode deliveryMode,
          Int32 priority,
          Int64 timeToLive) ;
```

Enviar uma mensagem para um destino especificado se estiver usando um produtor de mensagem para o qual nenhum destino foi especificado quando o produtor de mensagem foi criado. Envie a mensagem usando o modo de entrega, prioridade e tempo de vida especificados.

Geralmente, você especifica um destino ao criar um produtor de mensagens, mas, se não o fizer, deverá especificar um destino toda vez que enviar uma mensagem.

Parâmetros:

dest (entrada)
O objeto de Destino

msg (entrada)
O objeto de Mensagem

deliveryMode (entrada)
O modo de entrega para a mensagem, que deve ser um dos seguintes valores:

`DeliveryMode.Persistent`
`DeliveryMode.NonPersistent`

Para uma conexão em tempo real com um broker, o valor deve ser `DeliveryMode.NonPersistent`.

prioridade (entrada)
A prioridade da mensagem. O valor pode ser um número inteiro no intervalo 0, para a prioridade mais baixo, para 9, para a prioridade mais alta. Em uma conexão em tempo real com um broker, o valor é ignorado

timeToAtivo (entrada)
O tempo de vida da mensagem em milissegundos. Um valor 0 significa que a mensagem nunca expira. Para uma conexão em tempo real com um broker, o valor deve ser 0

Retorna:
Cancelado

Exceções:

- XMSException
- MessageFormatExceção
- Exceção de InvalidDestination
- Exceção IllegalState

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IObjectMessage

Uma mensagem de objeto é uma mensagem cujo corpo compreende um objeto Java ou .NET serializado.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.IObjectMessage
```

Referências relacionadas
Mensagens de objeto
O corpo de uma mensagem de objeto contém um objeto serializadoJava ou .NET..

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
Object	Get e configure o objeto que forma o corpo da mensagem do objeto.

Objeto-Obter e Configurar Objeto como Bytes.

Interface:

```
System.Object Object
    {
      get;
      set;
    }
```

```
Byte[] GetObject();
```

Get e configure o objeto que forma o corpo da mensagem do objeto.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException
- MessageNotWritableException

Propriedades e métodos herdados

As propriedades a seguir são herdadas da interface IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Os métodos a seguir são herdados da interface IMessage:

clearBody, clearProperties, PropertyExists

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IPropertyContext

IPropertyContext é uma superclasse abstrata que contém métodos que obtêm e configuram propriedades. Estes métodos são herdados por outras classes.

Hierarquia de herança:
Nenhum

Methods

Resumo dos métodos:

Método	Descrição
GetBooleanProperty	Obter o valor da propriedade booleana com o nome especificado
GetByteProperty	Obter o valor da propriedade de byte identificada por nome.
GetBytesProperty	Obtenha o valor da propriedade de matriz de bytes identificada por nome..
GetCharPropriedade	Obter o valor da propriedade de caractere de 2 bytes identificada por nome.
GetDoubleProperty	Obtenha o valor da propriedade de ponto flutuante de precisão dupla identificada por nome...
GetFloatProperty	Obter o valor da propriedade de ponto flutuante identificada por nome.
getIntProperty	Obter o valor da propriedade de número inteiro identificada pelo nome
GetLongProperty	Obter o valor da propriedade de número inteiro longo identificado por nome.
GetObjectProperty	Obtenha o valor e o tipo de dados da propriedade identificados por nome..

Método	Descrição
GetShortProperty	Obter o valor da propriedade de número inteiro curto identificada por nome.
GetStringProperty	Obter o valor da propriedade de sequência identificada por nome.
SetBooleanProperty	Configure o valor da propriedade booleana identificada por nome
SetByteProperty	Configure o valor da propriedade de byte identificada por nome..
SetBytesProperty	Configure o valor da propriedade de matriz de bytes identificada por nome..
SetCharPropriedade	Configure o valor da propriedade de caractere de 2 bytes identificada por nome..
SetDoubleProperty	Configure o valor da propriedade de ponto flutuante de precisão dupla identificado por nome..
SetFloatProperty	Configure o valor da propriedade de ponto flutuante identificada por nome..
SetIntProperty	Configure o valor da propriedade de número inteiro identificada por nome.
SetLongProperty	Configurar o valor da propriedade de número inteiro longo identificada pelo nome.
SetObjectProperty	Configure o valor e o tipo de dado de uma propriedade identificada por nome.
SetShortProperty	Configure o valor da propriedade de número inteiro curto identificada por nome..
SetStringProperty	Configure o valor da propriedade de cadeia identificada por nome..

Propriedade GetBoolean-Obter Propriedade Booleana

Interface:

```
Boolean GetBooleanProperty(String property_name);
```

Obter o valor da propriedade booleana com o nome especificado

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade.

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetByte-Obter Propriedade de Byte

Interface:

```
Byte    GetByteProperty(String property_name) ;
Int16   GetSignedByteProperty(String property_name) ;
```

Obter o valor da propriedade de byte identificada por nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade.

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetBytes-Propriedade da Matriz de Bytes de Obtenção

Interface:

```
Byte[]  GetBytesProperty(String property_name) ;
```

Obtenha o valor da propriedade de matriz de bytes identificada por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O número de bytes na matriz

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetChar-Propriedade de caractere de obtenção

Interface:

```
Char    GetCharProperty(String property_name) ;
```

Obter o valor da propriedade de caractere de 2 bytes identificada por nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade.

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetDouble-Obter Propriedade de Ponto Flutuante de Precisão Dupla

Interface:

```
Double  GetDoubleProperty(String property_name) ;
```

Obtenha o valor da propriedade de ponto flutuante de precisão dupla identificada por nome...

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade.

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetFloat-Obter propriedade de ponto flutuante

Interface:

```
Single  GetFloatProperty(String property_name) ;
```

Obter o valor da propriedade de ponto flutuante identificada por nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade.

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetInt-Propriedade GetInt

Interface:

```
Int32   GetIntProperty(String property_name) ;
```

Obter o valor da propriedade de número inteiro identificada pelo nome

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade.

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetLong-Obter propriedade de número inteiro longo

Interface:

```
Int64   GetLongProperty(String property_name) ;
```

Obter o valor da propriedade de número inteiro longo identificado por nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade.

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetObject-Obter Propriedade do Objeto

Interface:

```
Object  GetObjectProperty( String property_name) ;
```

Obtenha o valor e o tipo de dados da propriedade identificados por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade, que é um dos seguintes tipos de objeto:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetShort-Obter propriedade de número inteiro curto

Interface:

```
Int16   GetShortProperty(String property_name) ;
```

Obter o valor da propriedade de número inteiro curto identificada por nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
O valor da propriedade.

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade GetString-Propriedade GetString

Interface:

```
String  GetStringProperty(String property_name) ;
```

Obter o valor da propriedade de sequência identificada por nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

Retorna:
Um objeto String encapsulando a sequência que é o valor da propriedade. Se a conversão de dados for necessária, esse valor será a cadeia após a conversão..

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException

Propriedade SetBoolean-Configurar Propriedade Booleana

Interface:

```
void SetBooleanProperty( String property_name,  Boolean value) ;
```

Configure o valor da propriedade booleana identificada por nome

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetByte-Configurar propriedade de byte

Interface:

```
void SetByteProperty( String property_name,  Byte value) ;
      void SetSignedByteProperty( String property_name,  Int16 value) ;
```

Configure o valor da propriedade de byte identificada por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetBytes-Configurar Propriedade da Matriz de Bytes

Interface:

```
void SetBytesProperty( String property_name, Byte[] value ) ;
```

Configure o valor da propriedade de matriz de bytes identificada por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade, que é uma matriz de bytes.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetChar-Configurar Propriedade de Caractere

Interface:

```
void SetCharProperty( String property_name,  Char value) ;
```

Configure o valor da propriedade de caractere de 2 bytes identificada por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetDouble-Configurar Propriedade de Ponto Flutuante de Precisão Dupla

Interface:

```
void SetDoubleProperty( String property_name,  Double value) ;
```

Configure o valor da propriedade de ponto flutuante de precisão dupla identificado por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetFloat-Configurar propriedade de ponto flutuante

Interface:

```
void SetFloatProperty( String property_name,  Single value) ;
```

Configure o valor da propriedade de ponto flutuante identificada por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetInt-Configurar propriedade de número inteiro

Interface:

```
void SetIntProperty( String property_name,  Int32 value) ;
```

Configure o valor da propriedade de número inteiro identificada por nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetLong-Configurar propriedade de número inteiro longo

Interface:

```
void SetLongProperty( String property_name,  Int64 value) ;
```

Configurar o valor da propriedade de número inteiro longo identificada pelo nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetObject-Configurar Propriedade do objeto

Interface:

```
void SetObjectProperty( String property_name,  Object value) ;
```

Configure o valor e o tipo de dado de uma propriedade identificada por nome.

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

objectType (entrada)
O valor da propriedade, que deve ser um dos seguintes tipos de objeto:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

valor (entrada)
O valor da propriedade como uma matriz de bytes.

comprimento (entrada)
O número de bytes na matriz

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetShort-Configurar propriedade de número inteiro curto

Interface:

```
void SetShortProperty( String property_name,  Int16 value) ;
```

Configure o valor da propriedade de número inteiro curto identificada por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
O valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

Propriedade SetString-Configurar Propriedade de Cadeia

Interface:

```
void SetStringProperty( String property_name,  String value);
```

Configure o valor da propriedade de cadeia identificada por nome..

Parâmetros:

property_name (entrada)
Um objeto String encapsulando o nome da propriedade.

valor (entrada)
Um objeto String encapsulando a sequência que é o valor da propriedade.

Retorna:
Cancelado

Contexto do encadeamento:
Determinado pela subclasse

Exceções:

- XMSException
- MessageNotWritableException

IQueueBrowser

Um aplicativo usa um navegador de filas para pesquisar mensagens em uma fila sem removê-las.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
System.Collections.IEnumerable
   |
   +----IBM.XMS.IQueueBrowser
```

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
MessageSelector	Obtenha o seletor de mensagem do navegador de fila.
Fila	Obtenha a fila associada ao navegador de fila como um objeto de destino que representa a fila

MessageSelector -Obter Seletor de Mensagem

Interface:

```
String MessageSelector
{
  get;
}
```

Obtenha o seletor de mensagem do navegador de fila.

O seletor de mensagem é um objeto String que encapsula a expressão do seletor de mensagem. Se a conversão de dados for necessária, esse valor será a expressão do seletor de mensagem após a conversão.. Se o navegador da fila não tiver um seletor de mensagens, o método retornará um objeto String nulo.

Exceções:

- XMSException

Fila-Fila de obtenção

Interface:

```
IDestination Queue
{
  get;
}
```

Obtenha a fila associada ao navegador de fila como um objeto de destino que representa a fila

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
Fechar	Feche o navegador da filas.
GetEnumerator	Obter uma lista das mensagens na fila

Fechar-Fechar Navegador da Fila

Interface:

```
void  Close();
```

Feche o navegador da filas.

Se um aplicativo tentar fechar um browser de fila que já esteja fechado, a chamada será ignorada

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException

GetEnumerator -Obter mensagens

Interface:

```
IEnumerator GetEnumerator();
```

Obter uma lista das mensagens na fila

O método retorna um enumerador que encapsula uma lista de objetos de Mensagem. A ordem dos objetos de Mensagem é igual à ordem na qual as mensagens seriam recuperadas da fila. O aplicativo pode então usar o enumerador para pesquisar cada mensagem por vez.

O enumerador é atualizado dinamicamente conforme as mensagens são colocadas na fila e removidas da fila. Toda vez que o aplicativo chama IEnumerator.MoveNext () para procurar a próxima mensagem na fila, a mensagem reflete o conteúdo atual da fila.

Se um aplicativo chamar esse método mais de uma vez para um navegador de filas, cada chamada retornará um novo enumerador O aplicativo pode, portanto, usar mais de um enumerador para procurar as mensagens em uma fila e manter várias posições dentro da fila.

Parâmetros:
Nenhum

Retorna:
O objeto do Iterator (Iterator)

Exceções:

- XMSException

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

Solicitante

Um aplicativo usa um solicitante para enviar uma mensagem de solicitação e, em seguida, aguardar e receber a resposta..

Hierarquia de herança:
Nenhum

Construtores

Resumo de construtores:

Construtor	Descrição
Solicitante	Criar um solicitante.

Solicitante-Criar Solicitante

Interface:

```
Requestor(ISession sess, IDestination dest);
```

Criar um solicitante.

Parâmetros:

sess (entrada)
Um objeto de Sessão A sessão não deve ser transacionada e deve ter um dos seguintes modos de confirmação:

`AcknowledgeMode.AutoAcknowledge`
`AcknowledgeMode.DupsOkAcknowledge`

dest (entrada)
Um objeto de Destino representando o destino no qual o aplicativo pode enviar mensagens de solicitação.

Contexto do encadeamento:
A sessão associada ao solicitante

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
Fechar	Feche o solicitante.
Solicitação	Envie uma mensagem de solicitação e, em seguida, aguarde e receba uma resposta do aplicativo que recebe a mensagem de solicitação

Fechar-Fechar Solicitante

Interface:

```
void  Close();
```

Feche o solicitante.

Se um aplicativo tentar fechar um solicitante que já esteja fechado, a chamada será ignorada..

Nota: Quando um aplicativo fecha um solicitante, a sessão associada não é fechada também Nesse aspecto, XMS se comporta de forma diferente em comparação com JMS.

Parâmetros:
Nenhum

Retorna:
Cancelado

Contexto do encadeamento:
Qualquer

Exceções:

- XMSException

Solicitação-Resposta de solicitação

Interface:

```
IMessage Request(IMessage requestMessage);
```

Envie uma mensagem de solicitação e, em seguida, aguarde e receba uma resposta do aplicativo que recebe a mensagem de solicitação

Uma chamada para esse método é bloqueada até que uma resposta seja recebida ou até que a sessão termine, o que ocorrer primeiro.

Parâmetros:

requestMessage (entrada)
O objeto de Mensagem encapsulando a mensagem de solicitação

Retorna:
Um ponteiro para o objeto de Mensagem que contém a mensagem de resposta.

Contexto do encadeamento:
A sessão associada ao solicitante

Exceções:

- XMSException

Exceção de ResourceAllocation

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.ResourceAllocationException
```

XMS lança essa exceção se XMS não puder alocar os recursos necessários por um método.

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

SecurityException

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.SecurityException
```

O XMS lança essa exceção se o identificador de usuário e a senha fornecidos para autenticar um aplicativo forem rejeitados XMS também lança essa exceção se uma verificação de autoridade falhar e impedir que um método seja concluído.

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

ISession

Uma sessão é um único contexto encadeado para enviar e receber mensagens.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.ISession
```

Para obter uma lista das propriedades definidas XMS de um objeto Session, consulte “Propriedades da Sessão.” na página 201.

Propriedades .NET

Resumo de propriedades .NET:

Propriedade .NET	Descrição
AcknowledgeMode	Obter o modo de confirmação da sessão.
Realizado	Determine se a sessão foi transacionada

AcknowledgeMode -Obter Modo de Confirmação

Interface:

```
AcknowledgeMode AcknowledgeMode
    {
      get;
    }
```

Obter o modo de confirmação da sessão.

O modo de confirmação é especificado quando a sessão é criada

Desde que a sessão não seja transacionada, o modo de confirmação é um dos seguintes valores:

`AcknowledgeMode.AutoAcknowledge`
`AcknowledgeMode.ClientAcknowledge`
`AcknowledgeMode.DupsOkAcknowledge`

Para obter mais informações sobre os modos de confirmação, consulte “Confirmação da mensagem..” na página 28

Uma sessão transacionada não possui modo de confirmação. Se a sessão for transacionada, o método retornará `AcknowledgeMode.SessionTransacted` .

Exceções:

- XMSException

Transacionado-Determine se Transacionado

Interface:

```
Boolean Transacted
    {
      get;
    }
```

Determine se a sessão foi transacionada

O transacionado declarado é:

- `True`, se a sessão for transacionada..
- `False`, se a sessão não for transacionada..

Para uma conexão em tempo real com um broker, o método sempre retorna `False`..

Exceções:

- XMSException

Methods

Resumo dos métodos:

Método	Descrição
Fechar	Feche a sessão.
confirmar	Confirmar todas as mensagens processada na transação atual
CreateBrowser	Crie um navegador de filas da fila especificada.
CreateBrowser	Crie um navegador de fila para a fila especificada usando um seletor de mensagem
CreateBytesMensagem	Crie uma mensagem de bytes
CreateConsumer	Crie um consumidor de mensagens para o destino especificado
CreateConsumer	Criar um consumidor de mensagens para o destino especificado usando um seletor de mensagens
CreateConsumer	Crie um consumidor de mensagens para o destino especificado utilizando um seletor de mensagens e, se o destino for um tópico, especificando se o consumidor de mensagens recebe as mensagens publicadas pela sua própria conexão
CreateDurableAssinante	Crie um assinante durável para o tópico especificado
CreateDurableAssinante	Crie um assinante durável para o tópico especificado usando um seletor de mensagem e especificando se o assinante durável recebe as mensagens publicadas pela sua própria conexão
CreateMapMensagem	Crie uma mensagem de mapa.
CreateMessage	Crie uma mensagem que não tenha corpo.
CreateObjectMensagem	Criar uma mensagem de objeto
CreateProducer	Crie um produtor de mensagem para enviar mensagens ao destino especificado.
CreateQueue	Crie um objeto de Destino para representar uma fila no servidor de sistema de mensagens
CreateStreamMensagem	Criar uma mensagem de fluxo
CreateTemporary	Crie uma fila temporária.
TópicoCreateTemporary	Crie um tópico temporário
MensagemCreateText	Crie uma mensagem de texto com um corpo vazio
MensagemCreateText	Crie uma mensagem de texto cujo corpo seja inicializado com o texto especificado
CreateTopic	Crie um objeto de Destino para representar um tópico
Recuperar	Recuperar a sessão.
Retroceder	Retroceder todas as mensagens processadas na transação atual.
Cancelar assinatura	Excluir uma assinatura durável.

Fechar-Fechar Sessão

Interface:

```
void Close();
```

Feche a sessão. Se a sessão for transacionada, qualquer transação em andamento será retrocedida

Se um aplicativo tentar fechar uma sessão que já tenha sido fechada, a chamada será ignorada

Parâmetros:
Nenhum

Retorna:
Cancelado

Contexto do encadeamento:
Qualquer

Exceções:

- XMSException

Confirmação-Confirmar

Interface:

```
void Commit();
```

Confirmar todas as mensagens processada na transação atual

A sessão deve ser uma sessão transacionada

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException
- Exceção IllegalState
- TransactionRolledBackException

CreateBrowser -Criar Navegador da Fila

Interface:

```
IQueueBrowser CreateBrowser(IDestination queue) ;
```

Crie um navegador de filas da fila especificada.

Parâmetros:

fila (entrada)
Um objeto de Destino que representa a fila

Retorna:
O objeto QueueBrowser .

Exceções:

- XMSException
- Exceção de InvalidDestination

CreateBrowser -Create Queue Browser (com seletor de mensagens)

Interface:

```
IQueueBrowser CreateBrowser(IDestination queue, String selector) ;
```

Crie um navegador de fila para a fila especificada usando um seletor de mensagem

Parâmetros:

fila (entrada)
Um objeto de Destino que representa a fila

seletor (entrada)
Um objeto String encapsulando uma expressão do seletor de mensagem. Apenas as mensagens com propriedades que correspondem à expressão do seletor de mensagem são entregues para o navegador de filas

Um objeto String nulo significa que não há nenhum seletor de mensagens para o navegador de fila

Retorna:
O objeto QueueBrowser .

Exceções:

- XMSException
- Exceção de InvalidDestination
- InvalidSelectorExceção

Mensagem CreateBytes-Criar Mensagem de Bytes

Interface:

```
IBytesMessage CreateBytesMessage();
```

Crie uma mensagem de bytes

Parâmetros:
Nenhum

Retorna:
O objeto BytesMessage ..

Exceções:

- XMSException
- IllegalStateExceção (A sessão foi encerrada)

CreateConsumer -Criar Consumidor

Interface:

```
IMessageConsumer CreateConsumer(IDestination dest) ;
```

Crie um consumidor de mensagens para o destino especificado

Parâmetros:

dest (entrada)
O objeto de Destino

Retorna:
O objeto MessageConsumer ..

Exceções:

- XMSException
- Exceção de InvalidDestination

CreateConsumer -Criar Consumidor (com seletor de mensagens).

Interface:

```
IMessageConsumer CreateConsumer(IDestination dest,
                                String selector) ;
```

Criar um consumidor de mensagens para o destino especificado usando um seletor de mensagens

Parâmetros:

dest (entrada)
O objeto de Destino

seletor (entrada)
Um objeto String encapsulando uma expressão do seletor de mensagem. Apenas as mensagens com propriedades que correspondem à expressão do seletor de mensagem são entregues para o consumidor de mensagem

Um objeto String nulo significa que não há seletor de mensagem para o consumidor de mensagens.

Retorna:
O objeto MessageConsumer ..

Exceções:

- XMSException
- Exceção de InvalidDestination
- InvalidSelectorExceção

CreateConsumer -Criar Consumidor (com seletor de mensagens e sinalização de mensagem local)

Interface:

```
IMessageConsumer CreateConsumer(IDestination dest,
                                String selector,
                                Boolean noLocal) ;
```

Crie um consumidor de mensagens para o destino especificado utilizando um seletor de mensagens e, se o destino for um tópico, especificando se o consumidor de mensagens recebe as mensagens publicadas pela sua própria conexão

Parâmetros:

dest (entrada)
O objeto de Destino

seletor (entrada)
Um objeto String encapsulando uma expressão do seletor de mensagem. Apenas as mensagens com propriedades que correspondem à expressão do seletor de mensagem são entregues para o consumidor de mensagem

Um objeto String nulo significa que não há seletor de mensagem para o consumidor de mensagens.

noLocal (entrada)
O valor `True` significa que o consumidor de mensagens não recebe as mensagens publicadas por sua própria conexão O valor `False` significa que o consumidor de mensagens recebe as mensagens publicadas por sua conexão. O valor padrão é `False`.

Retorna:
O objeto MessageConsumer ..

Exceções:

- XMSException
- Exceção de InvalidDestination

- InvalidSelectorExceção

CreateDurableAssinante-Criar Assinante Durável

Interface:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,
                                         String subscription) ;
```

Crie um assinante durável para o tópico especificado

Este método não é válido para uma conexão em tempo real com um broker

Para obter mais informações sobre assinantes duráveis, consulte “Assinantes duráveis” na página 36..

Parâmetros:

dest (entrada)
Um objeto de Destino que representa o tópico O tópico não deve ser temporário.

assinatura (entrada)
Um objeto String encapsulando um nome que identifica a assinatura durável. O nome deve ser exclusivo no identificador de cliente para a conexão.

Retorna:
O objeto MessageConsumer representando o assinante durável.

Exceções:

- XMSException
- Exceção de InvalidDestination

CreateDurableAssinante-Criar Assinante Durável (com seletor de mensagem e sinalizador de mensagem local)

Interface:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,
                                         String subscription,
                                         String selector,
                                         Boolean noLocal) ;
```

Crie um assinante durável para o tópico especificado usando um seletor de mensagem e especificando se o assinante durável recebe as mensagens publicadas pela sua própria conexão

Este método não é válido para uma conexão em tempo real com um broker

Para obter mais informações sobre assinantes duráveis, consulte “Assinantes duráveis” na página 36..

Parâmetros:

dest (entrada)
Um objeto de Destino que representa o tópico O tópico não deve ser temporário.

assinatura (entrada)
Um objeto String encapsulando um nome que identifica a assinatura durável. O nome deve ser exclusivo no identificador de cliente para a conexão.

seletor (entrada)
Um objeto String encapsulando uma expressão do seletor de mensagem. Apenas as mensagens com propriedades que correspondem à expressão do seletor de mensagem são entregues para o assinante durável

Um objeto String nulo significa que não há seletor de mensagem para o assinante durável.

noLocal (entrada)
O valor `True` significa que o assinante durável não recebe as mensagens publicadas por sua própria conexão O valor `False` significa que o assinante durável recebe as mensagens publicadas por sua conexão. O valor padrão é `False`.

Retorna:
O objeto MessageConsumer representando o assinante durável.

Exceções:

- XMSException
- Exceção de InvalidDestination
- InvalidSelectorExceção

Mensagem CreateMap-Criar Mensagem de Mapa

Interface:

```
IMapMessage CreateMapMessage();
```

Crie uma mensagem de mapa.

Parâmetros:
Nenhum

Retorna:
O objeto MapMessage ..

Exceções:

- XMSException
- IllegalStateExceção (A sessão foi encerrada)

CreateMessage -Criar mensagem

Interface:

```
IMessage CreateMessage();
```

Crie uma mensagem que não tenha corpo.

Parâmetros:
Nenhum

Retorna:
O objeto de Mensagem

Exceções:

- XMSException
- IllegalStateExceção (A sessão foi encerrada)

Mensagem CreateObject-Criar Mensagem de Objeto

Interface:

```
IObjectMessage CreateObjectMessage();
```

Criar uma mensagem de objeto

Parâmetros:
Nenhum

Retorna:
O objeto ObjectMessage ..

Exceções:

- XMSException
- IllegalStateExceção (A sessão foi encerrada)

CreateProducer -Criar Produtor

Interface:

```
IMessageProducer CreateProducer(IDestination dest) ;
```

Crie um produtor de mensagem para enviar mensagens ao destino especificado.

Parâmetros:

dest (entrada)
O objeto de Destino

Se você especificar um objeto de Destino nulo, o produtor de mensagem será criado sem um destino Nesse caso, o aplicativo deve especificar um destino sempre que usar o produtor de mensagem para enviar uma mensagem.

Retorna:
O objeto MessageProducer .

Exceções:

- XMSException
- Exceção de InvalidDestination

CreateQueue -Criar Fila

Interface:

```
IDestination CreateQueue(String queue) ;
```

Crie um objeto de Destino para representar uma fila no servidor de sistema de mensagens

Este método não cria a fila no servidor de mensagens. Deve-se criar a fila antes que um aplicativo possa chamar esse método

Parâmetros:

fila (entrada)
Um objeto String encapsulando o nome da fila ou encapsulando um identificador uniforme de recursos (URI) que identifica a fila.

Retorna:
O objeto de Destino que representa a fila

Exceções:

- XMSException

Mensagem CreateStream-Criar Mensagem de Fluxo

Interface:

```
IStreamMessage CreateStreamMessage();
```

Criar uma mensagem de fluxo

Parâmetros:
Nenhum

Retorna:
O objeto StreamMessage ..

Exceções:

- XMSException
- XMS_ILLEGAL_STATE_EXCEPTION

Fila CreateTemporary-Criar Fila Temporária

Interface:

```
IDestination CreateTemporaryQueue() ;
```

Crie uma fila temporária.

O escopo da fila temporária é a conexão Apenas as sessões criadas pela conexão podem usar a fila temporária

A fila temporária permanece até que seja explicitamente excluída ou a conexão termine, o que for anterior.

Para obter mais informações sobre as filas temporárias, consulte “Destinos Temporários” na página 34

Parâmetros:
Nenhum

Retorna:
O objeto de Destino que representa a fila temporária

Exceções:

- XMSException

Tópico CreateTemporary-Criar tópico temporário

Interface:

```
IDestination CreateTemporaryTopic() ;
```

Crie um tópico temporário

O escopo do tópico temporário é a conexão. Apenas as sessões criadas pela conexão podem usar o tópico temporário

O tópico temporário permanece até que seja excluído explicitamente ou a conexão seja encerrada, o que ocorrer primeiro.

Para obter mais informações sobre tópicos temporários, consulte “Destinos Temporários” na página 34..

Parâmetros:
Nenhum

Retorna:
O objeto de Destino que representa o tópico temporário

Exceções:

- XMSException

Mensagem CreateText-Criar Mensagem de Texto

Interface:

```
ITextMessage CreateTextMessage();
```

Crie uma mensagem de texto com um corpo vazio

Parâmetros:
Nenhum

Retorna:
O objeto TextMessage ..

Exceções:

- XMSException

Mensagem CreateText-Criar Mensagem de Texto (inicializada)

Interface:

```
ITextMessage CreateTextMessage(String initialValue);
```

Crie uma mensagem de texto cujo corpo seja inicializado com o texto especificado

Parâmetros:

initialValue (entrada)
Um objeto String encapsulando o texto para inicializar o corpo da mensagem de texto.

Nenhum

Retorna:
O objeto TextMessage ..

Exceções:

- XMSException

CreateTopic -Criar Tópico

Interface:

```
IDestination CreateTopic(String topic) ;
```

Crie um objeto de Destino para representar um tópico

Parâmetros:

tópico (entrada)
Um objeto String encapsulando o nome do tópico ou encapsulando um identificador uniforme de recursos (URI) que identifica o tópico.

Retorna:
O objeto de Destino que representa o tópico

Exceções:

- XMSException

Recuperar-Recuperar

Interface:

```
void Recover();
```

Recuperar a sessão. A entrega de mensagem é interrompida e, em seguida, reiniciada com a mensagem não reconhecida mais antiga

A sessão não deve ser uma sessão transacionada

Para obter mais informações sobre a recuperação de uma sessão, consulte “Confirmação da mensagem..” na página 28

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException
- Exceção IllegalState

Retrocesso-Retrocesso

Interface:

```
void Rollback();
```

Retroceder todas as mensagens processadas na transação atual.

A sessão deve ser uma sessão transacionada

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException
- Exceção IllegalState

Cancelar assinatura-Cancelar assinatura

Interface:

```
void Unsubscribe(String subscription);
```

Excluir uma assinatura durável. O servidor de mensagens exclui o registro da assinatura durável que ele está mantendo e não envia mais mensagens para o assinante durável.

Um aplicativo não pode excluir uma assinatura durável em qualquer uma das seguintes circunstâncias:

- Enquanto há um consumidor de mensagens ativo para a assinatura durável
- Enquanto uma mensagem consumida faz parte de uma transação pendente
- Enquanto uma mensagem consumida não foi reconhecida

Este método não é válido para uma conexão em tempo real com um broker

Parâmetros:

assinatura (entrada)
Um objeto String encapsulando o nome que identifica a assinatura durável.

Retorna:
Cancelado

Exceções:

- XMSException
- Exceção de InvalidDestination
- Exceção IllegalState

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IStreamMessage

Uma mensagem de fluxo é uma mensagem cujo corpo compreende um fluxo de valores, em que cada valor possui um tipo de dados associado. Os conteúdos do corpo são gravados e lidos sequencialmente.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.IStreamMessage
```

Quando um aplicativo lê um valor do fluxo de mensagens, o valor pode ser convertido pelo XMS em outro tipo de dados. Para obter mais informações sobre essa forma de conversão implícita, consulte “Mensagens de Fluxo” na página 82

Referências relacionadas

Mensagens de Fluxo
O corpo de uma mensagem de fluxo contém um fluxo de valores, em que cada valor possui um tipo de dados associado.

Methods

Resumo dos métodos:

Método	Descrição
ReadBoolean	Leia um valor booleano do fluxo de mensagens.
ReadByte	Leia um número inteiro de 8 bits assinado do fluxo de mensagens.
ReadBytes	Ler uma matriz de bytes do fluxo de mensagens.
ReadChar	Leia um caractere de 2 bytes do fluxo de mensagens.
ReadDouble	Leia um número de ponto flutuante de precisão dupla de 8 bytes a partir do fluxo de mensagens
ReadFloat	Leia um número de ponto flutuante de 4 bytes a partir do fluxo de mensagens
ReadInt	Leia um número inteiro assinado de 32 bits do fluxo de mensagens.
ReadLong	Leia um número inteiro assinado de 64 bits a partir do fluxo de mensagens.
ReadObject	Leia um valor do fluxo de mensagens e retorne seu tipo de dados.
ReadShort	Leia um número inteiro de 16 bits assinado a partir do fluxo de mensagens.
ReadString	Leia uma cadeia do fluxo de mensagens.
Reconfigurar	Coloque o corpo da mensagem no modo somente leitura e posicione o cursor no início do fluxo de mensagens.
WriteBoolean	Grave um valor booleano no fluxo de mensagens.
WriteByte	Grave um byte no fluxo de mensagens
WriteBytes	Gravar uma matriz de bytes no fluxo de mensagens.
WriteChar	Grave um caractere no fluxo de mensagens como 2 bytes, primeiro byte de alta ordem.
WriteDouble	Converta um número de vírgula flutuante de precisão dupla em um número inteiro longo e grave o número inteiro longo no fluxo de mensagem como 8 bytes, primeiro byte de alta ordem...
WriteFloat	Converta um número de ponto flutuante em um número inteiro e escreva o número inteiro no fluxo de mensagens como 4 bytes, primeiro byte de alta ordem.
WriteInt	Escreva um número inteiro no fluxo de mensagem como 4 bytes, primeiro byte de alta ordem.

Método	Descrição
WriteLong	Grave um número inteiro longo no fluxo de mensagens como 8 bytes, primeiro byte de alta ordem.
WriteObject	Grave um valor, com um tipo de dado especificado, no fluxo de mensagens.
WriteShort	Grave um número inteiro curto no fluxo de mensagens como 2 bytes, primeiro byte de alta ordem.
WriteString	Grave uma sequência no fluxo de mensagens.

ReadBoolean -Valor Booleano de leitura

Interface:

```
Boolean ReadBoolean();
```

Leia um valor booleano do fluxo de mensagens.

Parâmetros:
Nenhum

Retorna:
O valor booleano que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadByte -Byte de Leitura

Interface:

```
Int16   ReadSignedByte();
    Byte    ReadByte();
```

Leia um número inteiro de 8 bits assinado do fluxo de mensagens.

Parâmetros:
Nenhum

Retorna:
O byte lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadBytes -Bytes de leitura

Interface:

```
Int32   ReadBytes(Byte[] array);
```

Ler uma matriz de bytes do fluxo de mensagens.

Parâmetros:

matriz (entrada)
O buffer contendo a matriz de bytes que é lida e o comprimento do buffer em bytes.

Se o número de bytes na matriz for menor ou igual ao comprimento do buffer, a matriz inteira será lida no buffer. Se o número de bytes na matriz for maior que o comprimento do buffer, o buffer será preenchido com parte da matriz e um cursor interno marcará a posição do byte seguinte a ser lido. Uma chamada subsequente para readBytes() lê bytes a partir da matriz iniciando na posição atual do cursor.

Se você especificar um ponteiro nulo na entrada, a chamada ignorará a matriz de bytes sem lê-la.

Retorna:
O número de bytes lidos no buffer. Se o buffer for parcialmente preenchido, o valor será menor que o comprimento do buffer, indicando que não há mais bytes na matriz restantes a serem lidos. Se não houver bytes restantes a serem lidos na matriz antes da chamada, o valor será `XMSC_END_OF_BYTEARRAY`.

Se você especificar um ponteiro nulo na entrada, o método não retornará valor.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadChar -Caractere de Leitura

Interface:

```
Char    ReadChar();
```

Leia um caractere de 2 bytes do fluxo de mensagens.

Parâmetros:
Nenhum

Retorna:
O caractere lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadDouble -Ler Número de Ponto Flutuante de Precisão Dupla

Interface:

```
Double  ReadDouble();
```

Leia um número de ponto flutuante de precisão dupla de 8 bytes a partir do fluxo de mensagens

Parâmetros:
Nenhum

Retorna:
O número do ponto flutuante de precisão dupla que é lido

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadFloat -Ler número de ponto flutuante

Interface:

```
Single  ReadFloat();
```

Leia um número de ponto flutuante de 4 bytes a partir do fluxo de mensagens

Parâmetros:
Nenhum

Retorna:
O número de ponto flutuante que é lido

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadInt -Ler Número Inteiro

Interface:

```
Int32   ReadInt();
```

Leia um número inteiro assinado de 32 bits do fluxo de mensagens.

Parâmetros:
Nenhum

Retorna:
O número inteiro que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadLong -Número inteiro longo de leitura

Interface:

```
Int64   ReadLong();
```

Leia um número inteiro assinado de 64 bits a partir do fluxo de mensagens.

Parâmetros:
Nenhum

Retorna:
O número inteiro longo que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadObject -Ler Objeto

Interface:

```
Object  ReadObject();
```

Leia um valor do fluxo de mensagens e retorne seu tipo de dados.

Parâmetros:
Nenhum

Retorna:
O valor, que é um dos seguintes tipos de objeto:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

Exceções:
XMSException

ReadShort -Número Inteiro Curto de Leitura

Interface:

```
Int16   ReadShort();
```

Leia um número inteiro de 16 bits assinado a partir do fluxo de mensagens.

Parâmetros:
Nenhum

Retorna:
O número inteiro curto que é lido.

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadString -Sequência de Leitura

Interface:

```
String  ReadString();
```

Leia uma cadeia do fluxo de mensagens. Se necessário, XMS converte os caracteres na sequência na página de código local.

Parâmetros:
Nenhum

Retorna:
Um objeto String encapsulando a sequência que é lida. Se a conversão de dados for necessária, esta será a sequência após a conversão

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

Reconfigurar-Reconfigurar

Interface:

```
void  Reset();
```

Coloque o corpo da mensagem no modo somente leitura e posicione o cursor no início do fluxo de mensagens.

Parâmetros:
Nenhum

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotReadableException
- MessageEOFException

WriteBoolean -Gravar Valor Booleano

Interface:

```
void  WriteBoolean(Boolean value);
```

Grave um valor booleano no fluxo de mensagens.

Parâmetros:

valor (entrada)
O valor booleano a ser gravado.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteByte -Byte de gravação

Interface:

```
void  WriteByte(Byte value);
void  WriteSignedByte(Int16 value);
```

Grave um byte no fluxo de mensagens

Parâmetros:

valor (entrada)
O byte a ser gravado..

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteBytes -Bytes de Gravação

Interface:

```
void  WriteBytes(Byte[] value);
```

Gravar uma matriz de bytes no fluxo de mensagens.

Parâmetros:

valor (entrada)
A matriz de bytes a ser gravada

comprimento (entrada)
O número de bytes na matriz

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteChar -Caractere de gravação

Interface:

```
void  WriteChar(Char value);
```

Grave um caractere no fluxo de mensagens como 2 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O caractere a ser gravado

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteDouble -Número do ponto flutuante de precisão dupla de gravação

Interface:

```
void  WriteDouble(Double value);
```

Converta um número de vírgula flutuante de precisão dupla em um número inteiro longo e grave o número inteiro longo no fluxo de mensagem como 8 bytes, primeiro byte de alta ordem...

Parâmetros:

valor (entrada)
O número de ponto flutuante de precisão dupla a ser gravado

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteFloat -Número do ponto flutuante de gravação

Interface:

```
void  WriteFloat(Single value);
```

Converta um número de ponto flutuante em um número inteiro e escreva o número inteiro no fluxo de mensagens como 4 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O número de ponto flutuante a ser gravado

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteInt -Número inteiro de gravação

Interface:

```
void  WriteInt(Int32 value);
```

Escreva um número inteiro no fluxo de mensagem como 4 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O número inteiro a ser gravado.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteLong -Número inteiro longo de gravação

Interface:

```
void  WriteLong(Int64 value);
```

Grave um número inteiro longo no fluxo de mensagens como 8 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O número inteiro longo a ser gravado.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteObject -Gravar Objeto

Interface:

```
void  WriteObject(Object value);
```

Grave um valor, com um tipo de dado especificado, no fluxo de mensagens.

Parâmetros:

objectType (entrada)
O valor, que deve ser um dos seguintes tipos de objeto:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

valor (entrada)
Uma matriz de bytes contendo o valor a ser gravado.

comprimento (entrada)
O número de bytes na matriz

Retorna:
Cancelado

Exceções:

- XMSException

WriteShort -Escrever Número Inteiro Curto

Interface:

```
void  WriteShort(Int16 value);
```

Grave um número inteiro curto no fluxo de mensagens como 2 bytes, primeiro byte de alta ordem.

Parâmetros:

valor (entrada)
O número inteiro curto a ser gravado.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

WriteString -Sequência de Gravação

Interface:

```
void  WriteString(String value);
```

Grave uma sequência no fluxo de mensagens.

Parâmetros:

valor (entrada)
Um objeto String encapsulando a sequência a ser gravada.

Retorna:
Cancelado

Exceções:

- XMSException
- MessageNotWritableException

Propriedades e métodos herdados

As propriedades a seguir são herdadas da interface IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Os métodos a seguir são herdados da interface IMessage:

clearBody, clearProperties, PropertyExists

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

ITextMessage

Uma mensagem de texto é uma mensagem cujo corpo compreende uma sequência.

Hierarquia de herança:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.ITextMessage
```

Referências relacionadas

Mensagens de texto
O corpo de uma mensagem de texto contém uma sequência.

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
texto	Get e configure a sequência que forma o corpo da mensagem de texto.

Texto-Obter e Configurar Texto

Interface:

```
String Text
{
  get;
  set;
}
```

Get e configure a sequência que forma o corpo da mensagem de texto.

Se necessário, XMS converte os caracteres na sequência na página de código local.

Exceções:

- XMSException
- MessageNotReadableException
- MessageNotWritableException

- MessageEOFException

Propriedades e métodos herdados

As propriedades a seguir são herdadas da interface IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Os métodos a seguir são herdados da interface IMessage:

clearBody, clearProperties, PropertyExists

Os métodos a seguir são herdados da interface IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

TransactionInProgressException

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.TransactionInProgressException
```

XMS lançará essa exceção se um aplicativo solicitar uma operação que não seja válida porque uma transação está em andamento

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

TransactionRolledBackException

Hierarquia de herança:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.TransactionRolledBackException
```

XMS lança esta exceção se um aplicativo chamar Session.commit() para confirmar a transação atual, mas a transação será, então, revertida.

Propriedades e métodos herdados

Os métodos a seguir são herdados da interface XMSException:

GetErrorCode, GetLinkedException

XMSException

Se XMS detectar um erro ao processar uma chamada para um método .NET , XMS lançará uma exceção. Uma exceção é um objeto que contém informações sobre o erro.

Hierarquia de herança:

```
System.Exception
   |
   +----IBM.XMS.XMSException
```

Há diferentes tipos de exceção XMS , e um objeto XMSException é apenas um tipo de excepção. Entretanto, a classe XMSException é uma superclasse das outras classes de exceção XMS . XMS lança um objeto XMSException em situações em que nenhum dos outros tipos de exceção é apropriado.

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
ErrorCode	Obter o código de erro
LinkedException	Obter a próxima exceção na cadeia de exceção.

ErrorCode -Obter Código de Erro

Interface:

```
public String ErrorCode
{
  get {return errorCode_;}
}
```

Obter o código de erro

Exceções:

- XMSException

LinkedException -Obter Exceção Vinculada

Interface:

```
public Exception LinkedException
{
  get { return linkedException_;}
  set { linkedException_ = value;}
}
```

Obter a próxima exceção na cadeia de exceção.

O método retornará um nulo se não houver mais exceções na cadeia.

Exceções:

- XMSException

XMSFactoryFactory

Se um aplicativo não estiver usando objetos administrados, utilize essa classe para criar connection factories, filas e tópicos..

Hierarquia de herança:
Nenhum

Propriedades .NET

Resumo de propriedades .NET :

propriedade .NET	Descrição
MetaData	Obtenha os metadados apropriados para o tipo de conexão do objeto XMSFactoryFactory .

Metadados-Recuperar metadados

Interface:

```
IConnectionMetaData MetaData
```

Obtenha os metadados apropriados para o tipo de conexão do objeto XMSFactoryFactory .

Exceções:
Nenhum

Methods

Resumo dos métodos:

Método	Descrição
CreateConnectionFactory	Crie um objeto ConnectionFactory do tipo declarado.
CreateQueue	Crie um objeto de Destino para representar uma fila no servidor de sistema de mensagens
CreateTopic	Crie um objeto de Destino para representar um tópico
GetInstance	Criar uma instância de XMSFactoryFactory. Um aplicativo XMS usa um objeto XMSFactoryFactory para obter uma referência a um objeto ConnectionFactory apropriado para o tipo de protocolo necessário. Esse objeto ConnectionFactory pode então produzir conexões somente para esse tipo de protocolo.

CreateConnectionFactory-Criar Connection Factory

Interface:

```
IConnectionFactory CreateConnectionFactory();
```

Crie um objeto ConnectionFactory do tipo declarado.

Parâmetros:
Nenhum

Retorna:
O objeto ConnectionFactory .

Exceções:

- XMSException

CreateQueue -Criar Fila

Interface:

```
IDestination CreateQueue(String name);
```

Crie um objeto de Destino para representar uma fila no servidor de sistema de mensagens

Este método não cria a fila no servidor de mensagens. Deve-se criar a fila antes que um aplicativo possa chamar esse método

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome da fila ou encapsulando um identificador uniforme de recursos (URI) que identifica a fila.

Retorna:
O objeto de Destino que representa a fila

Exceções:

- XMSException

CreateTopic -Criar Tópico

Interface:

```
IDestination CreateTopic(String name);
```

Crie um objeto de Destino para representar um tópico

Parâmetros:

nome (entrada)
Um objeto String encapsulando o nome do tópico ou encapsulando um identificador uniforme de recursos (URI) que identifica o tópico.

Retorna:
O objeto de Destino que representa o tópico

Exceções:

- XMSException

GetInstance -Obtenha uma instância de XMSFactoryFactory

Interface:

```
static XMSFactoryFactory GetInstance(int connectionType);
```

Criar uma instância de XMSFactoryFactory. Um aplicativo XMS usa um objeto XMSFactoryFactory para obter uma referência a um objeto ConnectionFactory apropriado para o tipo de protocolo necessário. Esse objeto ConnectionFactory pode então produzir conexões somente para esse tipo de protocolo.

Parâmetros:

connectionType (entrada)
O tipo de conexão para o qual o objeto ConnectionFactory produz conexões:

- XMSC.CT_WPM
- XMSC.CT_RTT
- XMSC.CT_WMQ

Retorna:
O objeto XMSFactoryFactory dedicado ao tipo de conexão declarado.

Exceções:

- Exceção NotSupported

Propriedades de objetos XMS

Isso seçãocapítulo documenta as propriedades de objeto definidas por XMS

O seçãocapítulo contém o seguinte tópicosseções:

- “Propriedades da Conexão” na página 184
- “Propriedades de ConnectionFactory” na página 185
- “Propriedades de Dados ConnectionMeta” na página 192
- “Propriedades de Destino” na página 192
- “Propriedades de InitialContext” na página 195
- “Propriedades de Mensagem” na página 196
- “Propriedades de MessageConsumer” na página 201
- “Propriedades do MessageProducer” na página 201
- “Propriedades da Sessão.” na página 201

Cada tópicoseção lista as propriedades de um objeto do tipo especificado e fornece uma descrição simples de cada propriedade..

Esse seçãocapítulo também contém o “Definições de propriedades” na página 201 tópicoseção, que fornece uma definição de cada propriedade.

Se um aplicativo definir suas próprias propriedades dos objetos descritos neste seçãocapítulo, ele não causará um erro, mas poderá causar resultados imprevisíveis

Nota: Os nomes e valores da propriedade nesta seção são mostrados no formulário XMSC.*NAME*, que é o formulário usado para C e C + +. No entanto, no .NET, o formulário do nome da propriedade pode ser XMSC.*NAME* ou XMSC_*NAME*, dependendo de como você está usando:

- Se você estiver especificando uma propriedade, o nome da propriedade deverá estar no formato XMSC.*NAME* conforme mostrado no exemplo a seguir:

  ```
  cf.SetStringProperty(XMSC.WMQ_CHANNEL, "DOTNET.SVRCONN");
  ```

- Se você estiver especificando uma cadeia, o nome da propriedade deverá estar no formato XMSC_*NAME* , conforme mostrado no exemplo a seguir:

  ```
  cf.SetStringProperty("XMSC_WMQ_CHANNEL", "DOTNET.SVRCONN");
  ```

  No .NET, os nomes e valores de propriedades são fornecidos como constantes na classe XMSC Essas constantes identificam sequências e seriam usadas por qualquer aplicativo do XMS .NET . Se você estiver usando essas constantes predefinidas, os nomes e valores de propriedades estarão no formato XMSC.*NAME*, portanto, por exemplo, você usaria XMSC.USERID, em vez de XMSC_USERID.

Os tipos de dados também estão no formato usado para C/C + +. É possível localizar os valores correspondentes para .NET em“Tipos de dados para .NET” na página 46

Conceitos relacionados

Construindo seus próprios aplicativos
Você constrói seus próprios aplicativos, como você constrói os aplicativos de amostra.

Referências relacionadas

Interfaces do .NET
Este tópicoseção documenta as interface de classe .NET e suas propriedades e métodos.

Propriedades da Conexão

Uma visão geral das propriedades do objeto de Conexão, com links para informações de referência mais detalhadas

Tabela 28. Propriedades da Conexão

Nome da propriedade	Descrição
“XMSC_WMQ_RESOLVED_QUEUE_MANAGER” na página 238	Essa propriedade é usada para obter o nome do gerenciador de filas ao qual ele está conectado

Tabela 28. Propriedades da Conexão (continuação)	
Nome da propriedade	**Descrição**
“XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID” na página 238	Essa propriedade é preenchida com o ID do Gerenciador de Filas após a conexão.
XMSC_WPM_CONNECTION_PROTOCOL	O protocolo de comunicações usado para a conexão com o mecanismo do sistema de mensagens. Essa propriedade é somente leitura.
XMSC_WPM_HOST_NAME	O nome do host ou o endereço IP do sistema que contém o mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Essa propriedade é somente leitura.
XMSC_WPM_ME_NAME	O nome do mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Essa propriedade é somente leitura.
XMSC_WPM_PORT	O número da porta atendida pelo mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Essa propriedade é somente leitura.

Um objeto Connection também possui propriedades somente leitura que são derivadas das propriedades do connection factory que foi usado para criar a conexão Essas propriedades são derivadas não apenas das propriedades do connection factory que foram definidas no momento em que a conexão foi criada, mas também dos valores padrão das propriedades não configuradas. As propriedades incluem apenas aquelas relevantes para o tipo de servidor de sistema de mensagens ao qual o aplicativo está conectado. Os nomes das propriedades são iguais aos nomes das propriedades do connection factory.

Propriedades de ConnectionFactory

Uma visão geral das propriedades do objeto ConnectionFactory , com links para informações de referência mais detalhadas.

Tabela 29. Propriedades de ConnectionFactory	
Nome da propriedade	**Descrição**
“XMSC_ASYNC_EXCEPÇÕES” na página 214	Essa propriedade determina se o XMS informa um ExceptionListener apenas quando uma conexão é quebrada ou quando qualquer exceção ocorre de forma assíncrona para uma chamada da API do XMS. Essa propriedade se aplica a todas as Conexões criadas por meio dessa ConnectionFactory que possui um ExceptionListener registrado.
XMSC_CLIENT_ID	O identificador do cliente para uma conexão.
XMSC_CONNECTION_TYPE	O tipo de servidor de mensagens para o qual um aplicativo se conecta.
XMSC_PASSWORD	Uma senha que pode ser usada para autenticar o aplicativo quando ele tenta se conectar a um servidor de mensagens.
“XMSC_RTT_BROKER_PING_INTERVAL” na página 219	O intervalo de tempo, em milissegundos, após o qual o XMS .NET verifica a conexão com um servidor do sistema de mensagens Real Time para detectar qualquer atividade.
XMSC_RTT_CONNECTION_PROTOCOL	O protocolo de comunicações usado para uma conexão em tempo real com um broker.

Tabela 29. Propriedades de ConnectionFactory (continuação)	
Nome da propriedade	**Descrição**
XMSC_RTT_HOST_NAME	O nome do host ou o endereço IP do sistema no qual um broker é executado.
XMSC_RTT_LOCAL_ADDRESS	O nome do host ou o endereço IP da interface de rede local a ser usada para uma conexão em tempo real com um broker.
XMSC_RTT_MULTICAST	A configuração multicast para um connection factory ou destino.
XMSC_RTT_PORT	O número da porta na qual um broker atende às solicitações recebidas.
XMSC_USERID	Um identificador de usuário que pode ser usado para autenticar o aplicativo quando ele tenta se conectar a um servidor de mensagens.
XMSC_WMQ_BROKER_CONTROLQ	O nome da fila de controle usada por um broker. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.
XMSC_WMQ_BROKER_PUBQ	O nome da fila monitorada por um broker na qual os aplicativos enviam mensagens que publicam. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.
XMSC_WMQ_BROKER_QMGR	O nome do gerenciador de filas ao qual um broker está conectado. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.

Tabela 29. Propriedades de ConnectionFactory (continuação)	
Nome da propriedade	**Descrição**
XMSC_WMQ_BROKER_SUBQ	O nome da fila de assinantes para um consumidor de mensagens não durável. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.
XMSC_WMQ_BROKER_VERSION	O tipo de corretor usado pelo aplicativo para uma conexão ou para o destino. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.
“XMSC_WMQ_CCDTURL” na página 224	Um Localizador Uniforme de Recursos (URL) que identifica o nome e o local do arquivo que contém a tabela de definição de canal do cliente e também especifica como o arquivo pode ser acessado.
XMSC_WMQ_CHANEXO L	O nome do canal a ser usado para uma conexão.
“XMSC_WMQ_CLIENT_RECONNECT_OPTIONS” na página 225	Esta propriedade especifica as opções de reconexão do cliente para novas conexões criadas por este factory..
“XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT” na página 225	Esta propriedade especifica a duração de tempo, em segundos, que uma conexão do cliente tenta reconectar.
XMSC_WMQ_CONNECTION_MODE	O modo pelo qual um aplicativo se conecta a um gerenciador de filas.
“XMSC_WMQ_CONNECTION_NAME_LIST” na página 226	Esta propriedade especifica os hosts aos quais o cliente tenta se reconectar depois que sua conexão é interrompida
XMSC_WMQ_FAIL_IF_QUIESCE	Se as chamadas para determinados métodos falharão se o gerenciador de filas ao qual o aplicativo está conectado estiver em um estado quiesce.
XMSC_WMQ_HOST_NAME	O nome do host ou o endereço IP do sistema no qual um gerenciador de filas é executado.
XMSC_WMQ_LOCAL_ADDRESS	Para uma conexão com um gerenciador de filas, essa propriedade especifica a interface de rede local a ser usada, a porta local ou o intervalo de portas locais a serem usadas ou ambos.

Tabela 29. Propriedades de ConnectionFactory (continuação)

Nome da propriedade	Descrição
XMSC_WMQ_MESSAGE_SELECTION	Determina se a seleção de mensagem é feita pelo cliente XMS ou pelo broker **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.
XMSC_WMQ_MSG_BATCH_SIZE	O número máximo de mensagens a serem recuperadas de uma fila em um lote ao usar a entrega de mensagem assíncrona. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.
XMSC_WMQ_POLLING_INTERVAL	Se cada listener de mensagem dentro de uma sessão não tiver mensagens adequadas em sua fila, este valor será o intervalo máximo, em milissegundos, que decorrerá antes que cada listener da mensagem tente novamente obter uma mensagem de sua fila. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.
“XMSC_WMQ_PROVIDER_VERSION” na página 235	A versão, liberação, nível de modificação e fix pack do gerenciador de filas ao qual o aplicativo pretende se conectar.
XMSC_WMQ_PORT	O número da porta na qual um gerenciador de filas atende às solicitações recebidas.
XMSC_WMQ_PUB_ACK_INTERVAL	O número de mensagens publicadas por um publicador antes do cliente XMS solicitar uma confirmação do broker. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.

Tabela 29. Propriedades de ConnectionFactory (continuação)	
Nome da propriedade	**Descrição**
"XMSC_WMQ_PUT_ASYNC_ALLOWED" na página 230	Essa propriedade determina se os produtores de mensagens têm permissão para usar as postagens assíncronas para enviar mensagens para esse destino.
XMSC_WMQ_QMGR_CCSID	O identificador (CCSID) do conjunto de caracteres codificados, ou página de código, no qual os campos de dados de caracteres definidos na Message Queue Interface (MQI) são trocados entre o cliente XMS e o cliente IBM WebSphere MQ .
XMSC_WMQ_QUEUE_MANAGER	O nome do gerenciador de filas para conexão.
XMSC_WMQ_RECEIVE_EXIT	Identifica uma saída de recebimento do canal para ser executada.
XMSC_WMQ_RECEIVE_EXIT_INIT	Os dados do usuário que são transmitidos para uma saída de recebimento de canal quando ela é chamada.
XMSC_WMQ_SECURITY_EXIT	Identifica uma saída de segurança do canal.
XMSC_WMQ_SECURITY_EXIT_INIT	Os dados do usuário que são transmitidos para uma saída de segurança do canal quando ela é chamada.
"XMSC_WMQ_SEND_CHECK_COUNT" na página 239	O número de chamadas de envio a serem permitidas entre a verificação de erros de postagem assíncrona, dentro de uma única sessão XMS não transacionada.
XMSC_WMQ_SEND_EXIT	Identifica uma saída de envio de canal.
XMSC_WMQ_SEND_EXIT_INIT	Os dados do usuário que são transmitidos para as saídas de envio do canal quando são chamadas.
"XMSC_WMQ_SHARE_CONV_ALLOWED" na página 240	Se uma conexão do cliente pode compartilhar seu soquete com outras conexões XMS de nível superior do mesmo processo para o mesmo gerenciador de filas, se as definições de canais corresponderem Essa propriedade é fornecida para permitir o isolamento completo de Conexões em soquetes separados, se necessário para desenvolvimento de aplicativos, manutenção ou razões operacionais.
XMSC_WMQ_SSL_CERT_STORES	Os locais dos servidores que retêm as listas de revogação de certificado (CRLs) a serem usadas em uma conexão SSL com um gerenciador de filas.
XMSC_WMQ_SSL_CIPHER_SPEC	O nome da CipherSpec a ser usada em uma conexão segura com um gerenciador de filas.
XMSC_WMQ_SSL_CIPHER_SUITE	O nome do CipherSuite a ser usado em uma conexão SSL ou TLS com um gerenciador de filas O protocolo usado para negociar a conexão segura depende do CipherSuite especificado.
XMSC_WMQ_SSL_CRYPTO_HW	Detalhes de configuração para o hardware criptográfico conectado ao sistema do cliente.

Tabela 29. Propriedades de ConnectionFactory (continuação)	
Nome da propriedade	**Descrição**
XMSC_WMQ_SSL_FIPS_REQUIRED	O valor dessa propriedade determina se um aplicativo pode ou não usar conjuntos de cifras compatíveis não FIPS. Se essa propriedade for configurada como true, apenas algoritmos do FIPS serão usados para a conexão cliente-servidor.
XMSC_WMQ_SSL_KEY_REPOSITORY	O local do arquivo do banco de dados de chaves no qual chaves e certificados são armazenados.
XMSC_WMQ_SSL_KEY_RESETCOUNT	O KeyResetCount representa o número total de bytes não criptografados enviados e recebidos dentro de uma conversa SSL antes de a chave secreta ser renegociada.
XMSC_WMQ_SSL_PEER_NAME	O nome do peer a ser usado em uma conexão SSL com um gerenciador de filas.
XMSC_WMQ_SYNCPOINT_ALL_GETS	Se todas as mensagens devem ser recuperadas de filas dentro do controle de ponto de sincronização.
“XMSC_WMQ_TARGET_CLIENT” na página 248	
XMSC_WMQ_TEMP_Q_PREFIX	O prefixo usado para formar o nome da IBM WebSphere MQ fila dinâmica que é criada quando o aplicativo cria um XMS fila temporária.
XMSC_WMQ_TEMP_TOPIC_PREFIX	Ao criar tópicos temporários, o XMS gera uma sequência de tópicos no formato "`TEMP/TEMPTOPICPREFIX/unique_id`" ou se essa propriedade for deixada com o valor padrão, apenas "`TEMP/unique_id`". Especificar um valor não vazio permite que as filas modelo específicas sejam definidas para criar as filas gerenciadas para assinantes de tópicos temporários criados sob essa conexão.
XMSC_WMQ_TEMPORARY_MODEL	O nome da fila modelo do IBM WebSphere MQ a partir da qual uma fila dinâmica é criada quando o aplicativo cria um XMS fila temporária.
XMSC_WPM_BUS_NAME	Para um connection factory, o nome do barramento de integração de serviços ao qual o aplicativo se conecta ou, para um destino, o nome do barramento de integração de serviços no qual o destino existe.
XMSC_WPM_CONNECTION_PROXIMIDADE	A configuração de proximidade de conexão para a conexão.
XMSC_WPM_DUR_SUB_HOME	O nome do mecanismo do sistema de mensagens no qual todas as assinaturas duráveis para uma conexão ou um destino são gerenciadas.
XMSC_WPM_LOCAL_ADDRESS	Para uma conexão com um barramento de integração de serviços, essa propriedade especifica a interface de rede local a ser usada, a porta local ou o intervalo de portas locais a serem usados ou ambos.
XMSC_WPM_NON_PERSISTENT_MAP	O nível de confiabilidade de mensagens não persistentes que são enviadas usando a conexão.
XMSC_WPM_PERSISTENT_MAP	O nível de confiabilidade de mensagens persistentes que são enviadas usando a conexão.

Tabela 29. Propriedades de ConnectionFactory (continuação)	
Nome da propriedade	**Descrição**
XMSC_WPM_PROVIDER_ENDPOINTS	Uma sequência de um ou mais endereços de terminal de servidores de autoinicialização.
XMSC_WPM_TARGET_GROUP	O nome de um grupo de destinos de mecanismos do sistema de mensagens.
XMSC_WPM_TARGET_SIGNIFICATIVO	O significado do grupo de destinos dos mecanismos do sistema de mensagens.
XMSC_WPM_TARGET_TRANSPORT_CHAIN	O nome da cadeia de transporte de entrada que o aplicativo deve usar para se conectar a um mecanismo do sistema de mensagens.
XMSC_WPM_TARGET_TYPE	O tipo do grupo de destinos de mecanismos do sistema de mensagens.
XMSC_WPM_TEMP_Q_PREFIX	O prefixo usado para formar o nome da fila temporária que é criada no barramento de integração de serviços quando o aplicativo cria um XMS fila provisória.
XMSC_WPM_TEMP_TOPIC_PREFIX	O prefixo usado para formar o nome de um tópico temporário que é criado pelo aplicativo.

Conceitos relacionados

ConnectionFactories e objetos de Conexão
Um objeto ConnectionFactory fornece um modelo que um aplicativo usa para criar um objeto Connection. O aplicativo usa o objeto Connection para criar um objeto Session.

Conexão com um Barramento de Integração de Serviços do WebSphere
O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

Conexões seguras para um IBM WebSphere MQ gerenciador de filas
Para permitir que um aplicativo XMS .NET faça conexões seguras para um IBM WebSphere MQ gerenciador de filas, as propriedades relevantes devem ser definidas no objeto ConnectionFactory .

Conexões seguras para um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus
Para ativar um XMS NET para fazer conexões seguras com um mecanismo do sistema de mensagens do WebSphere Serviço Integration Bus , as propriedades relevantes devem ser definidas no objeto ConnectionFactory .

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Referências relacionadas

Propriedades necessárias para objetos ConnectionFactory administrados
Quando um aplicativo cria um connection factory, um número de propriedades deve ser definido para criar uma conexão com um servidor de sistema de mensagens.

Propriedades de Dados ConnectionMeta

Uma visão geral do objeto de dados ConnectionMeta, com links para informações de referência mais detalhadas.

Tabela 30. Propriedades de Dados ConnectionMeta

Nome da propriedade	Descrição
XMSC_JMS_MAJOR_VERSION	O número da versão principal da especificação JMS na qual XMS é baseado. Essa propriedade é somente leitura.
XMSC_JMS_MINOR_VERSION	O número da versão secundária da especificação JMS na qual XMS é baseado. Essa propriedade é somente leitura.
XMSC_JMS_VERSION	O identificador de versão da especificação JMS na qual XMS é baseado. Essa propriedade é somente leitura.
XMSC_MAJOR_VERSION	O número da versão do cliente do XMS Essa propriedade é somente leitura.
XMSC_MINOR_VERSION	O número da liberação do cliente do XMS Essa propriedade é somente leitura.
XMSC_PROVIDER_NAME	O provedor do cliente XMS . Essa propriedade é somente leitura.
XMSC_VERSION	O identificador de versão do cliente XMS . Essa propriedade é somente leitura.

Propriedades de Destino

Uma visão geral das propriedades do objeto de Destino, com links para informações de referência mais detalhadas

Tabela 31. Propriedades de Destino

Nome da propriedade	Descrição
XMSC_DELIVERY_MODE	O modo de entrega de mensagens enviadas para o destino.
XMSC_PRIORITY	A prioridade das mensagens enviadas para o destino.
XMSC_RTT_MULTICAST	A configuração multicast para um connection factory ou destino.
XMSC_TIME_TO_LIVE	O tempo de vida para mensagens enviadas para o destino.
XMSC_WMQ_BROKER_VERSION	O tipo de corretor usado pelo aplicativo para uma conexão ou para o destino.
XMSC_WMQ_CCSID	O identificador (CCSID) do conjunto de caracteres codificados, ou página de códigos, em que as cadeias de dados de caracteres no corpo de uma mensagem estão quando o cliente XMS encaminha a mensagem para o destino.

Tabela 31. Propriedades de Destino (continuação)	
Nome da propriedade	**Descrição**
XMSC_WMQ_DUR_SUBQ	O nome da fila de assinantes para um assinante durável que está recebendo mensagens do destino. **Nota:** Essa propriedade pode ser usada com a Versão 2.0 do IBM Message Service Client para .NET , mas não tem efeito para um aplicativo conectado a um gerenciador de fila do IBM WebSphere MQ Versão 7.0 , a menos que a propriedade XMSC_WMQ_PROVIDER_VERSION do connection factory esteja configurada para um número de versão menor que 7.
XMSC_WMQ_ENCODING	Como os dados numéricos no corpo de uma mensagem são representados quando o cliente XMS encaminha a mensagem para o destino.
XMSC_WMQ_FAIL_IF_QUIESCE	Se as chamadas para determinados métodos falharão se o gerenciador de filas ao qual o aplicativo está conectado estiver em um estado quiesce.
"XMSC_WMQ_MESSAGE_BODY" na página 228	Essa propriedade determina se um aplicativo XMS processa o MQRFH2 de uma mensagem IBM WebSphere MQ como parte da carga útil da mensagem (ou seja, como parte do corpo da mensagem).
"XMSC_WMQ_MQMD_MESSAGE_CONTEXT" na página 229	Determina qual nível de contexto da mensagem deve ser configurado pelo aplicativo XMS . O aplicativo deve estar em execução com autoridade de contexto apropriado para esta propriedade entrar em vigor.
"XMSC_WMQ_MQMD_READ_ENABLED" na página 230	Essa propriedade determina se um aplicativo XMS pode extrair os valores de campos MQMD ou não
"XMSC_WMQ_MQMD_WRITE_ENABLED" na página 230	Esta propriedade determina se um aplicativo XMS pode ou não os valores de campos MQMD.
"XMSC_WMQ_READ_AHEAD_ALLOWED" na página 231	Essa propriedade determina se os consumidores de mensagens e os navegadores de fila têm permissão para usar leitura antecipada para obter mensagens não persistentes, não transacionais desse destino em um buffer interno antes de recebê-las.
"XMSC_WMQ_READ_AHEAD_CLOSE_POLICY" na página 231	Para mensagens que estão sendo entregues em um listener de mensagem assíncrona, essa propriedade determina o que acontece com as mensagens no buffer de leitura antecipada interno quando o consumidor de mensagens é fechado.
"XMSC_WMQ_RECEIVE_CCSID" na página 237	A propriedade de destino que configura o destino CCSID para a conversão de mensagens do gerenciador de filas. O valor é ignorado a menos que XMSC_WMQ_RECEIVE_CONVERSION seja configurado como WMQ_RECEIVE_CONVERSION_QMGR.
"XMSC_WMQ_RECEIVE_CONVERSION" na página 237	A propriedade de destino que determina se a conversão de dados será executada pelo gerenciador de filas.
XMSC_WMQ_TARGET_CLIENT	Se as mensagens enviadas para o destino contêm um cabeçalho MQRFH2.

Tabela 31. Propriedades de Destino (continuação)	
Nome da propriedade	**Descrição**
XMSC_WMQ_TEMP_TOPIC_PREFIX	Ao criar tópicos temporários, o XMS gera uma sequência de tópicos no formato "`TEMP/TEMPTOPICPREFIX/unique_id`" ou se essa propriedade for deixada com o valor padrão, apenas "`TEMP/unique_id`". Especificar um valor não vazio permite que as filas modelo específicas sejam definidas para criar as filas gerenciadas para assinantes de tópicos temporários criados sob essa conexão.
XMSC_WPM_BUS_NAME	Para um connection factory, o nome do barramento de integração de serviços ao qual o aplicativo se conecta ou, para um destino, o nome do barramento de integração de serviços no qual o destino existe.
XMSC_WPM_TOPIC_SPACE	O nome do espaço de tópico que contém o tópico.

Conceitos relacionados

ConnectionFactories e objetos de Conexão
Um objeto ConnectionFactory fornece um modelo que um aplicativo usa para criar um objeto Connection. O aplicativo usa o objeto Connection para criar um objeto Session.

Conexão com um Barramento de Integração de Serviços do WebSphere
O aplicativo Um XMS pode conectar a um WebSphere Serviço Integration Bus usando uma conexão TCP/IP direta ou usando HTTP sobre TCP/IP.

Destinos
Um aplicativo XMS usa um objeto de Destino para especificar o destino das mensagens que estão sendo enviadas e a origem de mensagens que estão sendo recebidas.

Curinga de destino
O XMS fornece suporte para curingas de destino, assegurando que os curingas possam ser transmitidos para o local no qual eles são necessários para correspondência Há um esquema curinga diferente para cada tipo de servidor com o qual XMS pode trabalhar.

Identificadores de recursos uniformes do tópico
O URI (Identificador Uniforme de Recursos (URI) do tópico especifica o nome do tópico; ele também pode especificar uma ou mais propriedades para ele.

Identificadores uniformes de recursos da fila
O URI para uma fila especifica o nome da fila; ele também pode especificar uma ou mais propriedades da fila.

Destinos Temporários
Os aplicativos XMS podem criar e usar destinos temporários.

Mapeamento de Propriedades para Objetos Administrados
Para permitir que os aplicativos usem IBM WebSphere MQ JMS e WebSphere Servidor de Aplicação connection factory e definições de objeto de destino, as propriedades recuperadas dessas definições devem ser mapeadas para as propriedades XMS correspondentes que podem ser configuradas em XMS connection factories e destinos.

Tarefas relacionadas

Criando Objetos Administrados
As definições de objeto ConnectionFactory e Destination que os aplicativos XMS requerem para fazer uma conexão com um servidor de sistema de mensagens devem ser criadas usando as ferramentas administrativas apropriadas.

Referências relacionadas

Propriedades Necessárias para Objetos de Destino Administrados

Um aplicativo que está criando um destino deve configurar várias propriedades que o aplicativo em um objeto Destination administrado.

Propriedades de InitialContext

Uma visão geral das propriedades do objeto InitialContext com links para informações de referência mais detalhadas.

Tabela 32. Propriedades de InitialContext	
Nome da propriedade	**Descrição**
XMSC_IC_URL	Para os contextos LDAP e FileSystem, o endereço do repositório que contém objetos administrados.

Tabela 33. Propriedades de InitialContext	
Nome da propriedade	**Descrição**
XMSC_IC_PROVIDER_URL	Usado para localizar o diretório de nomenclatura de JNDI para que o serviço de nomenclatura COS não precise estar no mesmo servidor que o serviço da web.
XMSC_IC_SECURITY_AUTHENTICATION	Com base na Java Interface de contexto SECURITY_AUTHENTICATION Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS
XMSC_IC_SECURITY_CREDENTIALS	Baseado na Java Interface de contexto SECURITY_CREDENTIALS Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS
XMSC_IC_SECURITY_PRINCIPAL	Baseado na Java Interface de contexto SECURITY_PRINCIPAL Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS
XMSC_IC_SECURITY_PROTOCOL	Com base na Java Interface de contexto SECURITY_PROTOCOL Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS
XMSC_IC_URL	Para os contextos LDAP e FileSystem, o endereço do repositório que contém objetos administrados. Para contextos de nomenclatura COS, o endereço do serviço da web que consulta os objetos no diretório.

Conceitos relacionados

Propriedades InitialContext
Os parâmetros do construtor InitialContext incluem o local do repositório de objetos administrados, fornecido como um indicador de recurso uniforme (URI). Para que um aplicativo estabeleça uma conexão com o repositório, pode ser necessário fornecer mais informações do que as informações contidas no URI.

Formato de URI para contextos iniciais XMS
O local do repositório de objetos administrados é fornecido como um indicador de recurso uniforme (URI). O formato do URI depende do tipo de contexto.

Recuperação de Objetos Administrados
XMS recupera um objeto administrado do repositório usando o endereço fornecido quando o objeto InitialContext é criado, ou nas propriedades InitialContext.

Tarefas relacionadas

Objetos InitialContext
Um aplicativo deve criar um contexto inicial a ser usado para fazer uma conexão com o repositório de objetos administrados para recuperar os objetos administrados necessários.

Propriedades de Mensagem

Uma visão geral das propriedades do objeto de Mensagem, com links para informações de referência mais detalhadas

Tabela 34. Propriedades de Mensagem

Nome da propriedade	Descrição
JMS_IBM_CHARACTER_SET	O identificador (CCSID) do conjunto de caracteres codificados, ou página de códigos, no qual as sequências de dados de caracteres no corpo da mensagem estão quando o cliente XMS encaminha a mensagem para seu destino desejado. No XMS , essa propriedade possui um valor numérico e é mapeado para CCSID. No entanto, essa propriedade é baseada em uma propriedade JMS, portanto, possui um valor de tipo de sequência e é mapeado para o conjunto de caracteres Java que representa esse CCSID numérico.
CODIFICAÇÃO DE JMS_IBM_ENCODING	Como os dados numéricos no corpo da mensagem são representados quando o cliente XMS encaminha a mensagem para seu destino desejado.
JMS_IBM_EXCEPTIONMESSAGE	Texto que descreve o motivo pelo qual a mensagem foi enviada para o destino de exceção. Essa propriedade é somente leitura.
JMS_IBM_ExceptionProblemDestination	O nome do destino em que a mensagem estava antes de a mensagem ser enviada para o destino de exceção.
JMS_IBM_EXCEPTIONREASON	Um código de razão que indica a razão pela qual a mensagem foi enviada para o destino de exceção.
JMS_IBM_EXCEPTIONTIMESTAMP	O horário em que a mensagem foi enviada para o destino de exceção.
JMS_IBM_FEEDBACK	Um código que indica a natureza de uma mensagem de relatório.
FORMATO JMS_IBM_FORMAT	A natureza dos dados do aplicativo na mensagem
JMS_IBM_LAST_MSG_IN_GROUP	Indicar se a mensagem é a última mensagem em um grupo de mensagens.
JMS_IBM_MSGTYPE	O tipo de mensagem.
JMS_IBM_PUTAPPLTYPE	O tipo de aplicativo que enviou a mensagem.
JMS_IBM_PUTDATE	A data em que a mensagem foi enviada.
JMS_IBM_PUTTIME	O horário em que a mensagem foi enviada.
JMS_IBM_REPORT_COA	Solicitar mensagens de relatório 'confirmar na chegada', especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.
JMS_IBM_REPORT_COD	Solicitar mensagens de relatório 'confirmar na entrega', especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.
JMS_IBM_REPORT_DISCARD_MSG	Solicitar que a mensagem seja descartada se não puder ser entregue a seu destino desejado.

Tabela 34. Propriedades de Mensagem (continuação)	
Nome da propriedade	**Descrição**
JMS_IBM_REPORT_EXCEPTION	Solicitar mensagens de relatório de exceção, especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.
JMS_IBM_REPORT_EXPIRATION	Solicitar mensagens de relatório de expiração, especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.
JMS_IBM_REPORT_NAN	Solicitar mensagens de relatório de notificação de ação negativa.
JMS_IBM_REPORT_PAN	Solicitar mensagens de relatório de notificação de ação positiva.
JMS_IBM_Report_Pass_Correl_ID	Solicitar que o identificador de correlação de qualquer mensagem de relatório ou de resposta seja o mesmo que o identificador de correlação da mensagem original.
JMS_IBM_REPORT_PASS_MSG_ID	Solicitar que o identificador de mensagem de qualquer mensagem de relatório ou resposta seja o mesmo que o identificador de mensagem da mensagem original.
JMS_IBM_RETAIN	Configurar essa propriedade indica ao gerenciador de filas para tratar uma mensagem como Publicação retida.
JMS_IBM_SYSTEM_MESSAGEID	Um identificador que identifica a mensagem com exclusividade dentro do barramento de integração de serviços. Essa propriedade é somente leitura.
JMSX_APPID	O nome do aplicativo que enviou a mensagem.
JMSX_DELIVERY_COUNT	O número de tentativas de entregar a mensagem.
JMSX_GROUPID	O identificador do grupo de mensagens ao qual a mensagem pertence.
JMSX_GROUPSEQ	O número de sequência da mensagem dentro de um grupo de mensagens.
JMSX_USERID	O identificador de usuários associado ao aplicativo que enviou a mensagem.

Propriedades JMS_IBM_MQMD*

IBM Message Service Client for .NET permite que aplicativos clientes leiam / gravem campos MQMD usando APIs. Ele também permite o acesso aos dados da mensagens do MQ Por padrão, o acesso ao MQMD é desativado e deve ser ativado explicitamente pelo aplicativo usando as propriedades de Destino XMSC_WMQ_MQMD_WRITE_ENABLED e XMSC_WMQ_MQMD_READ_ENABLED Essas duas propriedades são independentes entre si.

Todos os campos MQMD, exceto StrucId e Version são expostos como propriedades adicionais do objeto de Mensagem e são prefixados JMS_IBM_MQMD.

As propriedades JMS_IBM_MQMD* têm precedência mais alta sobre outras propriedades como JMS_IBM* descritas na tabela anterior.

Enviando mensagens

Todos os campos MQMD, exceto StrucId e Version, são representados. Essas propriedades referem-se apenas aos campos MQMD; quando uma propriedade ocorre tanto no MQMD quanto no

cabeçalho MQRFH2, a versão no MQRFH2 não é configurada nem extraída. Qualquer uma dessas propriedades pode ser configurada, exceto JMS_IBM_MQMD_BackoutCount. Qualquer valor configurado para JMS_IBM_MQMD_BackoutCount é ignorado.

Se uma propriedade tiver um comprimento máximo e você fornecer um valor que é muito longo, o valor será truncado.

Para determinadas propriedades, deve-se também configurar a propriedade XMSC_WMQ_MQMD_MESSAGE_CONTEXT no objeto de Destino O aplicativo deve estar em execução com autoridade de contexto apropriado para esta propriedade entrar em vigor. Se não configurar XMSC_WMQ_MQMD_MESSAGE_CONTEXT para um valor apropriado, o valor da propriedade será ignorado. Se você configurar XMSC_WMQ_MQMD_MESSAGE_CONTEXT para um valor apropriado, mas não tiver autoridade de contexto suficiente para o gerenciador de filas, uma exceção será emitida.. Propriedades que requerem valores específicos de XMSC_WMQ_MQMD_MESSAGE_CONTEXT são as seguintes.

As propriedades a seguir requerem que XMSC_WMQ_MQMD_MESSAGE_CONTEXT seja configurado como XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT ou XMSC_WMQ_MDCTX_SET_ALL_CONTEXT:

- JMS_IBM_MQMD_UserIdentifier
- JMS_IBM_MQMD_AccountingToken
- JMS_IBM_MQMD_ApplIdentityData

As seguintes propriedades requerem que XMSC_WMQ_MQMD_MESSAGE_CONTEXT seja configurado como XMSC_WMQ_MDCTX_SET_ALL_CONTEXT:

- JMS_IBM_MQMD_PutApplType
- JMS_IBM_MQMD_PutApplName
- JMS_IBM_MQMD_PutDate
- JMS_IBM_MQMD_PutTime
- JMS_IBM_MQMD_ApplOriginData

Como receber mensagens

Todas essas propriedades estarão disponíveis em uma mensagem recebida se a propriedade XMSC_WMQ_MQMD_READ_ENABLED estiver configurada como true, independentemente das propriedades reais que o aplicativo de produção configurou como true. Um aplicativo não pode modificar as propriedades de uma mensagem recebida a menos que todas as propriedades sejam limpas primeiro, de acordo com a especificação JMS. A mensagem recebida pode ser transmitida sem modificar as propriedades.

Nota: Se seu aplicativo receber uma mensagem de um destino com a propriedade XMSC_WMQ_MQMD_READ_ENABLED configurada como true e encaminhá-la para um destino com XMSC_WMQ_MQMD_WRITE_ENABLED configurado como true, isso resultará em todos os valores do campo MQMD da mensagem recebida sendo copiados na mensagem encaminhada. Tabela de propriedades

Tabela 35. Propriedades do objeto de Mensagem que representa os campos MQMD

Propriedade	**Descrição**	**Tipo**
JMS_IBM_MQMD_Report	Opções para as mensagens de relatório	System.Int32
JMS_IBM_MQMD_MsgType	Tipo de Mensagem	System.Int32
JMS_IBM_MQMD_Expiry	Tempo de vida da mensagem	System.Int32
JMS_IBM_MQMD_Feedback	Feedback ou código de razão	System.Int32
JMS_IBM_MQMD_Encoding	Codificação numérica de dados da mensagem	System.Int32

Tabela 35. Propriedades do objeto de Mensagem que representa os campos MQMD (continuação)		
Propriedade	**Descrição**	**Tipo**
JMS_IBM_MQMD_CodedCharSetId	Identificador do conjunto de caracteres de dados da mensagem	System.Int32
JMS_IBM_MQMD_Format	Nome do formato dos dados da mensagem	System.String
JMS_IBM_MQMD_PRIORITY **Nota:** Se você designar um valor para JMS_IBM_MQMD_PRIORITY que não esteja no intervalo de 0 a 9, esse valor violará a especificação JMS.	Prioridade da mensagem	System.Int32
JMS_IBM_MQMD_Persistence	Persistência de mensagem	System.Int32
JMS_IBM_MQMD_MSGID **Nota:** A especificação JMS indica que o ID de mensagem deve ser configurado pelo provedor JMS e que deve ser exclusivo ou nulo. Se você designar um valor para JMS_IBM_MQMD_MsgId, esse valor será copiado para o JMSMessageID. Portanto, ele não é configurado pelo provedor JMS e pode não ser exclusivo: este valor viola a especificação JMS..	ID da Mensagem	matriz de byte **Nota:** O uso de propriedades de matriz de bytes em uma mensagem viola a especificação JMS
JMS_IBM_MQMD_CORRELID **Nota:** Se você designar um valor para JMS_IBM_MQMD_CORRELID que inicia com a sequência 'ID:', esse valor viola a especificação JMS.	Identificador de correlação	matriz de byte **Nota:** O uso de propriedades de matriz de bytes em uma mensagem viola a especificação JMS
JMS_IBM_MQMD_BackoutCount	contador de backout	System.Int32
JMS_IBM_MQMD_ReplyToQ	Nome da fila de resposta	System.String
JMS_IBM_MQMD_ReplyToQMgr	Nome do gerenciador de filas de resposta	System.String
JMS_IBM_MQMD_UserIdentifier	Identificador de usuário	System.String
JMS_IBM_MQMD_AccountingToken	Símbolo de contabilidade	matriz de byte **Nota:** O uso de propriedades de matriz de bytes em uma mensagem viola a especificação JMS
JMS_IBM_MQMD_ApplIdentityData	dados do aplicativo relacionados à identidade	System.String
JMS_IBM_MQMD_PutApplType	Tipo de aplicativo que coloca a mensagem	System.Int32
JMS_IBM_MQMD_PutApplName	Nome do aplicativo que colocou a mensagem	System.String
JMS_IBM_MQMD_PutDate	Data quando a mensagem foi colocada	System.String

Tabela 35. Propriedades do objeto de Mensagem que representa os campos MQMD (continuação)		
Propriedade	**Descrição**	**Tipo**
JMS_IBM_MQMD_PutTime	Hora quando a mensagem foi colocada	System.String
JMS_IBM_MQMD_ApplOriginData	Os dados do aplicativo relacionados à origem	System.String
JMS_IBM_MQMD_GroupId	Identificador de grupo	matriz de byte **Nota:** O uso de propriedades de matriz de bytes em uma mensagem viola a especificação JMS
JMS_IBM_MQMD_MsgSeqNumber	Número de sequência da mensagem local no grupo	System.Int32
JMS_IBM_MQMD_Offset	Deslocamento dos dados na mensagem física a partir do início da mensagem lógica	System.Int32
JMS_IBM_MQMD_MsgFlags	Sinalizadores de mensagem	System.Int32
JMS_IBM_MQMD_OriginalLength	Comprimento da mensagem original	System.Int32

Consulte MQMD para obter detalhes adicionais..

Examples

Este exemplo resulta em uma mensagem sendo colocada em uma fila ou em um tópico com MQMD.UserIdentifier configurado como "JoeBloggs".

```
// Create a ConnectionFactory, connection, session, producer, message
// ...

// Create a destination
// ...

// Enable MQMD write
dest.setBooleanProperty(XMSC_WMQ_MQMD_WRITE_ENABLED,
      XMSC_WMQ_MQMD_WRITE_ENABLED_YES);

// Optionally, set a message context if applicable for this MD field
dest.setIntProperty(XMSC_WMQ_MQMD_MESSAGE_CONTEXT,
  XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT);

// On the message, set property to provide custom UserId
msg.setStringProperty(JMS_IBM_MQMD_USERIDENTIFIER, "JoeBloggs");

// Send the message
// ...
```

É necessário configurar XMSC_WMQ_MQMD_MESSAGE_CONTEXT antes de configurar JMS_IBM_MQMD_USERIDENTIFIER. Para obter mais informações sobre o uso de XMSC_WMQ_MQMD_MESSAGE_CONTEXT, consulte Propriedades do objeto Message.

Da mesma forma, é possível extrair o conteúdo dos campos do MQMD configurando XMSC_WMQ_MQMD_READ_ENABLED como true antes de receber uma mensagem e, em seguida, usando os métodos get da mensagem, como a propriedade getString As propriedades recebidas são somente leitura.

Este exemplo resulta no campo de valor contendo o valor do MQMD MQMD.ApplIdentityData campo de uma mensagem obtido de uma fila ou de um tópico.

```
// Create a ConnectionFactory, connection, session, consumer
// ...

// Create a destination
// ...

// Enable MQMD read
dest.setBooleanProperty(XMSC_WMQ_MQMD_READ_ENABLED, XMSC_WMQ_MQMD_READ_ENABLED_YES);

// Receive a message
// ...

// Get desired MQMD field value using a property
System.String value = rcvMsg.getStringProperty(JMS_IBM_MQMD_APPLIDENTITYDATA);
```

Propriedades de MessageConsumer

Uma visão geral das propriedades do objeto MessageConsumer com links para informações de referência mais detalhadas.

Tabela 36. Propriedades de MessageConsumer

Nome da propriedade	Descrição
XMSC_IS_SUBSCRIPTION_MULTICAST	Indica se as mensagens estão sendo entregues para o consumidor de mensagens usando WebSphere MQ Multicast Transport Essa propriedade é somente leitura.
XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST	Indica se as mensagens estão sendo entregues ao consumidor de mensagem usando WebSphere MQ Multicast Transport com uma qualidade de serviço confiável. Essa propriedade é somente leitura.

Consulte .As propriedades NET de IMessageConsumer para obter mais detalhes

Propriedades do MessageProducer

Uma visão geral das propriedades do objeto MessageProducer , com links para informações de referência mais detalhadas.

Consulte .Propriedades NET de IMessageProducer para obter mais detalhes.

Propriedades da Sessão.

Uma visão geral das propriedades do objeto de Sessão, com links para informações de referência mais detalhadas

Consulte .Propriedades NET de ISession para obter mais detalhes

Definições de propriedades

Este tópicoseção fornece uma definição de cada propriedade de objeto

Cada definição de propriedade inclui as seguintes informações:

- O tipo de dados da propriedade
- Os tipos de objeto que possuem a propriedade
- Para uma propriedade de Destino, o nome que pode ser usado em um URI (Identificador Uniforme de Recursos)
- Uma descrição mais detalhada da propriedade
- Os valores válidos da propriedade

- O valor padrão da propriedade

As propriedades cujos nomes começam com um dos seguintes prefixos são relevantes apenas para o tipo de conexão especificado:

XMSC_RTT
As propriedades são relevantes apenas para uma conexão em tempo real com um broker Os nomes das propriedades são definidos como constantes nomeadas no arquivo de cabeçalho `xmsc_rtt.h`

XMSC_WMQ
As propriedades são relevantes somente quando um aplicativo se conecta a um gerenciador de filas do IBM WebSphere MQ Os nomes das propriedades são definidos como constantes nomeadas no arquivo de cabeçalho `xmsc_wmq.h`

XMSC_WPM
As propriedades são relevantes apenas quando um aplicativo se conecta a um barramento de integração de serviços do WebSphere Os nomes das propriedades são definidos como constantes nomeadas no arquivo de cabeçalho `xmsc_wpm.h`

Salvo indicação em contrário em suas definições, as propriedades restantes são relevantes para todos os tipos de conexão. Os nomes das propriedades são definidos como constantes nomeadas no arquivo de cabeçalho `xmsc.h` Propriedades cujos nomes começam com o prefixo JMSX são JMS propriedades definidas de uma mensagem e propriedades cujos nomes começam com o prefixo JMS_IBM são IBM propriedades definidas de uma mensagem. Para obter mais informações sobre as propriedades das mensagens, consulte “Propriedades da mensagem um XMS” na página 74

A menos que indicado de outra forma em sua definição, cada propriedade é relevante nos domínios ponto-a-ponto e Publicação/Assinatura .

Um aplicativo pode obter e configurar o valor de qualquer propriedade, a não ser que a propriedade seja designada como somente leitura.

As propriedades a seguir são definidas:

JMS_IBM_CHARACTER_SET

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

O identificador (CCSID) do conjunto de caracteres codificados, ou página de códigos, no qual as sequências de dados de caracteres no corpo da mensagem estão quando o cliente XMS encaminha a mensagem para seu destino desejado. No XMS , essa propriedade possui um valor numérico e é mapeado para CCSID. No entanto, essa propriedade é baseada em uma propriedade JMS, portanto, possui um valor de tipo de sequência e é mapeado para o conjunto de caracteres Java que representa esse CCSID numérico. Essa propriedade substitui qualquer CCSID especificado para o destino pela propriedade XMSC_WMQ_CCSID .

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

CODIFICAÇÃO DE JMS_IBM_ENCODING

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Como os dados numéricos no corpo da mensagem são representados quando o cliente XMS encaminha a mensagem para seu destino desejado. Essa propriedade substitui qualquer codificação especificada para o destino pela propriedade XMSC_WMQ_ENCODING A propriedade especifica a representação de números inteiros binários, números inteiros decimais empacotados e números de vírgula flutuante..

Os valores válidos da propriedade são iguais aos valores que podem ser especificados no campo *Encoding* de um descritor de mensagens. Para obter mais informações sobre o campo *Encoding* , consulte o *IBM WebSphere MQ Application Programming Reference*.

Um aplicativo pode usar as seguintes constantes nomeadas para configurar a propriedade:

Constante nomeada	Significado
MQENC_INTEGER_NORMAL	Codificação de número inteiro normal
MQENC_INTEGER_REVERSED	Codificação de número inteiro revertida
MQENC_DECIMAL_NORMAL	Codificação decimal compactada normal
MQENC_DECIMAL_REVERSED	Codificação decimal compactada revertida
MQENC_FLOAT_IEEE_NORMAL	Codificação de ponto flutuante IEEE normal
MQENC_FLOAT_IEEE_REVERSED	Codificação de ponto flutuante IEEE revertida
MQENC_FLOAT_S390	Codificação de ponto flutuante da arquitetura z/OS
MQENC_NATIVE	Codificação de máquina nativa

Para formar um valor para a propriedade, o aplicativo pode incluir três dessas constantes da seguinte forma:

- Uma constante cujo nome começa com MQENC_INTEGER, para especificar a representação de inteiros binários
- Uma constante cujo nome começa com MQENC_DECIMAL, para especificar a representação de números inteiros decimais compactados
- Uma constante cujo nome começa com MQENC_FLOAT, para especificar a representação de números de ponto flutuante

Como alternativa, o aplicativo pode configurar a propriedade para MQENC_NATIVE, cujo valor é dependente de ambiente.

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

JMS_IBM_EXCEPTIONMESSAGE

Tipo de dado:
Sequência

Propriedade de:
Mensagem

Texto que descreve o motivo pelo qual a mensagem foi enviada para o destino de exceção. Essa propriedade é somente leitura.

Essa propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços e recebe uma mensagem de um destino de exceções

JMS_IBM_ExceptionProblemDestination

Tipo de dado:
Sequência

Propriedade de:
Mensagem

O nome do destino em que a mensagem estava antes de a mensagem ser enviada para o destino de exceção.

Essa propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços e recebe uma mensagem de um destino de exceções

JMS_IBM_EXCEPTIONREASON

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Um código de razão que indica a razão pela qual a mensagem foi enviada para o destino de exceção.

Para obter uma lista de todos os códigos de razão possíveis, consulte a definição da classe com.ibm.websphere.sib.SIRCConstants na documentação gerada pela ferramenta Javadoc, conforme fornecido com WebSphere Servidor de Aplicação

Essa propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços e recebe uma mensagem de um destino de exceções

JMS_IBM_EXCEPTIONTIMESTAMP

Tipo de dado:
System.Int64

Propriedade de:
Mensagem

O horário em que a mensagem foi enviada para o destino de exceção.

O tempo é expresso em milissegundos desde 00:00:00 GMT de 1 de janeiro de 1970.

Essa propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços e recebe uma mensagem de um destino de exceções

JMS_IBM_FEEDBACK

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Um código que indica a natureza de uma mensagem de relatório.

Os valores válidos da propriedade são os códigos de feedback e de razão que podem ser especificados no campo **Feedback** de um descritor de mensagens. Para obter mais informações sobre o campo **Feedback** , consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

FORMATO JMS_IBM_FORMAT

Tipo de dado:
Sequência

Propriedade de:
Mensagem

A natureza dos dados do aplicativo na mensagem

Os valores válidos da propriedade são iguais aos valores que podem ser especificados no campo **Format** de um descritor de mensagens. Para obter mais informações sobre o campo **Format** , consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

JMS_IBM_LAST_MSG_IN_GROUP

Tipo de dado:
System.Boolean

Propriedade de:
Mensagem

Indicar se a mensagem é a última mensagem em um grupo de mensagens.

Configure a propriedade como true se a mensagem for a última mensagem em um grupo de mensagens.. Caso contrário, configure a propriedade como false, ou não configure a propriedades. Por padrão, a propriedade não é configurada

O valor true corresponde à sinalização de status MQMF_LAST_MSG_IN_GROUP, que pode ser especificada no campo **MsgFlags** de um descritor de mensagens. Para obter mais informações sobre esse sinalizador, consulte o *IBM WebSphere MQ Application Programming Reference*.

Essa propriedade é ignorada no domínio Publicação/Assinatura e não é relevante quando um aplicativo se conecta a um barramento de integração de serviços..

JMS_IBM_MSGTYPE

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

O tipo de mensagem.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
MQMT_DATAGRAM	A mensagem é uma que não requer resposta.
MQMT_REQUEST	A mensagem é aquela que requer uma resposta.
MQMT_REPLY	A mensagem é uma mensagem de resposta
MQMT_REPORT	A mensagem é uma mensagem de relatório

Esses valores correspondem aos tipos de mensagens que podem ser especificados no campo **MsgType** de um descritor de mensagens Para obter mais informações sobre o campo **MsgType** , consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

JMS_IBM_PUTAPPLTYPE

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

O tipo de aplicativo que enviou a mensagem.

Os valores válidos da propriedade são os tipos de aplicativo que podem ser especificados no campo **PutApplType** de um descritor de mensagens Para obter mais informações sobre o campo **PutApplType** , consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

JMS_IBM_PUTDATE

Tipo de dado:
Sequência

Propriedade de:
Mensagem

A data em que a mensagem foi enviada.

Os valores válidos da propriedade são iguais aos valores que podem ser especificados no campo **PutDate** de um descritor de mensagens. Para obter mais informações sobre o campo **PutDate** , consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

JMS_IBM_PUTTIME

Tipo de dado:
Sequência

Propriedade de:
Mensagem

O horário em que a mensagem foi enviada.

Os valores válidos da propriedade são iguais aos valores que podem ser especificados no campo **PutTime** de um descritor de mensagens. Para obter mais informações sobre o campo **PutTime** , consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

Essa propriedade não é relevante quando um aplicativo se conecta a um barramento de integração de serviços

JMS_IBM_REPORT_COA

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar mensagens de relatório 'confirmar na chegada', especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
MQRO_COA	Solicite mensagens de relatório 'confirmar na chegada', sem dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Valor válido	Significado
MQRO_COA_WITH_DATA	Solicite 'confirmar na chegada' mensagens de relatório, com os primeiros 100 bytes de dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_COA_WITH_FULL_DATA	Solicite mensagens de relatório 'confirmar na chegada', com todos os dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Esses valores correspondem às opções do relatório que podem ser especificadas no campo **Report** de um descritor de mensagens Para obter mais informações sobre essas opções, consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

JMS_IBM_REPORT_COD

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar mensagens de relatório 'confirmar na entrega', especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
MQRO_COD	Solicite 'confirmar na entrega' mensagens de relatório, sem dados do aplicativo da mensagem original incluída em uma mensagem de relatório
MQRO_COD_WITH_DATA	Solicite 'confirmar na entrega' mensagens de relatório, com os primeiros 100 bytes de dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_COD_WITH_FULL_DATA	Solicite 'confirmar na entrega' mensagens de relatório, com todos os dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Esses valores correspondem às opções do relatório que podem ser especificadas no campo **Report** de um descritor de mensagens Para obter mais informações sobre essas opções, consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

JMS_IBM_REPORT_DISCARD_MSG

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar que a mensagem seja descartada se não puder ser entregue a seu destino desejado.

Configure a propriedade para MQRO_DISCARD_MSG para solicitar que a mensagem seja descartada se não puder ser entregue para seu destino desejado. Se você requerer que a mensagem seja colocada em uma fila de devoluções ou enviada para um destino de exceções, não configure a propriedade Por padrão, a propriedade não é configurada

O valor MQRO_DISCARD_MSG corresponde a uma opção de relatório que pode ser especificada no campo **Report** de um descritor de mensagens. Para obter mais informações sobre essa opção, consulte o *IBM WebSphere MQ Application Programming Reference*.

JMS_IBM_REPORT_EXCEPTION

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar mensagens de relatório de exceção, especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
MQRO_EXCEPTION	Mensagens de relatório de exceção de solicitação, sem dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_EXCEPTION_WITH_DATA	Mensagens de relatório de exceção de solicitação, com os primeiros 100 bytes de dados do aplicativo da mensagem original incluídos em uma mensagem de relatório.
MQRO_EXCEPTION_WITH_FULL_DATA	Mensagens de relatório de exceção de solicitação, com todos os dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Esses valores correspondem às opções do relatório que podem ser especificadas no campo **Report** de um descritor de mensagens Para obter mais informações sobre essas opções, consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

JMS_IBM_REPORT_EXPIRATION

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar mensagens de relatório de expiração, especificando quantos dados do aplicativo da mensagem original devem ser incluídos em uma mensagem de relatório.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
MQRO_EXPIRATION	Mensagens de relatório de expiração de solicitação, sem dados do aplicativo da mensagem original incluídos em uma mensagem de relatório
MQRO_EXPIRATION_WITH_DATA	Mensagens de relatório de expiração de solicitação, com os primeiros 100 bytes de dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Valor válido	Significado
MQRO_EXPIRATION_WITH_FULL_DATA	Mensagens de relatório de expiração de solicitação, com todos os dados do aplicativo da mensagem original incluídos em uma mensagem de relatório

Esses valores correspondem às opções do relatório que podem ser especificadas no campo **Report** de um descritor de mensagens Para obter mais informações sobre essas opções, consulte o *IBM WebSphere MQ Application Programming Reference*.

Por padrão, a propriedade não é configurada

JMS_IBM_REPORT_NAN

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar mensagens de relatório de notificação de ação negativa.

Configure a propriedade para MQRO_NAN para solicitar mensagens de notificação de ação negativa. Se você não precisar de mensagens de notificação de ação negativa, não configure a propriedade. Por padrão, a propriedade não é configurada

O valor MQRO_NAN corresponde a uma opção de relatório que pode ser especificada no campo **Report** de um descritor de mensagens Para obter mais informações sobre essa opção, consulte o *IBM WebSphere MQ Application Programming Reference*.

JMS_IBM_REPORT_PAN

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar mensagens de relatório de notificação de ação positiva.

Configure a propriedade como MQRO_PAN para solicitar mensagens de relatório de notificação de ação positiva Se você não precisar de mensagens de notificação de ação positiva, não configure a propriedade. Por padrão, a propriedade não é configurada

O valor MQRO_PAN corresponde a uma opção de relatório que pode ser especificada no campo **Report** de um descritor de mensagem. Para obter mais informações sobre essa opção, consulte o *IBM WebSphere MQ Application Programming Reference*.

JMS_IBM_Report_Pass_Correl_ID

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar que o identificador de correlação de qualquer mensagem de relatório ou de resposta seja o mesmo que o identificador de correlação da mensagem original.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
MQRO_PASS_CORREL_ID	Solicitar que o identificador de correlação de qualquer mensagem de relatório ou de resposta seja o mesmo que o identificador de correlação da mensagem original.
MQRO_COPY_MSG_ID_TO_CORREL_ID	Solicite que o identificador de correlação de qualquer mensagem de relatório ou de resposta seja igual ao identificador de mensagem da mensagem original.

Esses valores correspondem às opções do relatório que podem ser especificadas no campo **Report** de um descritor de mensagens Para obter mais informações sobre essas opções, consulte o *IBM WebSphere MQ Application Programming Reference*.

O valor padrão da propriedade é MQRO_COPY_MSG_ID_TO_CORREL_ID.

JMS_IBM_REPORT_PASS_MSG_ID

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Solicitar que o identificador de mensagem de qualquer mensagem de relatório ou resposta seja o mesmo que o identificador de mensagem da mensagem original.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
MQRO_PASS_MSG_ID	Solicitar que o identificador de mensagem de qualquer mensagem de relatório ou resposta seja o mesmo que o identificador de mensagem da mensagem original.
MQRO_NEW_MSG_ID	Solicite que um novo identificador de mensagem seja gerado para cada mensagem de relatório ou resposta.

Esses valores correspondem às opções do relatório que podem ser especificadas no campo **Report** de um descritor de mensagens Para obter mais informações sobre essas opções, consulte o *IBM WebSphere MQ Application Programming Reference*.

O valor padrão da propriedade é MQRO_NEW_MSG_ID.

JMS_IBM_RETAIN

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

Configurar essa propriedade indica ao gerenciador de filas para tratar uma mensagem como Publicação retida. Quando um assinante recebe mensagens de tópicos, ele pode receber mensagens adicionais imediatamente após a assinatura, além das mensagens recebidas em liberações anteriores Essas mensagens são as publicações opcionais retidas para os tópicos inscritos Para cada tópico correspondente à assinatura, se houver uma publicação retida, a publicação será disponibilizada para entrega ao consumidor de mensagens de assinatura.

RETAIN_PUBLICATION é o único valor válido para esta propriedade. Por padrão, esta propriedade não é definida.

Nota: Esta propriedade é relevante apenas no domínio de publicação / assinatura

JMS_IBM_SYSTEM_MESSAGEID

Tipo de dado:
Sequência

Propriedade de:
Mensagem

Um identificador que identifica a mensagem com exclusividade dentro do barramento de integração de serviços. Essa propriedade é somente leitura.

Esta propriedade é relevante apenas quando um aplicativo se conecta a um barramento de integração de serviços

JMSX_APPID

Tipo de dado:
Sequência

Propriedade de:
Mensagem

O nome do aplicativo que enviou a mensagem.

Essa propriedade é a propriedade definida JMS com o JMS name JMSXAppID. Para obter mais informações sobre a propriedade, consulte a *Java Message Service Specification, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

JMSX_DELIVERY_COUNT

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

O número de tentativas de entregar a mensagem.

Essa propriedade é a propriedade definida pelo JMS com o JMS name JMSXDeliveryCount Para obter mais informações sobre a propriedade, consulte a *Java Message Service Specification, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

JMSX_GROUPID

Tipo de dado:
Sequência

Propriedade de:
Mensagem

O identificador do grupo de mensagens ao qual a mensagem pertence.

Essa propriedade é a propriedade definida JMS com o JMS nome JMSXGroupID. Para obter mais informações sobre a propriedade, consulte a *Java Message Service Specification, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

JMSX_GROUPSEQ

Tipo de dado:
System.Int32

Propriedade de:
Mensagem

O número de sequência da mensagem dentro de um grupo de mensagens.

Essa propriedade é a propriedade definida JMS com o JMS nome JMSXGroupSeq. Para obter mais informações sobre a propriedade, consulte a *Java Message Service Specification, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

JMSX_USERID

Tipo de dado:
Sequência

Propriedade de:
Mensagem

O identificador de usuários associado ao aplicativo que enviou a mensagem.

Essa propriedade é a propriedade definida JMS com o JMS name JMSXUserID. Para obter mais informações sobre a propriedade, consulte a *Java Message Service Specification, Versão 1.1*.

Por padrão, a propriedade não é configurada

Essa propriedade não é válida para uma conexão em tempo real com um broker

XMSC_ASYNC_EXCEPÇÕES

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

Essa propriedade determina se o XMS informa um ExceptionListener apenas quando uma conexão é quebrada ou quando qualquer exceção ocorre de forma assíncrona para uma chamada da API do XMS. Essa propriedade se aplica a todas as Conexões criadas por meio dessa ConnectionFactory que possui um ExceptionListener registrado.

Os valores válidos para essa propriedade são:

XMSC_ASYNC_EXCEPTIONS_ALL
Qualquer exceção detectada de forma assíncrona, fora do escopo de uma chamada de API síncrona, e todas as exceções de conexão interrompida são enviadas para o ExceptionListener

XMSC_ASYNC_EXCEPTIONS_CONNECTIONBROKEN
Apenas exceções indicando uma conexão interrompida são enviadas para o ExceptionListener. Quaisquer outras exceções que ocorram durante o processamento assíncrono não são relatadas para o ExceptionListenere, portanto, o aplicativo não é informado sobre essas exceções

Por padrão, essa propriedade é configurada para XMSC_ASYNC_EXCEPTIONS_ALL

XMSC_CLIENT_ID

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O identificador do cliente para uma conexão.

Um identificador de cliente é usado apenas para suportar assinaturas duráveis no domínio Publicação/Assinatura e é ignorado no domínio ponto-a-ponto . Para obter mais informações sobre a configuração de identificadores de cliente, consulte “ConnectionFactories e objetos de Conexão” na página 23

Essa propriedade não é relevante para uma conexão em tempo real com um broker

XMSC_CONNECTION_TYPE

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O tipo de servidor de mensagens para o qual um aplicativo se conecta.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_CT_RTT	Uma conexão em tempo real com um corretor.
XMSC_CT_WMQ	Uma conexão com um gerenciador de filas do IBM WebSphere MQ
XMSC_CT_WPM	Uma conexão com um barramento de integração de serviços do WebSphere

Por padrão, a propriedade não é configurada

XMSC_DELIVERY_MODE

Tipo de dado:
System.Int32

Propriedade de:
Destino

Nome usado em um URI:

`persistência` (para um destino IBM WebSphere MQ)
`deliveryMode` (para um WebSphere destino do provedor de sistemas de mensagens padrão)

O modo de entrega de mensagens enviadas para o destino.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_DELIVERY_NOT_PERSISTENT	Uma mensagem enviada para o destino é não persistente O modo de entrega padrão do produtor de mensagem, ou qualquer modo de entrega especificado na chamada Enviar, é ignorado. Se o destino for uma fila IBM WebSphere MQ , o valor do atributo da fila *DefPersistence* também será ignorado.
XMSC_DELIVERY_PERSISTENT	Uma mensagem enviada para o destino é persistente O modo de entrega padrão do produtor de mensagem, ou qualquer modo de entrega especificado na chamada Enviar, é ignorado. Se o destino for uma fila IBM WebSphere MQ , o valor do atributo da fila *DefPersistence* também será ignorado.

Valor válido	Significado
XMSC_DELIVERY_AS_APP	Uma mensagem enviada para o destino tem o modo de entrega especificado na chamada Enviar. Se a chamada Enviar especificar nenhum modo de entrega, o modo de entrega padrão do produtor da mensagem será usado no lugar. Se o destino for uma fila IBM WebSphere MQ , o valor do atributo da fila *DefPersistence* será ignorado.
XMSC_DELIVERY_AS_DEST	Se o destino for uma fila IBM WebSphere MQ , uma mensagem colocada na fila terá o modo de entrega especificado pelo valor do atributo da fila *DefPersistence*. O modo de entrega padrão do produtor de mensagem, ou qualquer modo de entrega especificado na chamada Enviar, é ignorado. Se o destino não for uma fila IBM WebSphere MQ , o significado será o mesmo de XMSC_DELIVERY_AS_APP.

O valor-padrão é XMSC_DELIVERY_AS_APP.

XMSC_IC_PROVIDER_URL

Tipo de dado:
Sequência

Propriedade de:
InitialContext

Usado para localizar o diretório de nomenclatura de JNDI para que o serviço de nomenclatura COS não precise estar no mesmo servidor que o serviço da web.

XMSC_IC_SECURITY_AUTHENTICATION

Tipo de dado:
Sequência

Propriedade de:
InitialContext

Com base na Java Interface de contexto SECURITY_AUTHENTICATION Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS

XMSC_IC_SECURITY_CREDENTIALS

Tipo de dado:
Sequência

Propriedade de:
InitialContext

Baseado na Java Interface de contexto SECURITY_CREDENTIALS Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS

XMSC_IC_SECURITY_PRINCIPAL

Tipo de dado:
Sequência

Propriedade de:
InitialContext

Baseado na Java Interface de contexto SECURITY_PRINCIPAL Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS

XMSC_IC_SECURITY_PROTOCOL

Tipo de dado:
Sequência

Propriedade de:
InitialContext

Com base na Java Interface de contexto SECURITY_PROTOCOL . Essa propriedade é aplicável apenas ao contexto de nomenclatura do COS

XMSC_IC_URL

Tipo de dado:
Sequência

Propriedade de:
InitialContext

Para os contextos LDAP e FileSystem, o endereço do repositório que contém objetos administrados.

Para contextos de nomenclatura COS, o endereço do serviço da web que consulta os objetos no diretório.

XMSC_IS_SUBSCRIPTION_MULTICAST

Tipo de dado:
System.Boolean

Propriedade de:
MessageConsumer

Indica se as mensagens estão sendo entregues para o consumidor de mensagens usando WebSphere MQ Multicast Transport Essa propriedade é somente leitura.

O valor da propriedade será true se as mensagens estiverem sendo entregues ao consumidor de mensagens usando WebSphere MQ Multicast Transport. Caso contrário, o valor será false.

Essa propriedade é relevante apenas para uma conexão em tempo real com um broker

XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST

Tipo de dado:
System.Boolean

Propriedade de:
MessageConsumer

Indica se as mensagens estão sendo entregues ao consumidor de mensagem usando WebSphere MQ Multicast Transport com uma qualidade de serviço confiável. Essa propriedade é somente leitura.

O valor da propriedade será true se as mensagens estiverem sendo entregues para o consumidor de mensagens usando WebSphere MQ Multicast Transport com uma qualidade de serviço confiável Caso contrário, o valor será false.

Essa propriedade é relevante apenas para uma conexão em tempo real com um broker

XMSC_JMS_MAJOR_VERSION

Tipo de dado:
System.Int32

Propriedade de:
ConnectionMetaData

O número da versão principal da especificação JMS na qual XMS é baseado. Essa propriedade é somente leitura.

XMSC_JMS_MINOR_VERSION

Tipo de dado:
System.Int32

Propriedade de:
ConnectionMetaData

O número da versão secundária da especificação JMS na qual XMS é baseado. Essa propriedade é somente leitura.

XMSC_JMS_VERSION

Tipo de dado:
Sequência

Propriedade de:
ConnectionMetaData

O identificador de versão da especificação JMS na qual XMS é baseado. Essa propriedade é somente leitura.

XMSC_MAJOR_VERSION

Tipo de dado:
System.Int32

Propriedade de:
ConnectionMetaData

O número da versão do cliente do XMS Essa propriedade é somente leitura.

XMSC_MINOR_VERSION

Tipo de dado:
System.Int32

Propriedade de:
ConnectionMetaData

O número da liberação do cliente do XMS Essa propriedade é somente leitura.

XMSC_PASSWORD

Tipo de dado:
matriz de byte

Propriedade de:
ConnectionFactory

Uma senha que pode ser usada para autenticar o aplicativo quando ele tenta se conectar a um servidor de mensagens. A senha é usada com a propriedade XMSC_USERID

Por padrão, a propriedade não é configurada

Se você estiver se conectando ao IBM WebSphere MQ em plataformas distribuídas e configurar a propriedade XMSC_USERID do connection factory, ele deverá corresponder ao **userid** do usuário conectado. Se você não configurar essas propriedades, o gerenciador de filas usará o **userid** do usuário conectado por padrão. Se você precisar de autenticação no nível de conexão adicional de usuários individuais, poderá gravar uma saída de autenticação de cliente que está configurada em IBM WebSphere MQ É possível aprender mais sobre a criação de uma saída de autenticação de cliente no tópico Autenticação no manual de Clientes do IBM WebSphere MQ

Para autenticar o usuário ao conectar ao IBM WebSphere MQ on z/OS , é necessário usar uma saída de segurança.

XMSC_PRIORITY

Tipo de dado:
System.Int32

Propriedade de:
Destino

Nome usado em um URI:
priority

A prioridade das mensagens enviadas para o destino.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
Um número inteiro no intervalo 0, a prioridade mais baixa, para 9, a prioridade mais alta	Uma mensagem enviada para o destino tem a prioridade especificada A prioridade padrão do produtor de mensagens, ou qualquer prioridade especificada na chamada Enviar, é ignorada. Se o destino for uma fila IBM WebSphere MQ , o valor do atributo da fila **DefPriority** também será ignorado.
XMSC_PRIORITY_AS_APP	Uma mensagem enviada para o destino tem a prioridade especificada na chamada Enviar. Se a chamada Enviar especificar nenhuma prioridade, a prioridade padrão do produtor da mensagem será usada em seu lugar. Se o destino for uma fila IBM WebSphere MQ , o valor do atributo da fila **DefPriority** será ignorado.
XMSC_PRIORITY_AS_DEST	Se o destino for uma fila IBM WebSphere MQ , uma mensagem colocada na fila terá a prioridade especificada pelo valor do atributo da fila **DefPriority**. A prioridade padrão do produtor de mensagens, ou qualquer prioridade especificada na chamada Enviar, é ignorada. Se o destino não for uma fila IBM WebSphere MQ , o significado será o mesmo que XMSC_PRIORITY_AS_APP.

O valor-padrão é XMSC_PRIORITY_AS_APP.

WebSphere MQ Transporte em Tempo Real e WebSphere MQ Multicast Transport não tomam nenhuma ação com base na prioridade de uma mensagem.

XMSC_PROVIDER_NAME

Tipo de dado:
Sequência

Propriedade de:
ConnectionMetaData

O provedor do cliente XMS . Essa propriedade é somente leitura.

XMSC_RTT_BROKER_PING_INTERVAL

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O intervalo de tempo, em milissegundos, após o qual o XMS .NET verifica a conexão com um servidor do sistema de mensagens Real Time para detectar qualquer atividade. Se nenhuma atividade for detectada, o cliente iniciará um ping; a conexão será fechada se nenhuma resposta for detectada para o ping

O valor-padrão da propriedade é 30000.

XMSC_RTT_CONNECTION_PROTOCOL

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O protocolo de comunicações usado para uma conexão em tempo real com um broker.

O valor da propriedade deve ser XMSC_RTT_CP_TCP, o que significa uma conexão em tempo real com um broker sobre TCP/IP. O valor padrão é XMSC_RTT_CP_TCP.

XMSC_RTT_HOST_NAME

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome do host ou o endereço IP do sistema no qual um broker é executado.

Essa propriedade é usada com a propriedade XMSC_RTT_PORT para identificar o broker

Por padrão, a propriedade não é configurada

XMSC_RTT_LOCAL_ADDRESS

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome do host ou o endereço IP da interface de rede local a ser usada para uma conexão em tempo real com um broker.

Esta propriedade é útil apenas se o sistema no qual o aplicativo está em execução tiver duas ou mais interfaces de rede e você precisar ser capaz de especificar qual interface deve ser usada para uma conexão em tempo real Se o sistema tiver apenas uma interface de rede, somente essa interface poderá ser usada Se o sistema tiver duas ou mais interfaces de rede, e a propriedade não estiver configurada, a interface será selecionada aleatoriamente

Por padrão, a propriedade não é configurada

XMSC_RTT_MULTICAST

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory e Destino

Nome usado em um URI:
mulicast

A configuração multicast para um connection factory ou destino. Apenas um destino que é um tópico pode ter essa propriedade

Um aplicativo usa esta propriedade para ativar multicast em associação com uma conexão em tempo real com um broker e, se multicast estiver ativado, para especificar a forma precisa na qual multicast é usado

para entregar mensagens do broker para um consumidor de mensagem.. A propriedade não tem efeito sobre como um produtor de mensagem envia mensagens para o broker

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_RTT_MULTICAST_DISABLED	As mensagens não são entregues a um consumidor de mensagens usando WebSphere MQ Multicast Transport Esse valor é o valor padrão para um objeto ConnectionFactory ..
XMSC_RTT_MULTICAST_ASCF	As mensagens são entregues a um consumidor de mensagens de acordo com a configuração de multicast para o connection factory associado ao consumidor de mensagem A configuração de multicast para o connection factory é observada no momento em que a conexão é criada Esse valor é válido apenas para um objeto de Destino e é o valor padrão para um objeto de Destino..
XMSC_RTT_MULTICAST_ENABLED	Se o tópico for configurado para multicast no broker, as mensagens serão entregues a um consumidor de mensagem usando WebSphere MQ Multicast Transport Uma qualidade de serviço confiável será usada, se o tópico for configurado para multicast confiável
XMSC_RTT_MULTICAST_RELIABLE	Se o tópico for configurado para multicast confiável no broker, as mensagens serão entregues a um consumidor de mensagens usando o WebSphere MQ Multicast Transport com uma qualidade de serviço confiável Se o tópico não estiver configurado para multicast confiável, não será possível criar um consumidor de mensagens para o tópico
XMSC_RTT_MULTICAST_NOT_RELIABLE	Se o tópico for configurado para multicast no broker, as mensagens serão entregues a um consumidor de mensagem usando WebSphere MQ Multicast Transport Uma qualidade de serviço confiável não será usada, mesmo se o tópico estiver configurado para multicast confiável

XMSC_RTT_PORT

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O número da porta na qual um broker atende às solicitações recebidas. No broker, deve-se configurar um nó de processamento de mensagens de Fluxo Real-timeInput ou Real-timeOptimizedpara atender nessa porta

Essa propriedade é usada com a propriedade XMSC_RTT_HOST_NAME para identificar o broker

O valor padrão da propriedade é XMSC_RTT_DEFAULT_PORT ou 1506.

XMSC_TIME_TO_LIVE

Tipo de dado:
System.Int32

Propriedade de:
Destino

Nome usado em um URI:

expiração (para um destino IBM WebSphere MQ)
timeToLive (para um destino do provedor de sistemas de mensagens padrão do WebSphere)

O tempo de vida para mensagens enviadas para o destino.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
0	Uma mensagem enviada ao destino nunca expira.
Um inteiro positivo	Uma mensagem enviada para o destino tem o tempo especificado para viver em milissegundos O tempo de vida padrão do produtor de mensagem, ou qualquer tempo de vida especificado na chamada Enviar, é ignorado.
XMSC_TIME_TO_LIVE_AS_APP	Uma mensagem enviada para o destino tem o tempo de vida especificado na chamada Enviar. Se a chamada de Envio especificar nenhum tempo de vida, o tempo de vida padrão do produtor da mensagem será usado em seu lugar

O valor padrão é XMSC_TIME_TO_LIVE_AS_APP.

XMSC_USERID

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Um identificador de usuário que pode ser usado para autenticar o aplicativo quando ele tenta se conectar a um servidor de mensagens. O identificador de usuário é usado com a propriedade XMSC_PASSWORD .

Por padrão, a propriedade não é configurada

Se você estiver se conectando a plataformas distribuídas do IBM WebSphere MQ e configurar a propriedade XMSC_USERID do connection factory, ela deverá corresponder ao **userid** do usuário conectado. Se você não configurar essas propriedades, o gerenciador de filas usará o **userid** do usuário conectado por padrão. If you require further connection-level authentication of individual users you can write a client authentication exit which is configured in IBM WebSphere MQ.

Para autenticar o usuário ao conectar ao IBM WebSphere MQ on z/OS , é necessário usar uma saída de segurança.

XMSC_VERSION

Tipo de dado:
Sequência

Propriedade de:
ConnectionMetaData

O identificador de versão do cliente XMS . Essa propriedade é somente leitura.

XMSC_WMQ_BROKER_CONTROLQ

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome da fila de controle usada por um broker.

O valor padrão da propriedade é SYSTEM.BROKER.CONTROL.QUEUE.

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura

XMSC_WMQ_BROKER_PUBQ

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome da fila monitorada por um broker na qual os aplicativos enviam mensagens que publicam.

O valor padrão da propriedade é SYSTEM.BROKER.DEFAULT.STREAM.

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura

XMSC_WMQ_BROKER_QMGR

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome do gerenciador de filas ao qual um broker está conectado.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura

XMSC_WMQ_BROKER_SUBQ

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome da fila de assinantes para um consumidor de mensagens não durável.

O nome da fila de assinantes deve começar com os seguintes caracteres:

SYSTEM.JMS.ND.

Se você desejar que todos os consumidores de mensagens não duráveis compartilhem uma fila de assinantes, especifique o nome completo da fila compartilhada Uma fila com o nome especificado deve existir antes que um aplicativo possa criar um consumidor de mensagens não duráveis

Se desejar que cada consumidor de mensagens não duráveis recupere mensagens de sua própria fila de assinantes exclusivos, especifique um nome de fila que termine com um asterisco (*). Em seguida, quando um aplicativo cria um consumidor de mensagens não duráveis, o cliente XMS cria uma fila dinâmica para uso exclusivo pelo consumidor de mensagens.. O cliente XMS usa o valor da propriedade para configurar o conteúdo do campo **DynamicQName** no descritor de objeto usado para criar a fila dinâmica.

O valor padrão da propriedade é SYSTEM.JMS.ND.SUBSCRIBER.QUEUE, que significa que XMS usa a abordagem de fila compartilhada por padrão.

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura

XMSC_WMQ_BROKER_VERSION

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory e Destino

Nome usado em um URI:
brokerVersion

O tipo de corretor usado pelo aplicativo para uma conexão ou para o destino. Apenas um destino que é um tópico pode ter essa propriedade

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_WMQ_BROKER_V1	O aplicativo está usando um broker do IBM WebSphere MQ Publicação/Assinatura O aplicativo também pode usar esse valor se você migrar de IBM WebSphere MQ Publicação/Assinatura para WebSphere Event Broker ou WebSphere Message Broker , mas não alterou o aplicativo.
XMSC_WMQ_BROKER_V2	O aplicativo está usando um broker de WebSphere Event Broker ou WebSphere Message Broker
XMSC_WMQ_BROKER_UNSPECIFIED	Após o broker ser migrado da Versão 6 para a Versão 7, configure essa propriedade para que os cabeçalhos RFH2 não sejam mais usados. Após a migração, essa propriedade não é mais relevante.

O valor padrão para um connectionfactory é XMSC_WMQ_BROKER_UNESPECIFICADO mas, por padrão, a propriedade não é configurada para um destino. A configuração da propriedade para um destino substitui qualquer valor especificado pela propriedade do connection factory.

XMSC_WMQ_CCDTURL

Tipo de dado:
System.String

Propriedade de:
ConnectionFactory

Um Localizador Uniforme de Recursos (URL) que identifica o nome e o local do arquivo que contém a tabela de definição de canal do cliente e também especifica como o arquivo pode ser acessado.

Por padrão, essa propriedade não está configurada.

XMSC_WMQ_CCSID

Tipo de dado:
System.Int32

Propriedade de:
Destino

Nome usado em um URI:
CCSID

O identificador (CCSID) do conjunto de caracteres codificados, ou página de códigos, em que as cadeias de dados de caracteres no corpo de uma mensagem estão quando o cliente XMS

encaminha a mensagem para o destino. Se configurado para uma mensagem individual, a propriedade JMS_IBM_CHARACTER_SET substituirá o CCSID especificado para o destino por esta propriedade

O valor-padrão da propriedade é 1208.

Essa propriedade é relevante apenas para mensagens enviadas para o destino, não para mensagens recebidas do destino

XMSC_WMQ_CHANEXO

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome do canal a ser usado para uma conexão.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de fila no modo de cliente

XMSC_WMQ_CLIENT_RECONNECT_OPTIONS

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Essa propriedade especifica as opções de reconexão do cliente para novas conexões criadas por esse factory Ele é encontrado no XMSC e é um dos seguintes:

- WMQ_CLIENT_RECONNECT_AS_DEF (padrão).. Use o valor especificado no arquivo `mqclient.ini` . Configure o valor usando a propriedade **DefRecon** dentro da sub-rotina Canais. Ele pode ser configurado para um de:
  1. Sim. Comporta-se como a opção WMQ_CLIENT_RECONNECT
  2. NÃO. default. Não especifica nenhuma opção de reconexão
  3. QMGR. Comporta-se como a opção WMQ_CLIENT_RECONNECT_Q_MGR
  4. Desativar. Comporta-se como a opção WMQ_CLIENT_RECONNECT_DISABLED
- WMQ_CLIENT_RECONNECT. Reconecte a qualquer um dos gerenciadores de filas especificados na lista de nomes de conexões.
- WMQ_CLIENT_RECONNECT_Q_MGR. Reconecta ao mesmo gerenciador de filas ao qual ele está originalmente conectado. Ele retorna `MQRC_RECONNECT_QMID_MISMATCH` se o gerenciador de filas ao qual ele tenta se conectar (especificado na lista de nomes de conexão) tiver um QMID diferente para o gerenciador de filas originalmente conectado.
- WMQ_CLIENT_RECONNECT_DISABLED. A reconexão está desativada

XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Esta propriedade especifica a duração de tempo, em segundos, que uma conexão do cliente tenta reconectar.

Depois de tentar se reconectar por esse tempo, o cliente falhará com MQRC_RECONNECT_FAILED. A configuração padrão para esta propriedade é XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT_DEFAULT.

O valor-padrão desta propriedade é 1800.

XMSC_WMQ_CONNECTION_MODE

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O modo pelo qual um aplicativo se conecta a um gerenciador de filas.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_WMQ_CM_BINDINGS	Uma conexão com um gerenciador de fila no modo de ligações, para obter um desempenho ideal Esse valor é o valor padrão para C/C + +.
XMSC_WMQ_CM_CLIENT	Uma conexão com um gerenciador de filas no modo cliente para assegurar uma pilha totalmente gerenciada. Esse valor é o valor padrão para .NET.
XMSC_WMQ_CM_CLIENT_UNMANAGED (apenas para .NET)	Uma conexão com um gerenciador de filas que força uma pilha de clientes não gerenciada

Conceitos relacionados
Operações gerenciadas e não gerenciada em .NET ...
O código gerenciado é executado exclusivamente dentro do ambiente Common Language Runtime do .NET e depende totalmente dos serviços fornecidos por esse tempo de execução. Um aplicativo é classificado como não gerenciado se qualquer parte do aplicativo executar ou chamar serviços fora do ambiente de tempo de execução de linguagem comum .NET .

XMSC_WMQ_CONNECTION_NAME_LIST

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Essa propriedade especifica os hosts aos quais o cliente tenta se reconectar após sua conexão ser interrompida.

A lista de nomes de conexão é uma lista separada por vírgula de pares de portas host / ip. A configuração padrão para essa propriedade é WMQ_CONNECTION_NAME_LIST_DEFAULT

Por exemplo,`127.0.0.1 (1414),host2.example.com(1400)`

A configuração padrão dessa propriedade é `localhost (1414)`.

XMSC_WMQ_DUR_SUBQ

Tipo de dado:
Sequência

Propriedade de:
Destino

O nome da fila de assinantes para um assinante durável que está recebendo mensagens do destino. Apenas um destino que é um tópico pode ter essa propriedade

O nome da fila de assinantes deve começar com os seguintes caracteres:

SYSTEM.JMS.D.

Se você desejar que todos os assinantes duráveis compartilhem uma fila de assinantes, especifique o nome completo da fila compartilhada Uma fila com o nome especificado deve existir antes que um aplicativo possa criar um assinante durável

Se desejar que cada assinante durável recupere mensagens de sua própria fila de assinantes exclusivos, especifique um nome da fila que termine com um asterisco (*). Então, quando um aplicativo cria um assinante durável, o cliente XMS cria uma fila dinâmica para uso exclusivo pelo assinante durável. O cliente XMS usa o valor da propriedade para configurar o conteúdo do campo **DynamicQName** no descritor de objeto usado para criar a fila dinâmica.

O valor padrão da propriedade é SYSTEM.JMS.D.SUBSCRIBER.QUEUE, que significa que XMS usa a abordagem de fila compartilhada por padrão.

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura

XMSC_WMQ_ENCODING

Tipo de dado:
System.Int32

Propriedade de:
Destino

Como os dados numéricos no corpo de uma mensagem são representados quando o cliente XMS encaminha a mensagem para o destino. Se configurado para uma mensagem individual, a propriedade JMS_IBM_ENCODING substituirá a codificação especificada para o destino por essa propriedade A propriedade especifica a representação de números inteiros binários, números inteiros decimais empacotados e números de vírgula flutuante..

Os valores válidos da propriedade são iguais aos valores que podem ser especificados no campo **Encoding** de um descritor de mensagens.

Um aplicativo pode usar as seguintes constantes nomeadas para configurar a propriedade:

Constante nomeada	Significado
MQENC_INTEGER_NORMAL	Codificação de número inteiro normal
MQENC_INTEGER_REVERSED	Codificação de número inteiro revertida
MQENC_DECIMAL_NORMAL	Codificação decimal compactada normal
MQENC_DECIMAL_REVERSED	Codificação decimal compactada revertida
MQENC_FLOAT_IEEE_NORMAL	Codificação de ponto flutuante IEEE normal
MQENC_FLOAT_IEEE_REVERSED	Codificação de ponto flutuante IEEE revertida
MQENC_FLOAT_S390	Codificação de ponto flutuante da arquitetura do z/OS
MQENC_NATIVE	Codificação de máquina nativa

Para formar um valor para a propriedade, o aplicativo pode incluir três dessas constantes da seguinte forma:

- Uma constante cujo nome começa com MQENC_INTEGER, para especificar a representação de inteiros binários
- Uma constante cujo nome começa com MQENC_DECIMAL, para especificar a representação de números inteiros decimais compactados
- Uma constante cujo nome começa com MQENC_FLOAT, para especificar a representação de números de ponto flutuante

Como alternativa, o aplicativo pode configurar a propriedade para MQENC_NATIVE, cujo valor é dependente de ambiente.

O valor padrão da propriedade é MQENC_NATIVE.

Essa propriedade é relevante apenas para mensagens enviadas para o destino, não para mensagens recebidas do destino

XMSC_WMQ_FAIL_IF_QUIESCE

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory e Destino

Nome usado em um URI:
failIfQuiesce

Se as chamadas para determinados métodos falharão se o gerenciador de filas ao qual o aplicativo está conectado estiver em um estado quiesce.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_WMQ_FIQ_YES	Chamadas para determinados métodos falharão se o gerenciador de filas estiver em um estado quiesce. Quando o aplicativo detectar que o gerenciador de filas está em quiesce, o aplicativo poderá concluir sua tarefa imediata e fechar a conexão, permitindo que o gerenciador de filas pare.
XMSC_WMQ_FIQ_NO	Nenhuma chamada de método falha porque o gerenciador de fila está em um estado de quiesce Se você especificar esse valor, o aplicativo não poderá detectar que o gerenciador de filas está quiesce. O aplicativo pode continuar a executar operações no gerenciador de fila e, portanto, evitar que o gerenciador de filas pare.

O valor padrão para um connection factory é XMSC_WMQ_FIQ_YES mas, por padrão, a propriedade não é configurada para um destino. A configuração da propriedade para um destino substitui qualquer valor especificado pela propriedade do connection factory.

XMSC_WMQ_MESSAGE_BODY

Tipo de dado:
System.Int32

Propriedade de:
Destino

Essa propriedade determina se um aplicativo XMS processa o MQRFH2 de uma mensagem IBM WebSphere MQ como parte da carga útil da mensagem (ou seja, como parte do corpo da mensagem).

Nota: Ao enviar mensagens para um destino, a propriedade XMSC_WMQ_MESSAGE_BODY substitui a propriedade de destino XMSC_WMQ_TARGET_CLIENT existente do XMS

Os valores válidos para essa propriedade são:

XMSC_WMQ_MESSAGE_BODY_JMS
Receber: O tipo de mensagem e o corpo do XMS de entrada são determinados pelo conteúdo do MQRFH2 (se presente) ou do MQMD (se não houver MQRFH2) na mensagem recebida IBM WebSphere MQ .

Enviar: O corpo da mensagem XMS de saída contém um cabeçalho MQRFH2 pré-anexado e gerado automaticamente com base nas propriedades da mensagem e nos campos do cabeçalho do XMS

XMSC_WMQ_MESSAGE_BODY_MQ
Receber: O tipo de mensagem XMS de entrada é sempre ByteMessage, independentemente do conteúdo da mensagem recebida do IBM WebSphere MQ ou do campo de formato do MQMD recebido O corpo da mensagem XMS é os dados da mensagem inalterados retornados pela chamada API do provedor de sistemas de mensagens subjacente. O conjunto de caracteres e a codificação dos dados

no corpo da mensagem são determinados pelos campos CodedCharSetId e Codificação do MQMD. O formato dos dados no corpo da mensagem é determinado pelo campo Formato do MQMD.

Enviar: o corpo da mensagem XMS de saída contém a carga útil do aplicativo no estado em que se encontra e nenhum cabeçalho IBM WebSphere MQ gerado automaticamente é incluído no corpo.

XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED

Recebimento: O cliente XMS determina um valor adequado para essa propriedade.. No caminho de recebimento, esse valor é o valor da propriedade WMQ_MESSAGE_BODY_JMS.

Enviar: O cliente XMS determina um valor adequado para essa propriedade. No caminho de envio, esse valor é o valor da propriedade XMSC_WMQ_TARGET_CLIENT.

Por padrão, essa propriedade é configurada como XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED

XMSC_WMQ_MQMD_MESSAGE_CONTEXT

Tipo de dado:
System.Int32

Propriedade de:
Destino

Determina qual nível de contexto da mensagem deve ser configurado pelo aplicativo XMS . O aplicativo deve estar em execução com autoridade de contexto apropriado para esta propriedade entrar em vigor.

Os valores válidos para essa propriedade são:

XMSC_WMQ_MDCTX_DEFAULT
Para mensagens de saída, a chamada da API MQOPEN e a estrutura MQPMO não especificam opções de contexto de mensagem explícitas

XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT
A chamada da API MQOPEN especifica a opção de contexto da mensagem MQOO_SET_IDENTITY_CONTEXT e a estrutura MQPMO especifica MQPMO_SET_IDENTITY_CONTEXT

XMSC_WMQ_MDCTX_SET_ALL_CONTEXT
A chamada da API MQOPEN especifica a opção de contexto da mensagem MQOO_SET_ALL_CONTEXT e a estrutura MQPMO especifica MQPMO_SET_ALL_CONTEXT

Por padrão, essa propriedade é configurada para XMSC_WMQ_MDCTX_DEFAULT

Nota: Essa propriedade não é relevante quando um aplicativo se conecta ao Integration Busdo sistema.

As propriedades a seguir requerem que a propriedade XMSC_WMQ_MQMD_MESSAGE_CONTEXT seja configurada como o valor da propriedade XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT ou o valor da propriedade XMSC_WMQ_MDCTX_SET_ALL_CONTEXT ao enviar uma mensagem para que tenha efeito desejado:

- JMS_IBM_MQMD_UserIdentifier
- JMS_IBM_MQMD_AccountingToken
- JMS_IBM_MQMD_ApplIdentityData

As propriedades a seguir requerem que a propriedade XMSC_WMQ_MQMD_MESSAGE_CONTEXT seja configurada para o valor da propriedade XMSC_WMQ_MDCTX_SET_ALL_CONTEXT ao enviar uma mensagem para que tenha efeito desejado:

- JMS_IBM_MQMD_PutApplType
- JMS_IBM_MQMD_PutApplName
- JMS_IBM_MQMD_PutDate
- JMS_IBM_MQMD_PutTime
- JMS_IBM_MQMD_ApplOriginData

XMSC_WMQ_MQMD_READ_ENABLED

Tipo de dado:
System.Int32

Propriedade de:
Destino

Essa propriedade determina se um aplicativo XMS pode extrair os valores de campos MQMD ou não

Os valores válidos para essa propriedade são:

XMSC_WMQ_READ_ENABLED_NO
Ao enviar mensagens, as propriedades JMS_IBM_MQMD* em uma mensagem enviada não são atualizadas para refletir os valores de campos atualizados no MQMD.

Ao receber mensagens, nenhuma das propriedades JMS_IBM_MQMD* está disponível em uma mensagem recebida, mesmo que algumas ou todas elas sejam configuradas pelo emissor.

XMSC_WMQ_READ_ENABLED_YES
Ao enviar mensagens, todas as propriedades JMS_IBM_MQMD* em uma mensagem enviada são atualizadas para refletir os valores de campo atualizados no MQMD, incluindo as propriedades que o emissor não configurou explicitamente...

Ao receber mensagens, todas as propriedades JMS_IBM_MQMD* estão disponíveis em uma mensagem recebida, incluindo aquelas propriedades que o emissor não configurou explicitamente.

Por padrão, essa propriedade é configurada como XMSC_WMQ_READ_ENABLED_NO

XMSC_WMQ_MQMD_WRITE_ENABLED

Tipo de dado:
System.Int32

Propriedade de:
Destino

Esta propriedade determina se um aplicativo XMS pode ou não os valores de campos MQMD.

Os valores válidos para essa propriedade são:

XMSC_WMQ_WRITE_ENABLED_NO
Todas as propriedades JMS_IBM_MQMD* são ignoradas e seus valores não são copiados para a estrutura MQMD subjacente.

XMSC_WMQ_WRITE_ENABLED_YES
As propriedades JMS_IBM_MQMD* são processadas. Seus valores serão copiados para a estrutura do MQMD subjacente.

Por padrão, essa propriedade é configurada como XMSC_WMQ_WRITE_ENABLED_NO

XMSC_WMQ_PUT_ASYNC_ALLOWED

Tipo de dado:
System.Int32

Propriedade de:
Destino

Essa propriedade determina se os produtores de mensagens têm permissão para usar as postagens assíncronas para enviar mensagens para esse destino.

Os valores válidos para essa propriedade são:

XMSC_WMQ _PUT_ASYNC_ALLOWED_AS_DEST
Determine se as colocações assíncronas são permitidas consultando a definição de fila ou tópico.

XMSC_WMQ _PUT_ASYNC_ALLOWED_AS_Q_DEF
Determine se as entradas assíncronas são permitidas consultando a definição de fila.

XMSC_WMQ _PUT_ASYNC_ALLOWED_AS_TOPIC_DEF
Determine se as colocações assíncronas são permitidas referindo-se à definição de tópico

XMSC_WMQ _PUT_ASYNC_ALLOWED_DISABLED
Puts assíncronos não são permitidos.

XMSC_WMQ _PUT_ASYNC_ALLOWED_ENABLED
As colocações assíncronas são permitidas

Por padrão, essa propriedade é configurada como XMSC_WMQ _PUT_ASYNC_ALLOWED_AS_DEST

Nota: Esta propriedade não é relevante quando um aplicativo está se conectando ao Integration Busdo sistema

XMSC_WMQ_READ_AHEAD_ALLOWED

Tipo de dado:
System.Int32

Propriedade de:
Destino

Essa propriedade determina se os consumidores de mensagens e os navegadores de fila têm permissão para usar leitura antecipada para obter mensagens não persistentes, não transacionais desse destino em um buffer interno antes de recebê-las.

Os valores válidos para essa propriedade são:

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_Q_DEF
Determine se a leitura antecipada é permitida referindo-se à definição de fila

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_ TÓPICO _DEF
Determine se a leitura antecipada é permitida referindo-se à definição de tópico

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST
Determine se a leitura antecipada é permitida consultando a definição de fila ou tópico.

XMSC_WMQ_READ_AHEAD_ALLOWED_DISABLED
A leitura antecipada não é permitida ao consumir ou procurar mensagens.

XMSC_WMQ_READ_AHEAD_ALLOWED_ENABLED
A leitura antecipada é permitida

Por padrão, essa propriedade é configurada como XMSC_WMQ _READ_AHEAD_ALLOWED_AS_DEST

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY

Tipo de dado:
System.Int32

Propriedade de:
Destino

Para mensagens que estão sendo entregues em um listener de mensagem assíncrona, essa propriedade determina o que acontece com as mensagens no buffer de leitura antecipada interno quando o consumidor de mensagens é fechado.

Essa propriedade é aplicável ao especificar opções de fila de fechamento ao consumir mensagens de um destino e não aplicável ao enviar mensagens para um destino.

Essa propriedade é ignorada para Navegadores de Fila, pois durante a procura as mensagens ainda estão disponíveis nas filas.

Os valores válidos para essa propriedade são:

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT
Apenas a chamada do listener de mensagem atual é concluída antes de retornar, potencialmente deixando mensagens no buffer de leitura antecipada interno, que são, então, descartados

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_ALL
Todas as mensagens no buffer de leitura antecipada interno são entregues para o listener de mensagens do aplicativo antes de retornar.

Por padrão, essa propriedade é configurada como XMSC_WMQ
_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT

Nota:

- **Terminação anormal do aplicativo**

  Todas as mensagens no buffer de leitura antecipada são perdidas quando um aplicativo XMS termina abruptamente.

- **Implicações nas transações**

  A leitura antecipada é desativada quando os aplicativos usam transação. Portanto, o aplicativo não está vendo nenhuma diferença no comportamento quando eles usam sessões transacionados

- **Implicações dos modos de Confirmação de sessão**

  A leitura antecipada é ativada em uma sessão não transacionada quando os modos de confirmação são XMSC_AUTO_RECONHEÇO ou XMSC_DUPS_OK_RECONHEÇO A leitura antecipada será desativada se o modo de confirmação de sessão for XMSC_CLIENT_RECONHEÇO, independentemente das sessões transacionados e não transacionadas

- **Implicações em Navegadores de Fila e Seletores de Navegador de Fila**

  Os Navegadores de Fila e os Seletores de Navegador de Filas, usados em aplicativos XMS , obtêm a vantagem de desempenho da leitura antecipada. Fechar o Queue Browser não compromete o desempenho, pois a mensagem ainda está disponível na fila para operações adicionais.. Não há qualquer outra implicação em navegadores de filas e seletores de navegadores de filas além dos benefícios de desempenho de leitura antecipada.

- **Implicações das propriedades de destino de leitura antecipada em gerenciadores de filas WebSphere Message Broker V6 ou anteriores**

  Especificando as propriedades de destino XMSC_WMQ_READ_AHEAD_ALLOWED e XMSC_WMQ_READ_AHEAD_CLOSE_POLICY, quando o aplicativo XMS usa o gerenciador de fila WebSphere Message Broker V6 não será capaz de usar os valores especificados. Esses valores de propriedade de destino serão ignorados silenciosamente, e os aplicativos continuarão a funcionar sem leitura antecipada Não haverá erros lançados quando usados com gerenciadores de filas V6 .

XMSC_WMQ_HOST_NAME

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome do host ou o endereço IP do sistema no qual um gerenciador de filas é executado.

Essa propriedade é usada apenas quando um aplicativo se conecta a um gerenciador de filas em modo cliente A propriedade é utilizada com a propriedade XMSC_WMQ_PORT para identificar o gerenciador de filas.

O valor padrão da propriedade é `localhost`..

XMSC_WMQ_LOCAL_ADDRESS

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Para uma conexão com um gerenciador de filas, essa propriedade especifica a interface de rede local a ser usada, a porta local ou o intervalo de portas locais a serem usadas ou ambos.

O valor da propriedade é uma sequência com o seguinte formato:

[*host_name*] [(*low_port*) [,*high_port*]]

Os significados das variáveis são os seguintes:

host_name
O nome do host ou o endereço IP da interface de rede local a ser usado para a conexão

Fornecer essas informações é necessário apenas se o sistema no qual o aplicativo está em execução tiver duas ou mais interfaces de rede e você precisar ser capaz de especificar qual interface deve ser usada para a conexão.. Se o sistema tiver apenas uma interface de rede, somente essa interface poderá ser usada Se o sistema tiver duas ou mais interfaces de rede e você não especificar qual interface deve ser usada, a interface será selecionada aleatoriamente

low_port
O número da porta local a ser usada para a conexão

Se *high_port* também for especificado, *low_port* será interpretado como o número da porta mais baixo em um intervalo de números de portas

high_port
O número da porta mais alto em um intervalo de números de porta Uma das portas no intervalo especificado deve ser usada para a conexão..

O comprimento máximo da sequência é de 48 caracteres.

Aqui estão alguns exemplos de valores válidos da propriedade:

JÚPITER
9.20.4.98
JUPITER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de fila no modo de cliente

XMSC_WMQ_MESSAGE_SELECTION

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

Determina se a seleção de mensagem é feita pelo cliente XMS ou pelo broker

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_WMQ_MSEL_CLIENT	A seleção de mensagens é feita pelo cliente do XMS

Valor válido	Significado
XMSC_WMQ_MSEL_BROKER	A seleção de mensagens é feita pelo broker

O valor padrão é XMSC_WMQ_MSEL_CLIENT.

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura A seleção de mensagens pelo broker não será suportada se a propriedade XMSC_WMQ_BROKER_VERSION estiver configurada como XMSC_WMQ_BROKER_V1.

XMSC_WMQ_MSG_BATCH_SIZE

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O número máximo de mensagens a serem recuperadas de uma fila em um lote ao usar a entrega de mensagem assíncrona.

Quando um aplicativo está usando a entrega de mensagem assíncrona, sob determinadas condições, o cliente XMS recupera um lote de mensagens de uma fila antes de encaminhar cada mensagem individualmente para o aplicativo Esta propriedade especifica o número máximo de mensagens que podem estar no lote

O valor da propriedade é um inteiro positivo e o valor padrão é 10. Considere configurar a propriedade para um valor diferente somente se você tiver um problema de desempenho específico que precisa ser abordado.

Se um aplicativo for conectado a um gerenciador de filas por meio de uma rede, aumentar o valor dessa propriedade pode reduzir sobrecargas de rede e tempos de resposta, mas aumentar a quantidade de memória necessária para armazenar as mensagens no sistema do cliente. Por outro lado, reduzir o valor dessa propriedade pode aumentar as sobrecargas de rede e os tempos de resposta, mas reduzir a quantidade de memória necessária para armazenar as mensagens..

XMSC_WMQ_POLLING_INTERVAL

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

Se cada listener de mensagem dentro de uma sessão não tiver mensagens adequadas em sua fila, este valor será o intervalo máximo, em milissegundos, que decorrerá antes que cada listener da mensagem tente novamente obter uma mensagem de sua fila.

Se ocorrer com frequência o fato de nenhuma mensagem adequada estar disponível para qualquer um dos listeners da mensagem em uma sessão, considere aumentar o valor desta propriedade.

O valor da propriedade é um número inteiro positivo. O valor-padrão é 5000.

XMSC_WMQ_PORT

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O número da porta na qual um gerenciador de filas atende às solicitações recebidas.

Essa propriedade é usada apenas quando um aplicativo se conecta a um gerenciador de filas em modo cliente A propriedade é usada com a propriedade XMSC_WMQ_HOST_NAME para identificar o gerenciador de filas

O valor padrão da propriedade é XMSC_WMQ_DEFAULT_CLIENT_PORT ou 1414.

XMSC_WMQ_PROVIDER_VERSION

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

A versão, liberação, nível de modificação e fix pack do gerenciador de filas ao qual o aplicativo pretende se conectar. Os valores válidos para essa propriedade são:

- Não especificado

Ou uma sequência em um dos seguintes formatos

- V.R.M.F
- V.R.M
- V.R
- V

em que V, R, M e F são valores inteiros maiores ou iguais a zero.

Um valor de 7 ou superior indica que esta versão é destinada como um IBM WebSphere MQ Versão 7.0 ConnectionFactory para conexões com um gerenciador de filas IBM WebSphere MQ Versão 7.0 . Um valor inferior a 7 (por exemplo, "6.0.2.0") indica que foi planejado para uso com gerenciadores de filas anteriores à Versão 7.0. O valor padrão, não especificado, permite conexões com qualquer nível do gerenciador de filas, determinando as propriedades e a funcionalidade aplicáveis disponíveis com base nos recursos do gerenciador de filas...

Por padrão, essa propriedade é configurada como "não especificada"..

Nota:

- Nenhum compartilhamento de soquete ocorrerá se XMSC_WMQ_PROVIDER_VERSION estiver configurado como 6 2.
- A conexão falhará se XMSC_WMQ_PROVIDER_VERSION estiver configurado como 7 e no servidor SHARECNV para o canal estiver configurado como 0
- Os recursos específicos do IBM WebSphere MQ Versão 7.0 serão desativados se XMSC_WMQ_PROVIDER_VERSION estiver configurado como UNSPECIFIED e SHARECNV estiver configurado como 0..

A versão do IBM WebSphere MQ Client também desempenha uma função principal se um aplicativo cliente XMS pode usar recursos específicos do IBM WebSphere MQ Versão 7.0 . A tabela a seguir descreve o comportamento..

Nota: Uma propriedade de sistema XMSC_WMQ_OVERRIDEPROVIDERVERSION substitui a propriedade XMSC_WMQ_PROVIDER_VERSION.. Esta propriedade pode ser usada se você não conseguir alterar a configuração do connection factory

Tabela 37. cliente XMS -Capacidade de usar recursos específicos do IBM WebSphere MQ Versão 7.0 .

#	XMSC_WMQ_PROVIDER_VERSION	Versão do cliente IBM WebSphere MQ	Recursos do IBM WebSphere MQ Versão 7.0
1	não especificado	7	ATIVADA
2	não especificado	6	Desativado
3	7	7	ATIVADA
4	7	6	Exceção

Tabela 37. cliente XMS -Capacidade de usar recursos específicos do IBM WebSphere MQ Versão 7.0 . (continuação)

#	XMSC_WMQ_PROVIDER_VERSION	Versão do cliente IBM WebSphere MQ	Recursos do IBM WebSphere MQ Versão 7.0
5	6	6	Desativado
6	6	7	Desativado

XMSC_WMQ_PUB_ACK_INTERVAL

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O número de mensagens publicadas por um publicador antes do cliente XMS solicitar uma confirmação do broker.

Se você diminuir o valor dessa propriedade, o cliente solicita reconhecimentos com mais frequência e, portanto, o desempenho do publicador diminui. Se aumentar o valor, o cliente terá mais tempo para emitir uma exceção se o intermediário falhar.

O valor da propriedade é um número inteiro positivo. O valor padrão é 25.

XMSC_WMQ_QMGR_CCSID

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O identificador (CCSID) do conjunto de caracteres codificados, ou página de código, no qual os campos de dados de caracteres definidos na Message Queue Interface (MQI) são trocados entre o cliente XMS e o cliente IBM WebSphere MQ . Essa propriedade não se aplica às cadeias dos dados de caracteres nos corpos das mensagens.

Quando o aplicativo um XMS se conecta a um gerenciador de fila no modo cliente, o cliente XMS vincula ao cliente IBM WebSphere MQ . As informações trocadas entre os dois clientes contêm campos de dados de caracteres definidos no MQI. Sob circunstâncias normais, o cliente IBM WebSphere MQ assume que esses campos estão na página de códigos do sistema no qual os clientes estão em execução Se o cliente XMS fornecer e esperar receber esses campos em uma página de código diferente, deve-se configurar essa propriedade para informar o cliente IBM WebSphere MQ

Quando o cliente IBM WebSphere MQ encaminha esses campos de dados de caractere para o gerenciador de filas, os dados neles devem ser convertidos, se necessário, na página de códigos usada pelo gerenciador de fila Da mesma forma, quando o cliente IBM WebSphere MQ recebe esses campos do gerenciador de filas, os dados neles devem ser convertidos, se necessário, para a página de código na qual o cliente XMS espera receber os dados O cliente IBM WebSphere MQ usa essa propriedade para executar essas conversão de dados.

Por padrão, a propriedade não é configurada

A configuração dessa propriedade é equivalente à configuração da variável de ambiente MQCCSID para um cliente IBM WebSphere MQ que suporta aplicativos clientes nativos IBM WebSphere MQ . Para obter informações adicionais sobre esta variável de ambiente, consulte *IBM WebSphere MQ Clientes*.

XMSC_WMQ_QUEUE_MANAGER

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome do gerenciador de filas para conexão.

Por padrão, a propriedade não é configurada

XMSC_WMQ_RECEIVE_CCSID

A propriedade de destino que configura o destino CCSID para a conversão de mensagens do gerenciador de filas. O valor é ignorado a menos que XMSC_WMQ_RECEIVE_CONVERSION seja configurado como WMQ_RECEIVE_CONVERSION_QMGR.

Tipo de dado:
Integer

Valor:
Qualquer número inteiro positivo.

O valor padrão é 1208.

XMSC_WMQ_RECEIVE_CONVERSION

A propriedade de destino que determina se a conversão de dados será executada pelo gerenciador de filas.

Tipo de dado:
Integer

Valores:
XMSC_WMQ_RECEIVE_CONVERSION_CLIENT_MSG (DEFAULT): Execute a conversão de dados somente no cliente XMS A conversão é sempre feita usando a página de códigos 1208

XMSC_WMQ_RECEIVE_CONVERSION_QMGR: Executar conversão de dados no gerenciador de filas antes de enviar uma mensagem ao cliente XMS .

XMSC_WMQ_RECEIVE_EXIT

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Identifica uma saída de recebimento do canal para ser executada.

O valor da propriedade é uma sequência que identifica uma saída de recebimento do canal e possui o seguinte formato:

libraryName(entryPointNome)

em que,

- **libraryName** é o caminho completo da saída gerenciada `.dll`
- `entryPointName` é o nome da classe qualificado pelo namespace.

Por exemplo,`C:\MyReceiveExit.dll(MyReceiveExitNameSpace.MyReceiveExitClassName)`

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado Além disso, apenas as saídas gerenciadas são suportadas

XMSC_WMQ_RECEIVE_EXIT_INIT

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Os dados do usuário que são transmitidos para uma saída de recebimento de canal quando ela é chamada.

O valor da propriedade é uma sequência. Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado e a propriedade "XMSC_WMQ_RECEIVE_EXIT" na página 237 é configurada

XMSC_WMQ_RESOLVED_QUEUE_MANAGER

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Esta propriedade é usada para obter o nome do gerenciador de fila ao qual ele está conectado,

Quando usado com uma CCDT (Client Channel Definition Table), esse nome pode ser diferente do nome do gerenciador de filas especificado na Connection Factory.

XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Essa propriedade é preenchida com o ID do gerenciador de filas após a conexão

XMSC_WMQ_SECURITY_EXIT

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Identifica uma saída de segurança do canal.

O valor da propriedade é uma sequência que identifica uma saída de segurança do canal e possui o seguinte formato:

libraryName(entryPointNome)

em que,

- **libraryName** é o caminho completo da saída gerenciada .dll
- entryPointName é o nome da classe qualificado pelo namespace.

Por exemplo, C:\MySecurityExit.dll(MySecurityExitNameSpace.MySecurityExitClassName)

O comprimento máximo da cadeia é de 128 caracteres.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado Além disso, apenas as saídas gerenciadas são suportadas

XMSC_WMQ_SECURITY_EXIT_INIT

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Os dados do usuário que são transmitidos para uma saída de segurança do canal quando ela é chamada.

O comprimento máximo da sequência de dados do usuário é de 32 caracteres.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado e a propriedade "XMSC_WMQ_SECURITY_EXIT" na página 238 é configurada

XMSC_WMQ_SEND_EXIT

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Identifica uma saída de envio de canal.

O valor da propriedade é uma sequência. Uma saída de envio de canal tem o seguinte formato:

libraryName(entryPointNome)

em que,

- **libraryName** é o caminho completo da saída gerenciada .dll
- entryPointName é o nome da classe qualificado pelo namespace.

Por exemplo, C:\MySendExit.dll(MySendExitNameSpace.MySendExitClassName)

Por padrão, a propriedade não é configurada

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado Além disso, apenas as saídas gerenciadas são suportadas

XMSC_WMQ_SEND_EXIT_INIT

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Os dados do usuário que são transmitidos para as saídas de envio do canal quando são chamadas.

O valor da propriedade é uma sequência de um ou mais itens de dados do usuário separados por vírgula. Por padrão, a propriedade não é configurada

As regras para especificar dados do usuário que são transmitidos para uma sequência de saídas de envio de canal são as mesmas que as regras para especificar dados do usuário que são transmitidos para uma sequência de saídas de recebimento de canal.. Portanto, para obter as regras, consulte "XMSC_WMQ_RECEIVE_EXIT_INIT" na página 237

Essa propriedade é relevante apenas quando um aplicativo se conecta a um gerenciador de filas no modo de cliente gerenciado e a propriedade "XMSC_WMQ_SEND_EXIT" na página 239 é configurada

XMSC_WMQ_SEND_CHECK_COUNT

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O número de chamadas de envio a serem permitidas entre a verificação de erros de postagem assíncrona, dentro de uma única sessão XMS não transacionada.

Por padrão, esta propriedade é configurada como 0

XMSC_WMQ_SHARE_CONV_ALLOWED

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

Se uma conexão do cliente pode compartilhar seu soquete com outras conexões XMS de nível superior do mesmo processo para o mesmo gerenciador de filas, se as definições de canais corresponderem Essa propriedade é fornecida para permitir o isolamento completo de Conexões em soquetes separados, se necessário para desenvolvimento de aplicativos, manutenção ou razões operacionais. A configuração dessa propriedade apenas indica para XMS para tornar o soquete subjacente compartilhado Ele não indica quantas conexões compartilham um único soquete O número de conexões compartilhando um soquete é determinado pelo valor SHARECNV que é negociado entre o IBM WebSphere MQ Client e o IBM WebSphere MQ Server.

Um aplicativo pode configurar as seguintes constantes nomeadas para configurar a propriedade:

- XMSC_WMQ_SHARE_CONV_ALLOWED_FALSE-As conexões não compartilham um soquete
- XMSC_WMQ_SHARE_CONV_ALLOWED_TRUE-As conexões compartilham um soquete

Por padrão, a propriedade é configurada como XMSC_WMQ_SHARE_CONV_ALLOWED_ENABLED

Essa propriedade é relevante somente quando um aplicativo se conecta a um gerenciador de fila no modo de cliente

XMSC_WMQ_SSL_CERT_STORES

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Os locais dos servidores que retêm as listas de revogação de certificado (CRLs) a serem usadas em uma conexão SSL com um gerenciador de filas.

O valor da propriedade é uma lista de uma ou mais URLs separadas por vírgulas. Cada URL tem o seguinte formato:

```
[user[/password]@]ldap://[serveraddress][:portnum][,...]
```

Esse formato é compatível com, mas estendido do, formato MQJMS básico.

É válido ter um `serveraddress`vazio. Neste caso, XMS assume que o valor é a sequência "localhost".

Uma lista de exemplo é:

```
myuser/mypassword@ldap://server1.mycom.com:389
ldap://server1.mycom.com
ldap://
ldap://:389
```

Somente para .NET : conexões gerenciadas para IBM WebSphere MQ (WMQ_CM_CLIENT) não suportam conexões SSL, mas podem ser suportadas usando uma conexão não gerenciada (WMQ_CM_CLIENT_UNMANAGED).

Por padrão, a propriedade não é configurada

XMSC_WMQ_SSL_CIPHER_SPEC

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

V 7.5.0.2 O nome da CipherSpec a ser usada em uma conexão segura com um gerenciador de filas.

As especificações de cifra que podem ser usadas com o suporte SSL e TLS do IBM WebSphere MQ são lisadas na tabela a seguir: Ao exigir um certificado pessoal, você especifica um tamanho de chave para o par de chaves público e particular. O tamanho de chave que é usado durante o handshake SSL é o tamanho armazenado no certificado, a menos que ele seja determinado pelo CipherSpec, conforme indicado na tabela. Por padrão, essa propriedade não está configurada.

Nome do CipherSpec	Protocolo utilizado	Algoritmo hash	Algoritmo de criptografia	Bits de Criptografia	FIPS[1]	Conjunto B de 128 bits	Conjunto B de 192 bits
NULL_MD5	SSL 3.0 [8]	MD5	Nenhum	0	NÃO	NÃO	NÃO
NULL_SHA	SSL 3.0 [8]	SHA-1	Nenhum	0	NÃO	NÃO	NÃO
RC4_MD5_EXPORT [2]	SSL 3.0 [8]	MD5	RC4	40	NÃO	NÃO	NÃO
RC4_MD5_US	SSL 3.0 [8]	MD5	RC4	128	NÃO	NÃO	NÃO
RC4_SHA_US	SSL 3.0 [8]	SHA-1	RC4	128	NÃO	NÃO	NÃO
RC2_MD5_EXPORT[2]	SSL 3.0 [8]	MD5	RC2	40	NÃO	NÃO	NÃO
DES_SHA_EXPORT [2]	SSL 3.0 [8]	SHA-1	Padrão de Criptografia de Dados	56	NÃO	NÃO	NÃO
RC4_56_SHA_EXPORT1024[3]	SSL 3.0 [8]	SHA-1	RC4	56	NÃO	NÃO	NÃO
DES_SHA_EXPORT1024 [3]	SSL 3.0 [8]	SHA-1	Padrão de Criptografia de Dados	56	NÃO	NÃO	NÃO
TRIPLE_DES_SHA_US	SSL 3.0 [8]	SHA-1	3DES	168	NÃO	NÃO	NÃO
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	Padrão de Criptografia Avançado	128	Sim	NÃO	NÃO

Nome do CipherSpec	Protocolo utilizado	Algoritmo hash	Algoritmo de criptografia	Bits de Criptografia	FIPS[1]	Conjunto B de 128 bits	Conjunto B de 192 bits
TLS_RSA_WITH_AES_256_CBC_SHA[4]	TLS 1.0	SHA-1	Padrão de Criptografia Avançado	256	Sim	NÃO	NÃO
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	Padrão de Criptografia de Dados	56	Não[5]	NÃO	NÃO
TLS_RSA_WITH_3DES_EDE_CBC_SHA[8]	TLS 1.0	SHA-1	3DES	168	Sim	NÃO	NÃO
FIPS_WITH_DES_CBC_SHA	SSL 3.0	SHA-1	Padrão de Criptografia de Dados	56	Não[6]	NÃO	NÃO
FIPS_WITH_3DES_EDE_CBC_SHA	SSL 3.0	SHA-1	3DES	168	Não[7]	NÃO	NÃO
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	SHA-256	Padrão de Criptografia Avançado	128	Sim	NÃO	NÃO
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	SHA-384	Padrão de Criptografia Avançado	256	Sim	NÃO	NÃO
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	SHA-256	Padrão de Criptografia Avançado	128	Sim	NÃO	NÃO
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	SHA-256	Padrão de Criptografia Avançado	256	Sim	NÃO	NÃO
ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	NÃO	NÃO	NÃO
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Sim	NÃO	NÃO

Nome do CipherSpec	Protocolo utilizado	Algoritmo hash	Algoritmo de criptografia	Bits de Criptografia	FIPS[1]	Conjunto B de 128 bits	Conjunto B de 192 bits
ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	NÃO	NÃO	NÃO
ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Sim	NÃO	NÃO
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	Padrão de Criptografia Avançado	128	Sim	NÃO	NÃO
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	Padrão de Criptografia Avançado	256	Sim	NÃO	NÃO
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	Padrão de Criptografia Avançado	128	Sim	NÃO	NÃO
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	Padrão de Criptografia Avançado	256	Sim	NÃO	NÃO
ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	Padrão de Criptografia Avançado	128	Sim	Sim	NÃO
ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	Padrão de Criptografia Avançado	256	Sim	NÃO	Sim
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	Padrão de Criptografia Avançado	128	Sim	NÃO	NÃO

Nome do CipherSpec	Protocolo utilizado	Algoritmo hash	Algoritmo de criptografia	Bits de Criptografia	FIPS[1]	Conjunto B de 128 bits	Conjunto B de 192 bits
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	Padrão de Criptografia Avançado	256	Sim	NÃO	NÃO
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	Nenhum	0	NÃO	NÃO	NÃO
ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-256	Nenhum	0	NÃO	NÃO	NÃO
ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-256	Nenhum	0	NÃO	NÃO	NÃO
TLS_RSA_WITH_NULL_NULL	TLS 1.2	Nenhum	Nenhum	0	NÃO	NÃO	NÃO
TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	NÃO	NÃO	NÃO

Notes:

1. Especifica se o CipherSpec está em conformidade com o Federal Information Processing Standards (FIPS) 140-2. Para obter uma explicação do FIPS e informações sobre como configurar a operação compatível com o WebSphere MQ para FIPS 140-2, consulte *Federal Information Processing Standards (FIPS)* na documentação on-line do produto IBM IBM WebSphere MQ .
2. O tamanho de chave de handshake máximo é 512 bits. Caso nenhum dos certificados trocados durante o protocolo de reconhecimento do SSL tenha um tamanho de chave superior a 512 bits, uma chave de 512 bits temporária será gerada para uso durante o protocolo de reconhecimento.
3. O tamanho de chave de handshake é 1024 bits.
4. Esse CipherSpec não pode ser usado para proteger uma conexão do WebSphere MQ Explorer com um gerenciador de filas, a menos que os arquivos de políticas irrestritos apropriados sejam aplicados ao JRE usado pelo Explorer.
5. Este CipherSpec era certificado FIPS 140-2 antes de 19 de Maio de 2007.
6. Este CipherSpec era certificado FIPS 140-2 antes de 19 de Maio de 2007. O nome do FIPS_WITH_DES_CBC_SHA é histórico e reflete o fato de que esse CipherSpec era anteriormente (mas não é mais) compatível com FIPS. Este CipherSpec foi descontinuado
7. O nome do FIPS_WITH_3DES_EDE_CBC_SHA é histórico e reflete o fato de que esse CipherSpec era anteriormente (mas não é mais) compatível com FIPS. O uso deste CipherSpec está descontinuado.
8. Quando o WebSphere MQ é configurado para a operação compatível com FIPS 140-2, esse CipherSpec pode ser usado para transferir até 32 GB de dados antes que a conexão seja finalizada com o erro AMQ9288 Para evitar esse erro, evite usar DES triplo (que foi descontinuado) ou ative a reconfiguração de chave secreta ao usar esse CipherSpec em uma configuração FIPS 140-2.

Conceitos relacionados

Segurança

Integridade de dados de mensagens

Tarefas relacionadas

Especificando CipherSpecs

XMSC_WMQ_SSL_CIPHER_SUITE

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome do CipherSuite a ser usado em uma conexão SSL ou TLS com um gerenciador de filas O protocolo usado para negociar a conexão segura depende do CipherSuite especificado.

Essa propriedade possui os seguintes valores canônicos:

- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA
- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- SSL_RSA_WITH_NULL_MD5
- SSL_RSA_WITH_NULL_SHA
- SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
- SSL_RSA_EXPORT1024_WITH_RC4_56_SHA
- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_RC4_128_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_AES_128_CBC_SHA
- SSL_RSA_WITH_AES_256_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA

Esse valor pode ser fornecido como uma alternativa para XMSC_WMQ_SSL_CIPHER_SPEC.

Se um valor não vazio for especificado para XMSC_WMQ_SSL_CIPHER_SPEC, esse valor substituirá a configuração para XMSC_WMQ_SSL_CIPHER_SUITE Se XMSC_WMQ_SSL_CIPHER_SPEC não tiver um valor, o valor de XMSC_WMQ_SSL_CIPHER_SUITE será usado como o conjunto de cifras a ser fornecido ao GSKit. Nesse caso, o valor é mapeado para o valor CipherSpec equivalente, conforme descrito em “Mapeamentos de Nome CipherSuite e CipherSpec para Conexões com um IBM WebSphere MQgerenciador de filas” na página 69.

Se XMSC_WMQ_SSL_CIPHER_SPEC e XMSC_WMQ_SSL_CIPHER_SUITE estiverem vazios, o campo pChDef->SSLCipherSpec será preenchido com espaços.

Somente para .NET : conexões gerenciadas para IBM WebSphere MQ (WMQ_CM_CLIENT) não suportarão conexões SSL, mas podem ser suportadas usando uma conexão não gerenciada (WMQ_CM_CLIENT_UNMANAGED).

Por padrão, a propriedade não é configurada

XMSC_WMQ_SSL_CRYPTO_HW

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Detalhes de configuração para o hardware criptográfico conectado ao sistema do cliente.

Essa propriedade possui os seguintes valores canônicos:

- GSK_ACCELERATOR_RAINBOW_CS_OFF
- GSK_ACCELERATOR_RAINBOW_CS_ON
- GSK_ACCELERATOR_NCIPHER_NF_OFF
- GSK_ACCELERATOR_NCIPHER_NF_ON

Há um formato especial para o hardware criptográfico PKCS11 (em que DriverPath, TokenLabele TokenPassword são sequências especificadas pelo usuário):

```
GSK_PKCS11=PKCS#11 DriverPath; PKCS#11 TokenLabel;PKCS#11 TokenPassword
```

O XMS não interpreta ou altera o conteúdo da sequência. Ele copia o valor fornecido, até um limite de 256 caracteres de byte único, no MQSCO MQSCO.CryptoHardware de CryptoHardware

Somente para .NET : conexões gerenciadas para IBM WebSphere MQ (WMQ_CM_CLIENT) não suportam conexões SSL, mas podem ser suportadas usando uma conexão não gerenciada (WMQ_CM_CLIENT_UNMANAGED).

Por padrão, a propriedade não é configurada

XMSC_WMQ_SSL_FIPS_REQUIRED

Tipo de dado:
Booleana

Propriedade de:
ConnectionFactory

O valor dessa propriedade determina se um aplicativo pode ou não usar conjuntos de cifras compatíveis não FIPS. Se essa propriedade for configurada como true, apenas algoritmos do FIPS serão usados para a conexão cliente-servidor.

Essa propriedade pode ter os valores a seguir, que são convertidos para os dois valores canônicos para o MQSCO MQSCO.FipsRequired:

Tabela 38. Tabela de valores para MQSCO.FlipsRequired

Value	**Descrição**	**Valor correspondente de MQSCO.FipsRequired**
false	Qualquer CipherSpec pode ser usado	MQSSL_FIPS_NO (o padrão),
true	Somente algoritmos criptográficos certificados pelo FIPS podem ser usados no CipherSpec que se aplica a essa conexão do cliente	MQSSL_FIPS_YES

XMS copia o valor relevante no MQSCO.FipsRequired antes de chamar MQCONNX.

O parâmetro MQSCO.FipsRequired está disponível apenas a partir da versão 6 do IBM WebSphere MQ Se IBM WebSphere MQ versão 5.3, se essa propriedade for configurada, o XMS não tentará fazer a conexão com o gerenciador de fila e, em vez disso, emitirá uma exceção apropriada

Somente para .NET : conexões gerenciadas para IBM WebSphere MQ (WMQ_CM_CLIENT) não suportam conexões SSL, mas podem ser suportadas usando uma conexão não gerenciada (WMQ_CM_CLIENT_UNMANAGED).

XMSC_WMQ_SSL_KEY_REPOSITORY

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O local do arquivo do banco de dados de chaves no qual chaves e certificados são armazenados.

XMS copia a sequência, até um limite de 256 caracteres de byte único, no MQSCO.KeyRepository do KeyRepository IBM WebSphere MQ interpreta essa sequência como um nome de arquivo, incluindo o caminho completo..

Para .NET apenas: as conexões gerenciadas para IBM WebSphere MQ (WMQ_CM_CLIENT) não suportam conexões SSL, mas podem ser suportadas usando uma conexão não gerenciada (WMQ_CM_CLIENT_UNMANAGED).

Por padrão, a propriedade não é configurada

XMSC_WMQ_SSL_KEY_RESETCOUNT

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O KeyResetCount representa o número total de bytes não criptografados enviados e recebidos dentro de uma conversa SSL antes de a chave secreta ser renegociada. O número de bytes inclui informações de controle enviadas pelo MCA.

XMS copia o valor fornecido para essa propriedade no MQSCO.KeyResetCount antes de chamar MQCONNX.

O parâmetro MQSCO.KeyRestCount está disponível apenas a partir do IBM WebSphere MQ versão 6 Se IBM WebSphere MQ versão 5.3, se essa propriedade for configurada, o XMS não tentará fazer a conexão com o gerenciador de fila e, em vez disso, emitirá uma exceção apropriada

Para .NET apenas: as conexões gerenciadas para IBM WebSphere MQ (WMQ_CM_CLIENT) não suportam conexões SSL, mas podem ser suportadas usando uma conexão não gerenciada (WMQ_CM_CLIENT_UNMANAGED).

O valor padrão dessa propriedade é zero, o que significa que chaves secretas nunca são renegociadas.

XMSC_WMQ_SSL_PEER_NAME

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome do peer a ser usado em uma conexão SSL com um gerenciador de filas.

Não há nenhuma lista de valores canônicos para esta propriedade Em vez disso, deve-se construir essa cadeia de acordo com as regras para SSLPEER

Um exemplo de nome de peer é:

```
"CN=John Smith, O=IBM ,OU=Test , C=GB"
```

XMS copia a sequência na página de códigos de byte único correta e coloca os valores corretos em MQCD.SSLPeerNamePtr e MQCD.SSLPeerNameLength antes de chamar MQCONNX.

Essa propriedade será relevante apenas se o aplicativo se conectar a um gerenciador de filas em modo cliente

Somente para .NET : conexões gerenciadas para IBM WebSphere MQ (WMQ_CM_CLIENT) não suportam conexões SSL, mas podem ser suportadas usando uma conexão não gerenciada (WMQ_CM_CLIENT_UNMANAGED).

Por padrão, a propriedade não é configurada

XMSC_WMQ_SYNCPOINT_ALL_GETS

Tipo de dado:
System.Boolean

Propriedade de:
ConnectionFactory

Se todas as mensagens devem ser recuperadas de filas dentro do controle de ponto de sincronização.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
false	Quando as circunstâncias forem apropriadas, o cliente XMS poderá recuperar mensagens de filas fora do controle do ponto de sincronização..
true	O cliente XMS deve recuperar todas as mensagens das filas dentro do controle de ponto de sincronização.

O valor padrão é falso.

XMSC_WMQ_TARGET_CLIENT

Tipo de dado:
System.Int32

Propriedade de:
Destino

Nome usado em um URI:
targetClient

Se as mensagens enviadas para o destino contêm um cabeçalho MQRFH2.

Se um aplicativo enviar uma mensagem contendo um cabeçalho MQRFH2 , o aplicativo de recepção deverá ser capaz de manipular o cabeçalho.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_WMQ_TARGET_DEST_JMS	As mensagens enviadas ao destino contêm um cabeçalho MQRFH2 . Especifique esse valor se o aplicativo estiver enviando as mensagens para outro aplicativo XMS , um aplicativo WebSphere JMS ou um aplicativo IBM WebSphere MQ nativo projetado para manipular um cabeçalho MQRFH2 .
XMSC_WMQ_TARGET_DEST_MQ	As mensagens enviadas ao destino não contêm um cabeçalho MQRFH2 . Especifique esse valor se o aplicativo estiver enviando as mensagens para um aplicativo IBM WebSphere MQ nativo que não foi projetado para manipular um cabeçalho MQRFH2 .

O valor padrão é XMSC_WMQ_TARGET_DEST_JMS.

XMSC_WMQ_TEMP_Q_PREFIX

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O prefixo usado para formar o nome da IBM WebSphere MQ fila dinâmica que é criada quando o aplicativo cria um XMS fila temporária.

As regras para formar o prefixo são as mesmas que as regras para formar o conteúdo do campo **DynamicQName** em um descritor de objeto, mas o último caractere não em branco deve ser um asterisco (*). Se a propriedade não for configurada, o valor usado será `CSQ.*` on z/OS e `AMQ.*` nas outras plataformas.. Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas para o domínio ponto-a-ponto

XMSC_WMQ_TEMP_TOPIC_PREFIX

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory, Destino

Ao criar tópicos temporários, o XMS gera uma sequência de tópicos no formato "`TEMP/TEMPTOPICPREFIX/unique_id`" ou se essa propriedade for deixada com o valor padrão, apenas "`TEMP/unique_id`". Especificar um valor não vazio permite que as filas modelo específicas sejam definidas para criar as filas gerenciadas para assinantes de tópicos temporários criados sob essa conexão.

Qualquer sequência não nula que consiste apenas em caracteres válidos para uma sequência de tópicos IBM WebSphere MQ é um valor válido para essa propriedade.

Por padrão, essa propriedade é configurada como "" (sequência vazia).

Nota: Essa propriedade é relevante apenas no domínio de publicação / assinatura

XMSC_WMQ_TEMPORARY_MODEL

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome da fila modelo do IBM WebSphere MQ a partir da qual uma fila dinâmica é criada quando o aplicativo cria um XMS fila temporária.

O valor padrão da propriedade é SYSTEM.DEFAULT.MODEL.QUEUE.

Essa propriedade é relevante apenas para o domínio ponto-a-ponto

XMSC_WMQ_WILDCARD_FORMAT

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory, Destino

Essa propriedade determina qual versão de sintaxe curinga deve ser usada.

Ao usar a publicação / assinatura com IBM WebSphere MQ '*' e '?' são tratados como curingas Considerando que '#' e '+' são tratados como curingas ao usar publicar assinatura com WebSphere Message Broker. Essa propriedade substitui a propriedade XMSC_WMQ_BROKER_VERSION..

Os valores válidos para essa propriedade são:

XMSC_WMQ_WILDCARD_TOPIC_ONLY
Reconhece os curingas de nível de tópico apenas, ou seja, '#' e '+' são tratados como curingas Esse valor é igual a XMSC_WMQ_BROKER_V2.

XMSC_WMQ_WILDCARD_CHAR_ONLY
Reconhece os caracteres curingas apenas, ou seja, '*' e '?' são tratados como curingas Esse valor é igual a XMSC_WMQ_BROKER_V1.

Por padrão, essa propriedade é configurada para XMSC_WMQ_WILDCARD_TOPIC_ONLY

Nota: Essa propriedade não é relevante ao fazer a publicação / assinatura usando IBM WebSphere MQ Versão 6.0 e abaixo Em vez disso, deve-se usar a propriedade XMSC_WMQ_BROKER_VERSION.

XMSC_WPM_BUS_NAME

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory e Destino

Nome usado em um URI:
busName

Para um connection factory, o nome do barramento de integração de serviços ao qual o aplicativo se conecta ou, para um destino, o nome do barramento de integração de serviços no qual o destino existe.

Para um destino que é um tópico, essa propriedade é o nome do barramento de integração de serviços no qual o espaço de tópicos associado existe... Este espaço de tópico é especificado pelo XMSC_WPM_TOPIC_SPACE de propriedade

Se a propriedade não for configurada para um destino, a fila ou o espaço de tópico associado será considerado existente no barramento de integração de serviços ao qual o aplicativo se conecta.

Por padrão, a propriedade não é configurada

XMSC_WPM_CONNECTION_PROTOCOL

Tipo de dado:
System.Int32

Propriedade de:
Conexão

O protocolo de comunicações usado para a conexão com o mecanismo do sistema de mensagens. Essa propriedade é somente leitura.

Os valores possíveis da propriedade são os seguintes:

Value	Significado
XMSC_WPM_CP_HTTP	A conexão usa HTTP sobre TCP/IP..
XMSC_WPM_CP_TCP	A conexão usa TCP/IP.

XMSC_WPM_CONNECTION_PROXIMIDADE

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

A configuração de proximidade de conexão para a conexão. Essa propriedade determina quão próximo o mecanismo do sistema de mensagens ao qual o aplicativo se conecta deve estar do servidor de autoinicialização.

Os valores válidos da propriedade são os seguintes:

Valor válido	Configuração de proximidade de conexão..
XMSC_WPM_CONNECTION_PROXIMITY_BUS	Barramento
XMSC_WPM_CONNECTION_PROXIMITY_CLUSTER	Cluster
XMSC_WPM_CONNECTION_PROXIMITY_HOST	Host

Valor válido	Configuração de proximidade de conexão..
XMSC_WPM_CONNECTION_PROXIMITY_SERVER	Servidor

O valor padrão é XMSC_WPM_CONNECTION_PROXIMITY_BUS.

XMSC_WPM_DUR_SUB_HOME

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Nome usado em um URI:
durableSubscriptionInício

O nome do mecanismo do sistema de mensagens no qual todas as assinaturas duráveis para uma conexão ou um destino são gerenciadas. Mensagens a serem entregues aos assinantes duráveis são armazenadas no ponto de publicação do mesmo mecanismo do sistema de mensagens.

Um lar de assinaturas duráveis deve ser especificado para uma conexão antes que um aplicativo possa criar um assinante durável que use a conexão. Qualquer valor especificado para um destino substitui o valor especificado para a conexão.

Por padrão, a propriedade não é configurada

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura

XMSC_WPM_HOST_NAME

Tipo de dado:
Sequência

Propriedade de:
Conexão

O nome do host ou o endereço IP do sistema que contém o mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Essa propriedade é somente leitura.

XMSC_WPM_LOCAL_ADDRESS

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Para uma conexão com um barramento de integração de serviços, essa propriedade especifica a interface de rede local a ser usada, a porta local ou o intervalo de portas locais a serem usados ou ambos.

O valor da propriedade é uma sequência com o seguinte formato:

[*host_name*] [(*low_port*) [,*high_port*]]

Os significados das variáveis são os seguintes:

host_name
O nome do host ou o endereço IP da interface de rede local a ser usado para a conexão

Fornecer essas informações é necessário apenas se o sistema no qual o aplicativo está em execução tiver duas ou mais interfaces de rede e você precisar ser capaz de especificar qual interface deve ser usada para a conexão.. Se o sistema tiver apenas uma interface de rede, somente essa interface poderá ser usada Se o sistema tiver duas ou mais interfaces de rede e você não especificar qual interface deve ser usada, a interface será selecionada aleatoriamente

low_port
O número da porta local a ser usada para a conexão

Se *high_port* também for especificado, *low_port* será interpretado como o número da porta mais baixo em um intervalo de números de portas

high_port
O número da porta mais alto em um intervalo de números de porta Uma das portas no intervalo especificado deve ser usada para a conexão..

Aqui estão alguns exemplos de valores válidos da propriedade:

JÚPITER
9.20.4.98
JUPITER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

Por padrão, a propriedade não é configurada

XMSC_WPM_ME_NAME

Tipo de dado:
Sequência

Propriedade de:
Conexão

O nome do mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Essa propriedade é somente leitura.

XMSC_WPM_NON_PERSISTENT_MAP

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O nível de confiabilidade de mensagens não persistentes que são enviadas usando a conexão.

Os valores válidos da propriedade são os seguintes:

Valor válido	Nível de confiabilidade
XMSC_WPM_MAPPING_AS_DESTINATION	Determinado pelo nível de confiabilidade padrão especificado para a fila ou espaço de tópicos no barramento de integração de serviços
XMSC_WPM_MAPPING_BEST_SEM esforço PERSISTENT	Best effort nonpersistent
XMSC_WPM_MAPPING_EXPRESS_NON_ PERSISTENT	Express nonpersistent
XMSC_WPM_MAPPING_RELIABLE_NON_ PERSISTENT	Confiável Não Persistente
XMSC_WPM_MAPPING_RELIABLE_PERSISTENT	Persistente confiável
XMSC_WPM_MAPPING_ASSURED_PERSISTENT	Garantido Persistente

O valor padrão é XMSC_WPM_MAPPING_EXPRESS_NON_PERSISTENT.

XMSC_WPM_PERSISTENT_MAP

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O nível de confiabilidade de mensagens persistentes que são enviadas usando a conexão.

Os valores válidos da propriedade são os seguintes:

Valor válido	Nível de confiabilidade
XMSC_WPM_MAPPING_AS_DESTINATION	Determinado pelo nível de confiabilidade padrão especificado para a fila ou espaço de tópicos no barramento de integração de serviços
XMSC_WPM_MAPPING_BEST_SEM esforço PERSISTENT	Best effort nonpersistent
XMSC_WPM_MAPPING_EXPRESS_NON_ PERSISTENT	Express nonpersistent
XMSC_WPM_MAPPING_RELIABLE_NON_ PERSISTENT	Confiável Não Persistente
XMSC_WPM_MAPPING_RELIABLE_PERSISTENT	Persistente confiável
XMSC_WPM_MAPPING_ASSURED_PERSISTENT	Garantido Persistente

O valor padrão é XMSC_WPM_MAPPING_RELIABLE_PERSISTENT.

XMSC_WPM_PORT

Tipo de dado:
System.Int32

Propriedade de:
Conexão

O número da porta atendida pelo mecanismo do sistema de mensagens para o qual o aplicativo está conectado. Essa propriedade é somente leitura.

XMSC_WPM_PROVIDER_ENDPOINTS

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

Uma sequência de um ou mais endereços de terminal de servidores de autoinicialização. Os endereços de terminal são separados por vírgulas.

Um servidor de autoinicialização é um servidor de aplicativos responsável por selecionar o mecanismo do sistema de mensagens ao qual o aplicativo se conecta. O endereço do terminal de um servidor de autoinicialização tem o seguinte formato:

host_name:*port_number*:*chain_name*

Os significados dos componentes de um endereço de terminal são os seguintes:

host_name
O nome do host ou endereço IP do sistema no qual o servidor de autoinicialização reside. Se nenhum nome do sistema central ou endereço IP for especificado, o padrão será `localhost`.

port_number
O número da porta na qual o servidor de autoinicialização atende solicitações recebidas. Se nenhum número de porta for especificado, o padrão será 7276..

chain_name
O nome de uma cadeia de transporte de autoinicialização usada pelo servidor de autoinicialização. Os valores válidos são os seguintes:

Valor válido	Nome da cadeia de transporte de autoinicialização
XMSC_WPM_BOOTSTRAP_HTTP	Sistema de Mensagens BootstrapTunneled
XMSC_WPM_BOOTSTRAP_HTTPS	BootstrapTunneledSecureMessaging
XMSC_WPM_BOOTSTRAP_SSL	Sistema de Mensagens BootstrapSecure
XMSC_WPM_BOOTSTRAP_TCP	Sistema de Mensagens BootstrapBasic

Se nenhum nome for especificado, o valor padrão será XMSC_WPM_BOOTSTRAP_TCP.

Se nenhum endereço de terminal for especificado, o padrão será localhost:7276:BootstrapBasicMessaging.

XMSC_WPM_TARGET_GROUP

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome de um grupo de destinos de mecanismos do sistema de mensagens. A natureza do grupo de destino é determinada pelo XMSC_WPM_TARGET_TYPE de propriedade

Configure esta propriedade se desejar restringir a procura de um mecanismo do sistema de mensagens para um subgrupo dos mecanismos de sistema de mensagens no barramento de integração de serviços Se você desejar que seu aplicativo possa se conectar a qualquer mecanismo do sistema de mensagens no barramento de integração de serviços, não configure essa propriedade.

Por padrão, a propriedade não é configurada

XMSC_WPM_TARGET_SIGNIFICATIVO

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O significado do grupo de destinos dos mecanismos do sistema de mensagens.

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_WPM_TARGET_SIGNIFICANCE_ Preferido	Um mecanismo do sistema de mensagens no grupo de destino será selecionado, se um estiver disponível Caso contrário, um mecanismo do sistema de mensagens fora do grupo de destino será selecionado, contanto que ele esteja no mesmo barramento de integração de serviços

Valor válido	Significado
XMSC_WPM_TARGET_SIGNIFICANCE_ REQUIRED	O mecanismo do sistema de mensagens selecionado deve estar no grupo de destinos Se um mecanismo do sistema de mensagens no grupo de destino não estiver disponível, o processo de conexão falhará

O valor padrão da propriedade é XMSC_WPM_TARGET_SIGNIANCE_PREFERRED.

XMSC_WPM_TARGET_TRANSPORT_CHAIN

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O nome da cadeia de transporte de entrada que o aplicativo deve usar para se conectar a um mecanismo do sistema de mensagens.

O valor da propriedade pode ser o nome de qualquer cadeia de transporte de entrada disponível no servidor de aplicativos que hospeda o mecanismo do sistema de mensagens. A constante nomeada a seguir é fornecida para uma das cadeias de transporte de entrada predefinidas:

Constante nomeada	Nome da cadeia de transporte
XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC	Sistema de Mensagens InboundBasic

O valor padrão da propriedade é XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC.

XMSC_WPM_TARGET_TYPE

Tipo de dado:
System.Int32

Propriedade de:
ConnectionFactory

O tipo do grupo de destinos de mecanismos do sistema de mensagens. Essa propriedade determina a natureza do grupo de destino identificado pela propriedade XMSC_WPM_TARGET_GROUP

Os valores válidos da propriedade são os seguintes:

Valor válido	Significado
XMSC_WPM_TARGET_TYPE_BUSMEMBER	O nome do grupo de destino é o nome de um membro do barramento O grupo de destino é todos os mecanismos do sistema de mensagens no membro do barramento
XMSC_WPM_TARGET_TYPE_CUSTOM	O nome do grupo de destino é o nome de um grupo definido pelo usuário de mecanismos do sistema de mensagens O grupo de destino é todos os mecanismos do sistema de mensagens registrados com o grupo definido pelo usuário.
XMSC_WPM_TARGET_TYPE_ME	O nome do grupo de destino é o nome de um mecanismo do mensagens. O grupo de destino é o mecanismo do sistema de mensagens especificado

Por padrão, a propriedade não é configurada

XMSC_WPM_TEMP_Q_PREFIX

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O prefixo usado para formar o nome da fila temporária que é criada no barramento de integração de serviços quando o aplicativo cria um XMS fila provisória. O prefixo pode conter até 12 caracteres..

O nome de uma fila temporária começa com os caracteres "_Q" seguidos pelo prefixo. O restante do nome consiste em caracteres gerados pelo sistema..

Por padrão, a propriedade não é configurada, o que significa que o nome de uma fila temporária não tem um prefixo

Essa propriedade é relevante apenas para o domínio ponto-a-ponto

XMSC_WPM_TEMP_TOPIC_PREFIX

Tipo de dado:
Sequência

Propriedade de:
ConnectionFactory

O prefixo usado para formar o nome de um tópico temporário que é criado pelo aplicativo. O prefixo pode conter até 12 caracteres..

O nome de um tópico temporário começa com os caracteres "_T" seguidos pelo prefixo. O restante do nome consiste em caracteres gerados pelo sistema..

Por padrão, a propriedade não é configurada, o que significa que o nome de um tópico temporário não possui um prefixo..

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura

XMSC_WPM_TOPIC_SPACE

Tipo de dado:
Sequência

Propriedade de:
Destino

Nome usado em um URI:
topicSpace

O nome do espaço de tópico que contém o tópico. Apenas um destino que é um tópico pode ter essa propriedade

Por padrão, a propriedade não é configurada, o que significa que o espaço de tópico padrão é assumido

Essa propriedade é relevante apenas para o domínio Publicação/Assinatura

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 um IBM produto, programa ou serviço não se destina a estado ou significa que apenas esse produto IBM, programas ou serviços possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição. 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:

Relações Comerciais e Industriais da IBM
Av. Pasteur, 138-146
Botafogo
Rio, RJ 10504-1785
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:

licença de propriedade intelectual
IBM World Trade Asia Corporation Licensing
IBM Japan, Ltd.
Minato-ku
Tóquio 103-8510, Japão

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. Periodicamente, são feitas nas informações aqui contidas; essas 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.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem 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 IBM 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 comerciais 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 (http://www.eclipse.org/).

Java e todas as marcas comerciais e logotipos baseados em Java são marcas comerciais ou marcas registradas da Oracle e/ou de suas afiliadas.

Part Number:

(1P) P/N: